



“LFLATS” โปรแกรมการออกแบบแผ่นพื้นไว้ถาน

“LFLATS” FLAT SLAB AND FLAT PLATE DESIGN PROGRAM



โดย
นางสาวพัชรารัตน์ ไหม่ตา รหัสประจำตัว 35104291
นายสยาม กันเกตุ รหัสประจำตัว 35104449

วัน เดือน ปี..... ๑ ก.ค. ๒๕๕๐
เลขทะเบียน..... ๐๓๗๑๗๗
เลขเรียกหนังสือ..... T ๐๓๖๗๐ พ.๕๖๖ น.

โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมการก่อสร้าง

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2538

037177

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกกรณี

“ LFLATS ” FLAT SLAB AND FLAT PLATE DESIGN PROGRAM



**A SPECIAL PROJECT SUBMITTED IN PARTIAL FULLFILLMENT
OF THE REQUIREMENT FOR THE BACHELOR DEGREE OF
CONSTRUTION TECHNOLOGY DEPARTMENT
KING'S MONGKUT INSTITUTE OF TECHNOLOGY LADKRABANG**

1995

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมโยธา คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง
ใบรับรองโครงการพิเศษ

หัวข้อโครงการพิเศษ “LFLATS” โปรแกรมการออกแบบแผ่นพื้นไร้คาน
“LFLATS” FLAT SLAB AND FLAT PLATE
DESIGN PROGRAM

นักศึกษา นางสาว พัชรารัตน์ ไหม่ตา
นาย สยาม กันแกตุ

หลักสูตร วิศวกรรมศาสตรบัณฑิต
สาขา วิศวกรรมการก่อสร้าง
ภาควิชา วิศวกรรมโยธา
อาจารย์ที่ปรึกษา อาจารย์ วิบูลย์ วุฒินุญ

| คณะกรรมการตรวจสอบโครงการพิเศษ | ลายมือชื่อ |
|--------------------------------|------------|
| 1. อาจารย์ ศรีกริช หิรัญมาศ | |
| 2. อาจารย์ วิบูลย์ วุฒินุญ | |
| 3. อาจารย์ สมชาย สำลีรางค์กุล | |
| 4. อาจารย์ สมเกียรติ ขวัญพฤกษ์ | |

ภาควิชาวิศวกรรมโยธารับรองแล้ว

(นายอำนาจ พานิชกุลพงศ์)
หัวหน้าภาควิชาวิศวกรรมโยธา
วันที่ เดือน พ.ศ.

หัวข้อโครงการพิเศษ “LFLAT” โปรแกรมการออกแบบแผ่นพื้นไร้คาน
“LFLAT” : FLAT SLAB AND FLAT PLATE
DESIGN PROGRAM

นักศึกษา นางสาวพัชรารัตน์ ไหมตา รหัสประจำตัว 35104291
นายสยาม กันแกตุ รหัสประจำตัว 35104449
อาจารย์ที่ปรึกษา อาจารย์ขวัญชัย วุฒินุภาพ
ระดับการศึกษา วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมการก่อสร้าง
ภาควิชา วิศวกรรมโยธา คณะวิศวกรรมศาสตร์
ปีการศึกษา 2538

บทคัดย่อ

โครงการพิเศษนี้จัดทำขึ้นเพื่อนำ คอมพิวเตอร์เข้ามาช่วยอำนวยความสะดวกในการออกแบบแผ่นพื้นไร้คาน โดยการเขียนโปรแกรมขึ้นมาใช้งานในคอมพิวเตอร์โดยใช้ภาษา Visual BASIC ซึ่งสามารถทำงานบนระบบปฏิบัติการ Windows และมีการพัฒนาด้าน GUI (Graphical User Interface) ทำให้โปรแกรมนำใช้ และการออกแบบแผ่นพื้นคอนกรีตอัดแรง โดยการเขียนโปรแกรมขึ้นมาใช้งานในคอมพิวเตอร์ โดยใช้ภาษา Visual BASIC ซึ่งสามารถทำงานบนระบบปฏิบัติการ Windows และมีการพัฒนาด้าน GUI (Graphical User Interface) ทำให้โปรแกรมนำใช้ และสะดวกรวดเร็วมากขึ้นหลักการทำงานคือติดต่อกับผู้ใช้ รับรายละเอียดต่าง ๆ ที่ใช้ในการออกแบบแล้วนำมาคำนวณโครงสร้างด้วยโปรแกรมคำนวณการออกแบบใช้ข้อกำหนดตามมาตรฐาน ACI และ ว.ส.ท. ตามความเหมาะสม การวิเคราะห์โมเมนต์ใช้ ทฤษฎีโครงเปรียบเทียบ (Equivalent Frame) ซึ่งเป็นกรวิเคราะห์แบบ Elastic ที่มี สมมุติฐานที่แน่นอนใกล้เคียงกับความเป็นจริง ส่วนการวิเคราะห์เหล็กเสริมใช้ทฤษฎี กำลังประลัยซึ่งจะทำให้เกิดความประหยัด

ABSTRACT

This special project's aim is to create a Computer Aided Programme in designing of flat slab and flat plat structures. The program simplifies the complex process in designing of these structures. Equivalent frame method is used in the analysis part since it provides more accurate results than the normal empirical method. Member design part is based on strength design method following American Concrete Institute and Engineering Institute of Thailand Codes. Visual Basic language is used to provide better graphical user interface(GUI). The program runs on Windows 3.1 or higher.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สามารถสำเร็จลงได้ด้วยดี ก็เนื่องจกได้รับความช่วยเหลือ และคำแนะนำในการทำงานจากบุคคลต่างๆ ดังนั้นคณะผู้จัดทำจึงขอแสดงความขอบคุณ

| | | |
|------------------|--------------|----------------------------|
| อาจารย์วิบูลย์ | วุฒินุญาณ | อาจารย์ที่ปรึกษา |
| อาจารย์สมเกียรติ | ขวัญพฤษ์ | อาจารย์ภาควิชาวิศวกรรมโยธา |
| อาจารย์อำนาจ | พานิชกุลพงศ์ | อาจารย์ภาควิชาวิศวกรรมโยธา |
| อาจารย์ศรีกริช | หิรัญมาศ | อาจารย์ภาควิชาวิศวกรรมโยธา |
| อาจารย์สมชาย | คำลีรางค์กุล | อาจารย์ภาควิชาวิศวกรรมโยธา |
| อาจารย์ศิริวัฒน์ | ไชยชนะ | อาจารย์ภาควิชาวิศวกรรมโยธา |
| อาจารย์จักรพงษ์ | พงษ์เพ็ญ | อาจารย์ภาควิชาวิศวกรรมโยธา |
| อาจารย์สุพจน์ | ศรีนิต | อาจารย์ภาควิชาวิศวกรรมโยธา |
| อาจารย์คมสันต์ | มาลีศรี | อาจารย์ภาควิชาวิศวกรรมโยธา |

เพื่อนๆ และ พี่ๆ ทุกคนที่ให้ความช่วยเหลือในการทำโครงการนี้

คณะผู้จัดทำ

| สารบัญ | หน้า |
|--|------|
| ตอนที่ 1 | |
| บทที่ 1 บทนำ | 1 |
| บทที่ 2 ทฤษฎีและหลักการออกแบบแผ่นพื้นไร้คาน | 3 |
| 2.1 ลักษณะโดยทั่วไปของแผ่นพื้นไร้คาน | 3 |
| 2.2 คานขอบ (Edge beam) | 4 |
| 2.3 แป้นหัวเสา (Drop panel) | 5 |
| 2.4 หัวเสา (Column Capital) | 5 |
| 2.5 ข้อดีของพื้นระบบแผ่นพื้นไร้คาน | 6 |
| 2.6 หลักการในการออกแบบ | 6 |
| 2.7 ข้อกำหนดความปลอดภัย | 7 |
| บทที่ 3 ขั้นตอนการออกแบบ | 9 |
| 3.1 การหาความหนาของแผ่นพื้น | 9 |
| 3.2 การตรวจสอบแรงเฉือน | 9 |
| 3.2.1 ลักษณะคานกว้าง (Wide beam Action) | 10 |
| 3.2.2 ลักษณะเฉือนทะลุ (Two way Action) | 11 |
| 3.3 การวิเคราะห์หาค่าโมเมนต์ | 13 |
| 3.3.1 การหาค่า Fixed end moment | 21 |
| 3.3.2 การหาค่าการกระจายโมเมนต์ Distribution factor | 23 |
| 3.3.3 การหาค่า Carry over factor | 26 |
| 3.4 การหาโมเมนต์ในการออกแบบ | 26 |
| 3.4.1 การหาค่าโมเมนต์ลบ | 26 |
| 3.4.2 การหาค่าโมเมนต์บวกที่คึ่งกลางช่วงพื้น | 27 |
| 3.5 การกระจายโมเมนต์เข้าสู่แถบเสาและแถบกลาง | 28 |
| บทที่ 4 การหาปริมาณเหล็กเสริม | 30 |
| 4.1 กรณีเหล็กเสริมรับแรงดึง | 30 |
| 4.2 กรณีเหล็กเสริมรับแรงอัด | 31 |
| 4.3 การจัดเหล็กเสริมในแผ่นพื้นไร้คาน | 33 |

| | |
|--|-------------|
| สารบัญ (ต่อ) | หน้า |
| ตอนที่ 2 | |
| คู่มือการใช้โปรแกรม LFLATS | 35 |
| (Manual of LFLATS) | |
| บทนำ | 36 |
| การเข้าสู่โปรแกรม | 37 |
| อุปกรณ์และซอฟต์แวร์ที่ต้องใช้ควบคู่กับโปรแกรม LFLATS | 38 |
| ส่วนประกอบโปรแกรมโดยละเอียด | 39 |
| ลำดับขั้นตอนการใช้งานของโปรแกรม LFLATS | 62 |
| | |
| ตอนที่ 3 | |
| FLOW CHART สำหรับการเขียนโปรแกรม | 66 |
| 1. FLOW CHART FOR DETERMINE THE THICKNESS | 67 |
| 2. FLOW CHART FOR CHECKING SHEAR | 68 |
| 3. FLOW CHART FOR SEQUENCE OF OPERATIONS FOR DETERMINE THE MOMENT | 69 |
| 4. FLOW CHART FOR DISTRIBUTE THE DESIGN MOMENT | 70 |
| 5. FLOW CHART FOR THE SEQUENCE OF OPERATIONS FOR THE DESIGN OF REINFORCED | 71 |
| 6. FLOW CHART FOR ANALYSIS OF SINGLY REINFORCED | 72 |
| 7. FLOW CHART FOR ANALYSIS OF DOUBLY REINFORCED | 73 |
| | |
| ตอนที่ 4 | |
| ตัวอย่างการคำนวณ | 74 |
| บทสรุปและวิจารณ์ | 83 |
| ภาคผนวก | 85 |
| PROGRAM LISTING | 86 |
| บรรณานุกรม | |

| สารบัญรูปภาพ | หน้า |
|---|------|
| รูปที่ 2.1 ลักษณะของแผ่นพื้นไร้คาน | 3 |
| รูปที่ 2.1 ก แผ่นพื้นไร้คาน | 3 |
| รูปที่ 2.1 ข แผ่นพื้นบางไร้คาน | 3 |
| รูปที่ 2.2 รูปอธิบายหัวเสา | 5 |
| รูปที่ 2.3 แสดงลักษณะการแบ่งแผ่นพื้นเป็นโครง | 7 |
| รูปที่ 3.1 พื้นที่รับแรงเฉือนลักษณะคานกว้าง | 11 |
| รูปที่ 3.2 พื้นที่รับแรงเฉือนลักษณะเฉือนทะลุ | 12 |
| รูปที่ 3.2 ก กรณีเสาด้านใน | 12 |
| รูปที่ 3.2 ข กรณีเสาด้านนอก | 12 |
| รูปที่ 3.2 ค กรณีเสาที่มุม | 12 |
| รูปที่ 3.3 แผนภาพแสดงสัมประสิทธิ์ a,b,c,d,r,s | 16 |
| รูปที่ 3.4 แผนภาพการหาค่าประกอบสัมประสิทธิ์ และตัวแปรของเสา | 17 |
| รูปที่ 3.4.1 กรณีไม่มีเป็นรองรับหัวเสา | 17 |
| รูปที่ 3.4.2 กรณีที่มีเป็นรองรับหัวเสา แต่ไม่มีหัวเสา | 17 |
| รูปที่ 3.4.3 กรณีที่มีเป็นรองรับหัวเสา และหัวเสา | 17 |
| รูปที่ 3.5 แผนภาพประกอบการหาค่าสัมประสิทธิ์ และตัวแปรของแผ่นพื้น | 18 |
| รูปที่ 3.5.1 กรณีไม่มีเป็นรองรับหัวเสา และหัวเสา | 18 |
| รูปที่ 3.5.2 กรณีมีเป็นรองรับหัวเสา แต่ไม่มีหัวเสา | 19 |
| รูปที่ 3.5.3 กรณีมีเป็นรองรับหัวเสา และหัวเสา | 20 |
| รูปที่ 3.6.1 load pattern 1 | 21 |
| รูปที่ 3.6.2 load pattern 2 | 21 |
| รูปที่ 3.6.3 load pattern 3 | 22 |
| รูปที่ 3.6.4 load pattern 4 | 22 |
| รูปที่ 3.6.5 load pattern 5 | 22 |
| รูปที่ 3.7 หน้าตัดในการหาค่า C | 24 |
| รูปที่ 3.8 รูปแสดงระยะวิกฤติสำหรับการหาโมเมนต์ | 26 |
| รูปที่ 3.10 แสดงแถบเสา (column strip) และแถบกลาง (middle strip) | 29 |

| สารบัญรูปภาพ (ต่อ) | หน้า |
|---|------|
| รูปที่ 4.1 แสดงระยะทาบและการเสริมเหล็ก | 34 |
| | |
| ตอนที่ 2 | |
| รูปที่ 1 แสดงไอคอนของโปรแกรม LFLATS | 37 |
| รูปที่ 2 เป็นหน้าจอที่แสดงการแนะนำโปรแกรมการออกแบบ เมื่อเริ่มต้นใช้โปรแกรม | 38 |
| รูปที่ 3 แสดงส่วนต่างๆบนหน้าจอของตัวโปรแกรม | 39 |
| รูปที่ 4 แสดงหน้าจอซึ่งแสดง Pull down Menu File และเมนูย่อยต่างๆ | 40 |
| รูปที่ 5 แสดงหน้าจอซึ่งแสดง Pull down Menu Input และเมนูย่อยต่างๆ | 41 |
| รูปที่ 6 แสดงหน้าจอซึ่งแสดง Pull down Menu Design และเมนูย่อยต่างๆ | 42 |
| รูปที่ 7 แสดงหน้าจอซึ่งแสดง Pull down Menu Result และเมนูย่อยต่างๆ | 43 |
| รูปที่ 8 แสดงหน้าจอซึ่งแสดง Pull down Menu Save setting และเมนูย่อยต่างๆ | 44 |
| รูปที่ 9 แสดงหน้าจอซึ่งแสดง Pull down Menu Option และเมนูย่อยต่างๆ | 45 |
| รูปที่ 10 แสดงหน้าจอซึ่งแสดง Pull down Menu About และเมนูย่อยต่างๆ | 46 |
| รูปที่ 11 แสดงหน้าจอซึ่งแสดง การคลิก System Tool | 47 |
| รูปที่ 12 แสดงหน้าจอซึ่งแสดงหน้าต่าง Output ของการ Design | 49 |
| รูปที่ 13 แสดงหน้าจอซึ่งแสดงหน้าต่าง Output ของ Slab To Design | 50 |
| รูปที่ 14 แสดง Shear Force Diagram | 51 |
| รูปที่ 15 แสดง พื้นที่เหล็กเสริม | 52 |
| รูปที่ 16 แสดง Output รายการคำนวณ | 53 |
| รูปที่ 17 แสดงหน้าจอซึ่งแสดงการคลิก Data Tool | 54 |
| รูปที่ 18 แสดงหน้าจอซึ่งแสดง Panel การใส่ข้อมูลของพื้น | 55 |
| รูปที่ 19 แสดงหน้าจอซึ่งแสดง Panel ของ Design' strip | 57 |

สารบัญรูปภาพ (ต่อ)

หน้า

| | |
|---|----|
| รูปที่ 20 แสดงหน้าจอซึ่งแสดง Panel การใส่ข้อมูล ของแรงกระทำบนพื้นและ Load factor | 59 |
| รูปที่ 21 แสดงหน้าจอซึ่งแสดง Panel การใส่ข้อมูล Information | 60 |
| รูปที่ 22 แสดงหน้าจอซึ่งแสดง Panel การใส่ข้อมูล Information | 61 |





ตอนที่ 1

บทที่ 1

บทนำ

ในปัจจุบันการออกแบบพื้นไร้คานค่อนข้างจะเป็นที่นิยมแพร่หลายอันเนื่องมาจากการต้องการพื้นที่ใช้สอยมากขึ้นบนเนื้อที่และความสูงที่จำกัด ทำให้ต้องลดความสูงของอาคารลงโดยการตัดคานทิ้งออกไปซึ่งจะทำให้ความสูงจากชั้นถึงชั้นลดลงและยังสามารถทำการเจาะแผ่นพื้นในภายหลังได้อีกด้วย

ขั้นตอนการออกแบบแผ่นพื้นไร้คานมีความยุ่งยากมากกว่าการออกแบบแผ่นพื้นสองทางที่มีคานรองรับอยู่เพราะ ในแผ่นพื้นไร้คานจะถูกพิจารณาให้มีคานตามแนวเสาภายในแต่มีค่าความแข็งเกร็งยืดหยุ่นเท่ากับศูนย์ (Zero flexural stiffness) การหาโมเมนต์เพื่อนำมาใช้ในการออกแบบค่อนข้างจะซับซ้อน วิธีการที่ใช้โดยการประมาณค่าที่หามาได้มักจะมีค่ามากกว่าความจริงค่อนข้างมาก ซึ่งอาจจะส่งผลให้ราคาค่าก่อสร้างมากเกินไปหรืออาจเกิดความเสียหายต่อโครงสร้างภายหลังได้ เพราะส่วนใหญ่แผ่นพื้นไร้คานจะมีช่วงพื้นที่กว้างและรับน้ำหนักมากอยู่แล้ว อีกทั้งผู้ที่มีความเชี่ยวชาญในการออกแบบพื้นไร้คานยังมีอยู่น้อย หากมีการรวบรวมข้อมูลการออกแบบที่ถูกต้องแล้วนำมาเขียนเป็น โปรแกรมคอมพิวเตอร์สำหรับใช้งานจะสามารถลดปัญหาดังกล่าวนี้ได้ โครงการพิเศษนี้วิเคราะห์หาโมเมนต์และแรงเฉือนในแผ่นพื้นไร้คานด้วยวิธีที่ค่อนข้างจะได้ค่าที่ใกล้เคียงความจริงมากกว่าวิธี การออกแบบโดยตรง (Direct Design Method) ซึ่งได้แก่วิธี โครงเปรียบเทียบ (Equivalent Frame Method) เนื่องจากใช้ทฤษฎีที่มีสมมุติฐานที่ใกล้เคียงความเป็นจริงทำให้การออกแบบได้ผลถูกต้องใกล้เคียงกับความเป็นจริง และทำให้เกิดความปลอดภัย อีกทั้งเป็นวิธีที่เหมาะสมสำหรับการประยุกต์ใช้งานโดยไม่โครคอมพิวเตอร์ซึ่งเข้ามามีบทบาทสำคัญในยุคโลกาภิวัตน์ในปัจจุบัน ส่วนการตรวจสอบแรงเฉือนและการหาปริมาณเหล็กเสริมใช้ทฤษฎีกำลังประลัย เนื่องจากให้ความคาดหมายได้ใกล้เคียงกับพฤติกรรมตามความเป็นจริงของวัสดุ และจะทำให้ประหยัดค่าก่อสร้างมากกว่าใช้ทฤษฎีอีลาสติก

ข้อกำหนดต่างๆ ใช้มาตรฐาน ACI (The American Concrete Institute) เป็นหลัก เนื่องจากประเทศไทยมีมาตรฐานในการคำนวณเกี่ยวกับแรงเฉือน ซึ่งกำหนดโดย ว.ส.ท. (วิศวกรรมสถานแห่งประเทศไทย) โดยยึดหลักตามมาตรฐาน ACI

การนำมาเขียนเป็นโปรแกรมคอมพิวเตอร์เพื่อให้เกิดความรวดเร็วในการออกแบบ เนื่องจากการออกแบบตามทฤษฎีข้างต้นมีความซับซ้อนมาก และเครื่องคอมพิวเตอร์มีประสิทธิภาพสูงกว่าคนทั้งในด้านความคล่องตัวและความถูกต้องแม่นยำทำให้ความผิดพลาดที่จะเกิดขึ้นน้อยกว่าการวิเคราะห์โดยมนุษย์

การเขียนโปรแกรมเลือกใช้ภาษา Visual BASIC เนื่องจากเป็นภาษาที่มีการพัฒนาสำหรับใช้เขียนโปรแกรมอย่างสะดวกและมีการพัฒนาด้าน GUI (Graphical User Interface) ซึ่งเป็นการนำเสนอแก่ผู้ใช้โปรแกรมทำให้น่าใช้และใช้ได้ง่ายมากขึ้น นอกจากนี้ยังสามารถทำงานบนโปรแกรม Windows ทำให้ผู้ออกแบบสามารถทำงานโดยสะดวกยิ่งขึ้นทั้งนี้ผู้ที่ทำการออกแบบจะต้องมีความรู้พื้นฐานทางการออกแบบอยู่ด้วยพอสมควร



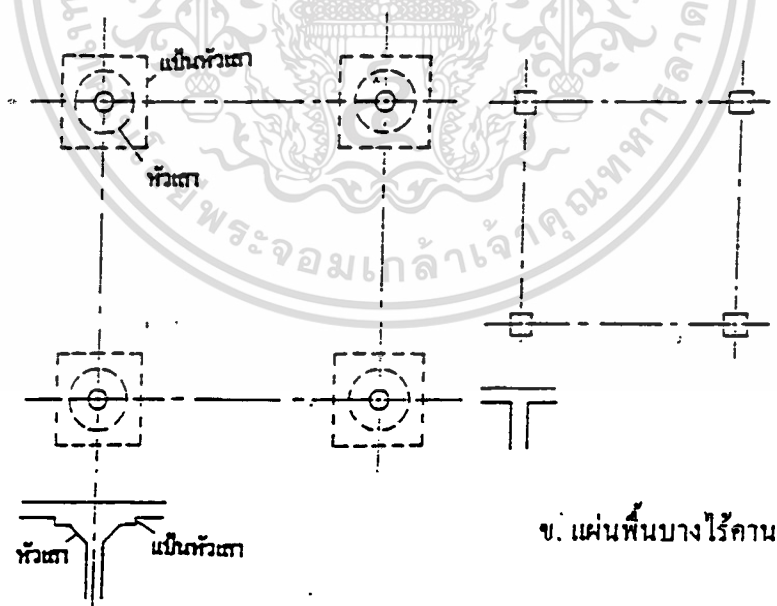
บทที่ 2

ทฤษฎีและหลักการออกแบบแผ่นพื้นไร้คาน

2.1 ลักษณะโดยทั่วไปของของแผ่นพื้นไร้คาน

แผ่นพื้นไร้คานเป็นแผ่นพื้นคอนกรีตเสริมเหล็กอย่างน้อยสองทางที่ตั้งได้ฉากกัน โดยปกติจะไม่มีคานสำหรับถ่ายน้ำหนักไปยังเสาที่รองรับ แต่จะถ่ายน้ำหนักโดยตรงจากพื้นไปยังเสาที่รองรับ แผ่นพื้นไร้คานแบ่งออกเป็น 2 ลักษณะคือ

1. Flat Slab ในกรณีที่แผ่นพื้นต้องรับน้ำหนักบรรทุกมาก ที่ปลายบนของเสามักขยายออกโดยมีลักษณะคล้ายกรวย เพื่อใช้ต้านทานแรงเฉือนที่รอบหัวเสาเรียกส่วนนี้ว่า “หัวเสา” (Column capital) ซึ่งจะออกแบบและหล่อให้เป็นเนื้อเดียวกันกับหัวเสาและแผ่นพื้น บางครั้งแผ่นพื้นส่วนที่อยู่เหนือเสามีความหนามากกว่าแผ่นพื้นทั่วไปเพื่อใช้ต้านทานแรงเฉือนและเพิ่มความสามารถต้านทานโมเมนต์คั้ระหว่างเสาและพื้น เรียกส่วนนี้ว่า “เป็นหัวเสา” (Drop panel)



ก. แผ่นพื้นไร้คาน

ข. แผ่นพื้นบางไร้คาน

รูปที่ 2.1 ลักษณะของแผ่นพื้นไร้คาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Flat Plate เป็นแผ่นพื้นไร้คานที่ประกอบด้วยแผ่นพื้นที่มีความหนาเท่ากันตลอด และมีเสารองรับน้ำหนัก ซึ่งอาจจะต้องมีการฝังเหล็กเสริม (shear head reinforcement) เหมาะสำหรับกรณีที่ช่วงความยาวสั้นๆและรับน้ำหนักบรรทุกไม่มาก เช่น อาคารที่พักอาศัย โดยทั่วไปแล้วแผ่นพื้นแบบ Flat Slab สามารถรับน้ำหนักบรรทุก และมีแผ่นพื้นที่มีช่วงของแผ่นพื้นได้มากกว่าและกว้างกว่าแผ่นพื้นแบบ Flat Plate

2.2 คานขอบ (Edge Beam)

โดยทั่วไปแม้ว่าคานจะไม่ถูกนำมาใช้ในการออกแบบแผ่นพื้นไร้คานก็ตาม แต่ในบริเวณช่วงพื้นนอกสุดด้านที่ไม่มีการต่อเนื่องยังคงมีการใช้คานอยู่บ้าง คานริมนอกดังกล่าวเรียกว่า “ คานขอบ ” (Edge Beam) ความแข็งเกร็ง (stiffness) ของคาน ขึ้นอยู่กับ EI/L เมื่อ E และ I ของคานมีค่าเท่ากันตลอด ดังนั้นค่า stiffness จึงขึ้นอยู่กับ I ซึ่งเท่ากับ $bh^3/12$

โดยที่

E = โมดูลัสยืดหยุ่นของคาน

I = โมเมนต์อินเนอร์เซียของคาน

L = ความยาวของคาน

b = ความกว้างของคาน

h = ความลึกของคาน

ความสัมพันธ์ของค่า stiffness ของคานต่อแผ่นพื้นที่อยู่ติดกับคานใช้สัญลักษณ์แทนด้วย α ค่า stiffness ของแผ่นพื้นคำนวณได้จาก $Bt^3/12$

เมื่อ

B = ความกว้างของแผ่นพื้นตามขวางที่ติดกับคานระหว่างแนวเส้นศูนย์กลางของแผ่นพื้นที่ตัดแล้วเปรียบเทียบกับเหมือน frame

t = ความหนาของแผ่นพื้น

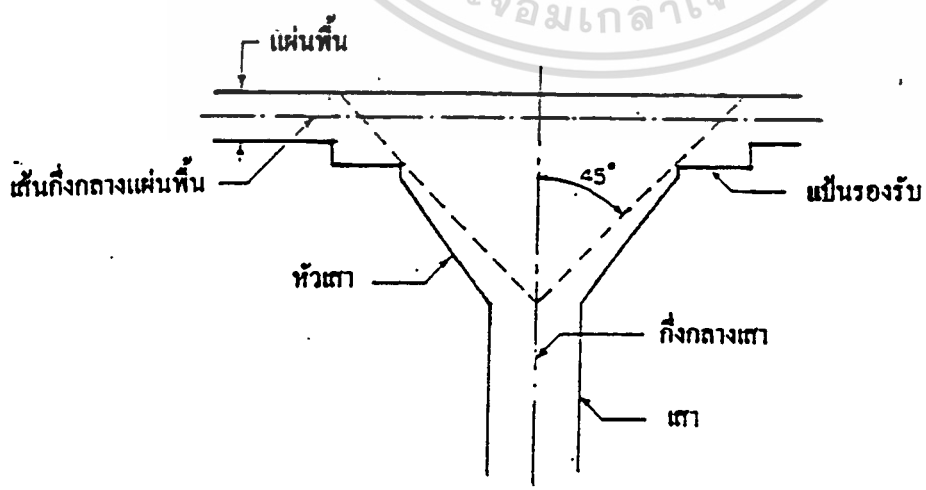
เมื่อไม่มีคานขอบหรือคานขอบเล็กมากจนกระทั่งค่า α น้อยกว่า 0.8 จะทำให้การยึดรั้งในแผ่นพื้นมีน้อยมาก จะต้องเพิ่มความหนาขึ้นอีก 10 % ตามข้อกำหนด ACI - 9.5.3.3 (สมการ 9-10 , 9-11 , 9-12)

2.3 แป้นหัวเสา (Drop panel)

แป้นหัวเสา คือ ส่วนของแผ่นพื้นที่อยู่บริเวณเหนือตำแหน่งเสาที่มีการเพิ่มความหนาให้มากกว่าแผ่นพื้นส่วนอื่นเพื่อช่วยการรับแรงเฉือน ขนาดกว้างหรือเส้นผ่านศูนย์กลางของแป้นหัวเสาจะต้องมีขนาดไม่น้อยกว่า 1/3 ของระยะช่วงที่วัดจากศูนย์กลางถึงศูนย์กลางของที่รองรับในทิศทางเดียวกัน ส่วนความหนาที่อยู่ใต้แผ่นพื้นจะต้องไม่น้อยกว่า 1/4 ของความหนาของแผ่นพื้นที่อยู่รอบนอกแป้นหัวเสา ข้อกำหนด ACI-9.5.3.2 อนุญาตให้ลดความหนาของแผ่นพื้นลงได้อีก 10% จากสมการ 9-10 , 9-11 , 9-12 และในการคำนวณหาปริมาณเหล็กเสริม จะต้องใช้ความหนาของแป้นหัวเสาส่วนที่อยู่ต่ำกว่าแผ่นพื้นไม่เกิน 1/4 ของระยะระหว่างขอบของแป้นหัวเสากับขอบของหัวเสา ตามข้อกำหนด ACI - 13.4.7.3

2.4 หัวเสา (Column capital)

หัวเสา คือ ส่วนขยายของเสาตรงที่พบกับแผ่นพื้นหรือแป้นหัวเสา จุดประสงค์ก็คือเพิ่มเส้นรอบรูปของเสาให้ใหญ่ขึ้นเพื่อจะรับการถ่ายทอดแรงเฉือนจากแผ่นพื้น การขยายกว้างออกของหัวเสาเป็นลักษณะกรวยกลมหงายหรือปิรามิดที่มีมุมกันกรวยหรือที่ยอดปิรามิดเท่ากับ 90 องศา มีรัศมีลดลงตามแนวเส้น 45 องศา เข้าหาแนวแกนเสาเพื่อกระจายแรงเฉือนเข้าสู่เสาตามข้อกำหนด ACI - 13.1.2 ปรกติเส้นผ่านศูนย์กลางของหัวเสาวัดที่ตำแหน่งใต้แผ่นพื้นหรือใต้แป้นหัวเสาจะกว้างประมาณ 20 ถึง 25 % ของความยาวช่วงเสาเฉลี่ย



รูปที่ 2.2 อธิบายหัวเสา

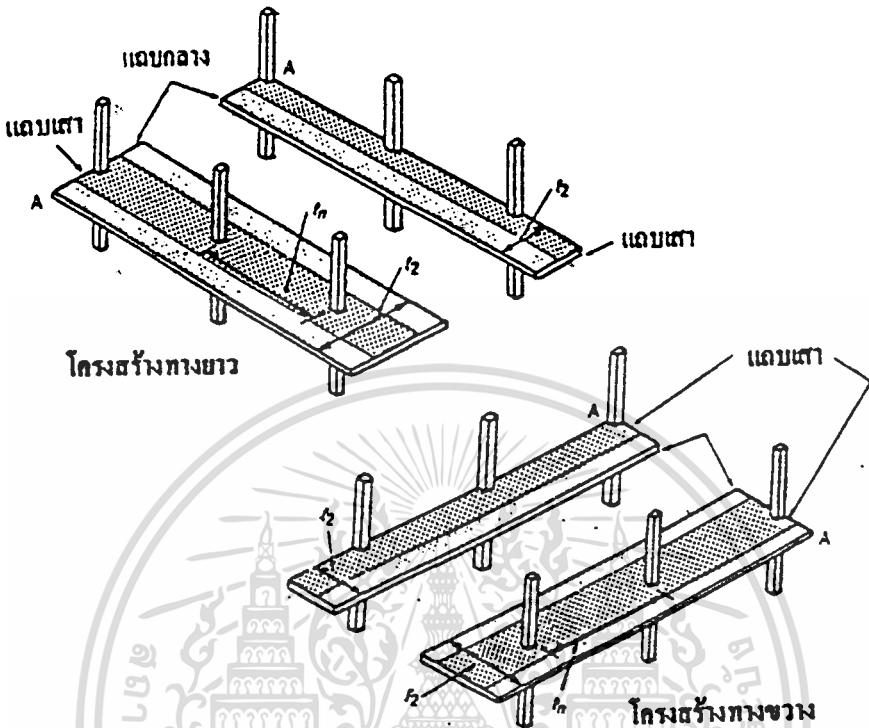
2.5 ข้อดีของพื้นระบบ แผ่นพื้นไร้คาน

1. ลดความสูงของอาคาร ความลึกของพื้นจะมีความหนาเพียงประมาณ 1/40 ถึง 1/45 ของช่วงเสาอีกทั้งเป็นระบบพื้นที่ไม่ต้องมีคานมารองรับ จึงสามารถลดความสูงของแต่ละชั้นลงได้ประมาณ 30-40 เซนติเมตร การลดความสูงของแต่ละชั้นลงได้นี้มีข้อดี คือในความสูงที่เท่ากันสามารถให้จำนวนชั้นที่มากกว่าและสำหรับจำนวนชั้นที่เท่ากันความสูงรวมของอาคารจะน้อยกว่า ซึ่งช่วยประหยัดค่าก่อสร้างเพราะช่วยประหยัดระบบรับแรงลม และเมื่อความสูงของผนังลดลงความยาวของท่อระบบต่างๆ ในแนวดิ่งก็ลดความยาวลงไปด้วย
2. ประหยัดแบบที่ใช้ในการก่อสร้าง ท้องพื้นที่เรียบการทำแบบจะไม่ยุ่งยากจึงสามารถประกอบแบบพื้นได้รวดเร็ว เพราะไม่ต้องตั้งไม้แบบและผูกเหล็กคานและยังสะดวกในการเดินสายไฟ ท่อน้ำ และท่อระบบต่างๆ ได้ที่ท้องพื้น
3. อาคารพักอาศัยที่ใช้พื้นระบบแผ่นพื้นไร้คานท้องเรียบ (Flat plate) ห้องจะได้รับแสงสว่างทั่วถึงดีขึ้นเพราะไม่มีท้องคานมาบัง
4. ประหยัดเวลาในการก่อสร้าง เนื่องจากการก่อสร้างแผ่นพื้นไร้คานทำได้สะดวก ท้องพื้นสามารถติดตั้งฝ้าเพดานได้เลย จึงสามารถลดเวลาในการก่อสร้างในส่วนอื่นๆ ได้
5. ในกรณีเกิดไฟไหม้ ท้องพื้นที่เรียบโล่งจะทนทานต่อเปลวไฟกว่า เพราะไม่มีส่วนที่เป็นสันหรือมุมซึ่งเมื่อถูกไฟจะกระเทาะแตกออกได้ง่าย

2.6 หลักการในการออกแบบ

แผ่นพื้นไร้คานถือเป็นแผ่นพื้นสองทางอย่างหนึ่ง ดังนั้นช่วงแผ่นพื้นต้องมีอัตราส่วนระหว่างด้านยาวกับด้านกว้างน้อยกว่า 2 การออกแบบเริ่มด้วยการกำหนดขนาดของแผ่นพื้นเสาะ และนำหนักที่ใช้ในการออกแบบ หลังจากการวิเคราะห์ที่ได้ความหนาของพื้น เป็นรองรับและหัวเสา (2 อย่างหลังอาจไม่มีก็ได้) จากนั้นทำการตรวจสอบแรงเฉือนซึ่งมี 2 ลักษณะ คือลักษณะคานกว้าง และลักษณะเฉือนทะลุ จากนั้นวิเคราะห์ค่าโมเมนต์คัดและหาปริมาณเหล็กเสริม การตรวจสอบแรงยึดเหนี่ยวโดยทั่วไปแล้วไม่จำเป็นต้องตรวจสอบเนื่องจากค่าแรงยึดเหนี่ยวที่กำหนดมกมีค่าสูงกว่าค่าแรงยึดเหนี่ยวที่เกิดขึ้นจริงค่อนข้างมาก

พื้นฐานในการออกแบบนั้นให้แบ่งโครงสร้างออกเป็นโครงข้อแข็ง (frame) โดยคิดทั้งตามทางยาว (longitudinal) และทางขวาง (transverse) ของสิ่งก่อสร้างเป็นแถบๆ ตาม ACI 8.9.1 โครงสร้างของโครงข้อแข็ง (rigid frame) ประกอบด้วยเสาหรือที่รองรับ (equivalent column) หนึ่งแถว พร้อมทั้งแถบของแผ่นพื้นแต่ละแถบมีความกว้างอยู่ระหว่างเส้นแบ่งกึ่งกลางของช่วงพื้นซึ่งอยู่สองข้างของเส้นแบ่งกึ่งกลางของเสาหรือที่รองรับ



รูปที่ 2.3 แสดงลักษณะการแบ่งแผ่นพื้นเป็นโครง (frame)

วิธีการออกแบบนี้เรียกว่าวิธี โครงเปรียบเทียบ (Equivalent frame method) ซึ่งถือว่าเป็นการวิเคราะห์โครงข้อแข็งที่มีความต่อเนื่อง ดังนั้นพื้นไร้คานต้องมีความต่อเนื่องอย่างน้อยสามช่วงติดต่อกันในแต่ละทิศทาง ความยาวของเสาที่ต่อเนื่องกันในแต่ละทิศทางจะต่างกันได้ไม่เกินร้อยละ 33 ตาม ACI 13.6.1 และตำแหน่งเสาอาจเอียงกันได้ไม่เกิน 10% ของระยะช่วงในทิศทางที่เอียงนั้น

2.7 ข้อกำหนดความปลอดภัย

แฟกเตอร์สำหรับการรับน้ำหนักประลัย, W_u

ตามมาตรฐาน ว.ส.ท (ว.ส.ท. 7203 ,72-1)

$$W_u = 1.7 DL + 2.0 LL$$

ตามมาตรฐาน ACI (ACI - 9.2.2)

$$W_u = 1.4 DL + 1.7 LL$$

เมื่อ

W_u = น้ำหนักที่ใช้ในการออกแบบ

DL = น้ำหนักบรรทุกคงที่ภายใต้สภาวะใช้งาน

LL = น้ำหนักบรรทุกจรภายใต้สภาวะใช้งาน

แฟกเตอร์ลดสำหรับการรับกำลังประลัย , ϕ

สำหรับการดัด $\phi = 0.90$ (ACI 9-3)

สำหรับแรงเฉือนและแรงบิด $\phi = 0.85$ (ACI 9-3)



บทที่ 3

ขั้นตอนการออกแบบ

3.1 การหาความหนาของแผ่นพื้น

ขนาดความหนาต่ำสุดสำหรับการควบคุมการทรุดตัว (deflection) ที่ได้จาก ACI - 9.5.3.2 , 9.5.3.3 ซึ่งได้แปลงค่าเป็นหน่วย SI แล้ว เป็นดังนี้

ACI (9-10)

$$h = \frac{\text{Ln} (0.8 + 2.844f_y / 40,000)}{36 + 5\alpha_m(\text{Ln}/S_n) - 0.6(1+\text{Ln}/S_n)}$$

ACI (9-11)

$$h = \frac{\text{Ln} (0.8 + 2.844f_y / 40,000)}{36 + 9\text{Ln}/S_n}$$

ACI (9-12)

$$h = \frac{\text{Ln} (0.8 + 2.844f_y / 40,000)}{36}$$

การพิจารณาความหนาของแผ่นพื้น ใช้สมการ 9-12 เป็นสมการควบคุม โดยที่

h = ความหนาของแผ่นพื้น

Ln = ความยาวของแผ่นพื้นประสิทธิภาพระหว่างเสาถึงเสาในแต่ละทิศทาง (ไม่นับความกว้างของเสา, หัวเสา)

S_n = ความกว้างของแผ่นพื้นที่มีระยะระหว่างแนวกึ่งกลางของแผ่นพื้นที่อยู่ติดกับแนวเสาที่กำลังพิจารณา (ไม่นับความกว้างของเสา, หัวเสา)

f_y = กำลังคลากของเหล็กเสริม (yield strength)

α_m = อัตราส่วนระหว่าง $E_b I_b$ และ $E_s I_s$

E_b = โมดูลัสยืดหยุ่นของคาน

E_s = โมดูลัสยืดหยุ่นของพื้น

I_b = โมเมนต์อินเนอร์เซียของคาน

I_s = โมเมนต์อินเนอร์เซียของพื้น

(ที่มีความกว้างของพื้นจากแนวเส้นศูนย์กลางของแผ่นพื้นที่ติดกับคานนั้น)

ความหนาของแผ่นพื้นต้องมีความหนาต่ำสุดไม่น้อยกว่า
 12.5 ซม. สำหรับแผ่นพื้นที่ไม่มีเป็นร่องรับหัวเสา
 10.0 ซม. สำหรับแผ่นพื้นที่มีเป็นร่องรับหัวเสา

3.2 การตรวจสอบแรงเฉือน (ACI - 11.12)

3.2.1 ลักษณะคานกว้าง (Wide - Beam Action)

แผ่นพื้นจะถูกนำมาพิจารณาเป็นคานกว้าง ที่มีความกว้างเท่ากับแผ่นพื้นที่ถูกตัดมาพิจารณาหน้าตัดของแผ่นพื้นที่ใช้ในการตรวจสอบแรงเฉือน จะอยู่ห่างจากขอบเสาหรือเป็นร่องรับ (drop panel) เป็นระยะเท่ากับความลึกประสิทธิภาพ (d) โดยมีน้ำหนักที่กระทำเท่ากับน้ำหนักที่ใช้ในการออกแบบคานพื้นที่จากหน้าตัดที่ตรวจสอบถึงกึ่งกลางระหว่างเสา

ซึ่งจะต้องมีค่าน้อยกว่าสมการดังต่อไปนี้

จาก ACI - 11.3.2.1 ได้

$$V_c = 0.53 \sqrt{f_c'} \cdot bw \cdot d \quad (\text{kg})$$

$$V_u = W_u \cdot bw \cdot L_f \quad (\text{kg})$$

โดยที่ $\phi V_c > V_u$

$$\phi = 0.85 \text{ สำหรับแรงเฉือน (ACI 9-3)}$$

$$V_c = \text{น้ำหนักที่คอนกรีตต้านทานได้ (กก.)}$$

$$V_u = \text{น้ำหนักที่กระทำกับหน้าตัดที่ตรวจสอบ (กก.)}$$

$$W_u = \text{น้ำหนักที่ใช้ในการออกแบบ (กก./ตร.ม.)}$$

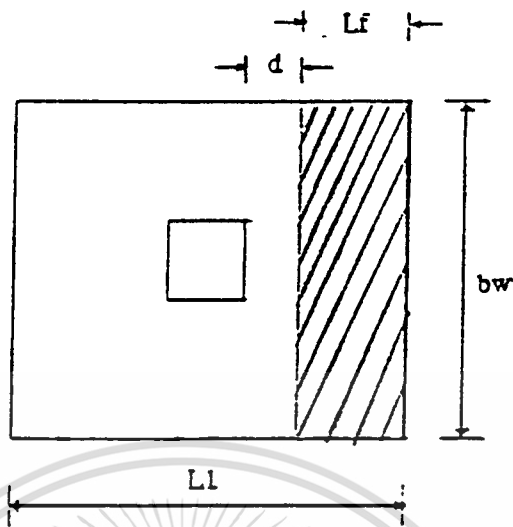
$$bw = \text{ความกว้างของแถบแผ่นพื้นที่ออกแบบ (ม.)}$$

$$L_f = \text{ความยาวจากหน้าตัดที่ตรวจสอบถึงกึ่งกลางระหว่างเสา (ม.)}$$

$$f_c' = \text{กำลังอัดประลัยของทรงกระบอกมาตรฐานที่ 28 วัน (กก./ซม.²)}$$

(Ultimate compressive strength)

$$d = \text{ความลึกประสิทธิภาพ (ซม.)}$$



รูปที่ 3.1 พื้นที่รับแรงเฉือนลักษณะคานกว้าง

3.2.2 ลักษณะเฉือนทะลุ (Two-way Action)

จะพิจารณาจากน้ำหนักของแผ่นพื้นที่กระทำต่อเสา หรือหัวเสาหรือแป้นรองรับหัวเสา หน้าตัดวิกฤติที่ทำการพิจารณาจะเป็นเส้นรอบรูปรอบที่รองรับ โดยมีระยะห่างจากขอบที่รองรับที่พิจารณาเป็นระยะ $d/2$ โดย d เท่ากับความลึกประสิทธิภาพตาม ACI-11.12.1.2

โดยมีสมการที่ใช้ในการตรวจสอบดังนี้

$$V_u = W_u \cdot (L1 \cdot L2 - A_i)$$

โดยที่ $\phi V_c > V_u$

V_u = น้ำหนักที่กระทำกับหน้าตัดที่ตรวจสอบ (ระยะ $d/2$ จากขอบเสาหรือแป้นรองรับ)

ϕ = 0.85 สำหรับแรงเฉือน

V_c = น้ำหนักที่ใช้ในการตรวจสอบ (กก.)

W_u = น้ำหนักที่ใช้ในการออกแบบ (กก. / ตร.ม.)

$L1$ = ความกว้างขอบแถบแผ่นพื้นที่ออกแบบ (ม.)

$L2$ = ความยาวจากหน้าตัดที่ตรวจสอบถึงกึ่งกลางระหว่างเสา (ม.)

A_i = พื้นที่ภายในหน้าตัดที่ทำการตรวจสอบ (ตร.ม.)

สำหรับค่าที่กำหนดใช้ในการตรวจสอบหาได้ดังนี้ (ACI - 11.12.2.1)

จาก ACI - 11.36

$$V_c = 0.265 (2 + 4/B_s) \sqrt{f_c'} b_o d$$

จาก ACI - 11.37

$$V_c = 0.265 (\alpha_s d/b_o + 2) \sqrt{f_c'} b_o d$$

จาก ACI - 11.38

$$V_c = 1.06 \sqrt{f_c'} b_o d$$

โดยที่

b_o = เส้นรอบรูปวงกุดรูปสี่เหลี่ยม (ซม.)

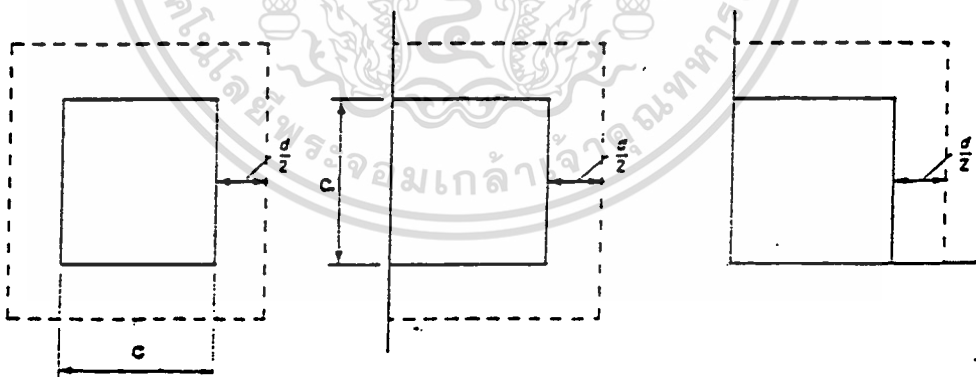
B_s = อัตราส่วนระหว่างด้านยาวกับด้านสั้น

α_s = 40 สำหรับเสาด้านใน

30 สำหรับเสาด้านนอก

20 สำหรับเสาที่มุม

โดยให้ใช้ค่า V_c ที่น้อยที่สุดจากสมการทั้งสาม



$$b_o = 4(c + d)$$

ก. กรณีสาด้านใน

$$b_o = 3c + 2d$$

ข. กรณีสาด้านนอก

$$b_o = 2c + d$$

ค. กรณีสาด้านมุม

รูปที่ 3.2 พื้นที่รับแรงเฉือนลักษณะเฉือนทะลุ

3.3 การวิเคราะห์หาค่าโมเมนต์

การวิเคราะห์โมเมนต์ใช้วิธีการกระจายโมเมนต์ (Moment Distribution) ซึ่งเป็น “ ยึดจุดต่อ ” (lock joint) และ “ ปลดจุดต่อ ” (unlock joint) สลับกันไปทุกๆ joint ในโครงสร้าง เริ่มแรกยึดจุดต่อทุกจุดไม่ให้หมุนทำให้โครงสร้างแต่ละส่วนมีลักษณะเหมือนคานที่มีปลายยึดแน่นทั้งสองข้าง (fixed end beam) ต่อไปก็ปลดจุดต่อเพียงจุดเดียวโมเมนต์ด้านที่เกิดขึ้นตรงปลายของทุกส่วนของโครงสร้างใกล้จุดต่อนี้ จะมีค่าแตกต่างกันตามส่วนความแข็งแรงของแต่ละชิ้นส่วนหรือตามค่าของ distribution factor ของแต่ละชิ้นส่วนในขณะเดียวกันโมเมนต์ด้านเหล่านี้จะถ่ายโมเมนต์ไปยังปลายตรงข้ามของชิ้นส่วน ซึ่งการถ่ายให้นี้ปลายตรงข้ามจะได้รับค่าเท่าไรก็ขึ้นอยู่กับค่า carry over factor ของชิ้นส่วนนั้น การยึดและปลดจุดต่อในโครงสร้างจะกระทำสลับกันไป เพื่อถ่ายโมเมนต์ให้จุดต่ออื่นๆ ที่อยู่ข้างเคียงจนได้ผลลัพธ์ตามความละเอียดที่ต้องการ

การวิเคราะห์โมเมนต์โดยวิธีการกระจายโมเมนต์จะต้องใช้ค่า fixed end moment, distribution factor, และค่า carry over factor ซึ่งตัวแปรดังกล่าวสามารถวิเคราะห์โดยใช้สมการจากวิธินุมลาคและการโก่ง ดังนี้

จากสมการหาอนุมลาค

$$\theta = \int_a^b (M_x / EI) dx$$

จากสมการหาการโก่งตัว

$$\Delta_A = \int_a^b (M_x / EI) x dx$$

$$\Delta_B = \int_a^b (M_x / EI) (l_1 - x) dx$$

จะได้

$$\theta_1 = \frac{F(M_A)}{EI_1 l_1} \int_a^b (l_1 - x) dx$$

$$= \frac{l_1 M_A}{EI_1} [C_1]$$

$$\text{เมื่อ } C_1 = [-F_B a(2-a)(1-F_A) - b(2-b)(1-F_B) + r(2-r)(1-F_R) - s(2-s)(1-F_R) \\ + c(2-c)(1-F_C) + F_C d(2-d)(1-F_D) + F_C F_D] / 2$$

$$\theta_b = \frac{F(M_B)}{EI_1 l_1} \int_A^B x dx$$

$$= \frac{l_1 M_B}{EI_1} [C_2]$$

$$\text{เมื่อ } C_2 = [-F_B a^2(1-F_A) - b^2(1-F_B) + r^2(1-F_R) - s^2(1-F_R) + c^2(1-F_C) + F_C d^2(1-F_D) \\ + F_C F_D] / 2$$

$$(\Delta_1)_A = \frac{F(M_A)}{EI_1 l_1} \int_A^B (l_1 - x) x dx$$

$$= \frac{l_1^2 M_A}{EI_1} [C_3]$$

$$\text{เมื่อ } C_3 = [-F_B a^2(3-2a)(1-F_A) - b^2(3-2b)(1-F_B) + r^2(3-2r)(1-F_R) - s^2(3-2s) \\ (1-F_R) + c^2(3-2c)(1-F_C) + F_C d^2(3-2d)(1-F_D) + F_C F_D] / 6$$

$$(\Delta_b)_A = \frac{F(M_B)}{EI_1 l_1} \int_A^B x^2 dx$$

$$= \frac{l_1^3 M_B}{EI_1} [C_4]$$

$$\text{เมื่อ } C_4 = [-F_B a^3(1-F_A) - b^3(1-F_B) + r^3(1-F_R) - s^3(1-F_R) + c^3(1-F_C) + F_C d^3(1-F_D) \\ + F_C F_D] / 3$$

$$(\Delta_C)_A = \frac{F(w)}{2EI_1} \int_A^B (l_1 - x) x^2 dx$$

$$= \frac{l_1^4 w}{EI_1} [C_5]$$

$$\text{เมื่อ } C_3 = [-F_B a^3 (4-3a)(1-F_A) - b^3 (4-3b)(1-F_B) + r^3 (4-3r)(1-F_R) - s^3 (4-3s)(1-F_S) + c^3 (4-3c)(1-F_C) + F_D^3 (4-3d)(1-F_D) + F_C F_D] / 24$$

$$\begin{aligned} (\Delta_v)_B &= \frac{F(M_A)}{EI_1 l_1} \int_A^B (l_1 - x)^2 dx \\ &= \frac{l_1^2 M_A [C_4]}{EI_1} \end{aligned}$$

$$\text{เมื่อ } C_6 = [-F_B a (3-3a+a^2)(1-F_A) - b(3-3b+b^2)(1-F_B) + r(3-3r+r^2)(1-F_R) - s(3-3s+s^2)(1-F_S) + c(3-3c+c^2)(1-F_C) + F_D(3-3d+d^2)(1-F_D) + F_C F_D] / 3$$

$$\begin{aligned} (\Delta_v)_B &= \frac{F(M_B)}{2EI_1 l_1} \int_A^B (l_1 - x) x dx \\ &= \frac{l_1^2 M_B [C_3]}{EI_1} \end{aligned}$$

$$\begin{aligned} (\Delta_c)_B &= \frac{F(W)}{2EI_1} \int_A^B (l_1 - x)^2 dx \\ &= \frac{l_1^4 W [C_7]}{EI_1} \end{aligned}$$

$$\text{เมื่อ } C_7 = [-F_B a^2 (6-8a+3a^2)(1-F_A) - b^2 (6-8b+3b^2)(1-F_B) + r^2 (6-8r+r^2)(1-F_R) - s^2 (6-8s+3s^2)(1-F_S) + c^2 (6-8c+c^2)(1-F_C) + F_D^2 (6-8d+3d^2)(1-F_D) + F_C F_D] / 24$$

ค่า carryover factors คือ

$$COF_{AB} = C_3 / C_4$$

$$COF_{BA} = C_3 / C_6$$

ค่า stiffness factors คือ

$$k_{AB} = \frac{C_4}{C_4 C_1 - C_3 C_2}$$

$$k_{BA} = \frac{C_6}{C_6 C_2 - C_3 C_1}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้า fixed - end moment coefficients คือ

$$FEMc_A = \frac{C_7C_4 - C_3C_7}{C_3C_4 - C_7C_3}$$

$$FEMc_B = \frac{C_8C_5 - C_7C_3}{C_3C_4 - C_7C_3}$$

โดยที่

$$F_A = (1 - C_{2A}/r_2)^2$$

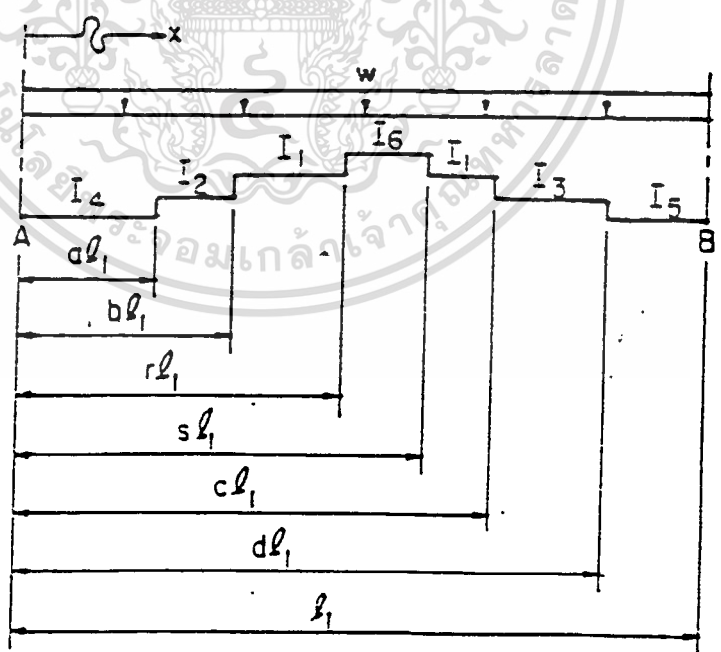
$$F_B = I_1/I_2$$

$$F_C = I_1/I_3$$

$$F_D = I_1/I_5$$

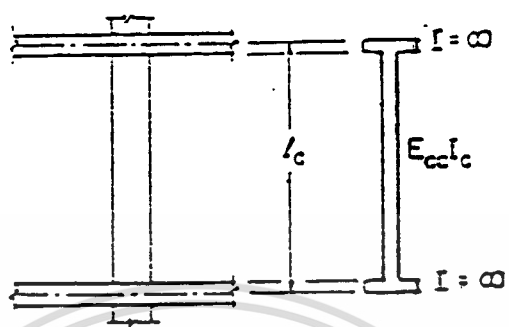
$$F_D = (1 - C_{2B}/r_2)^2$$

สัมประสิทธิ์ a,b,r,s,c,d ที่ทราบจากแผนภาพต่อไปนี้

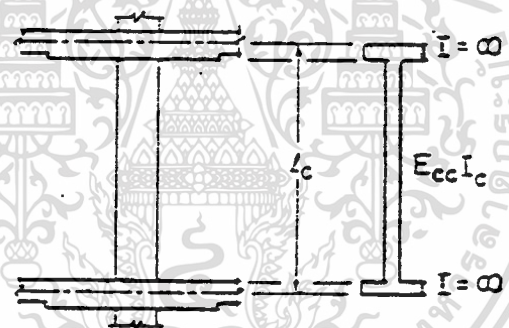


รูปที่ 3.3 แผนภาพแสดงสัมประสิทธิ์ a, b, r, s, c, d

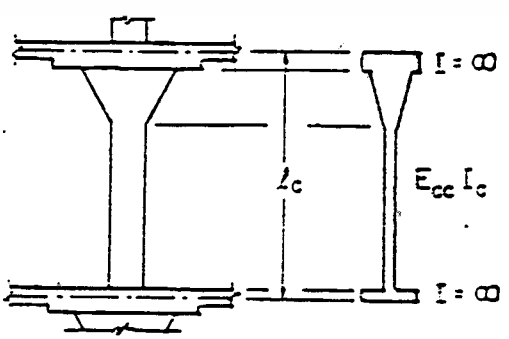
รูปที่ 3.4 แผนภาพประกอบการหาค่าสัมประสิทธิ์ และตัวแปรของเสา



3.4.1 กรณีไม่มีแป้นรองรับหัวเสาและหัวเสา



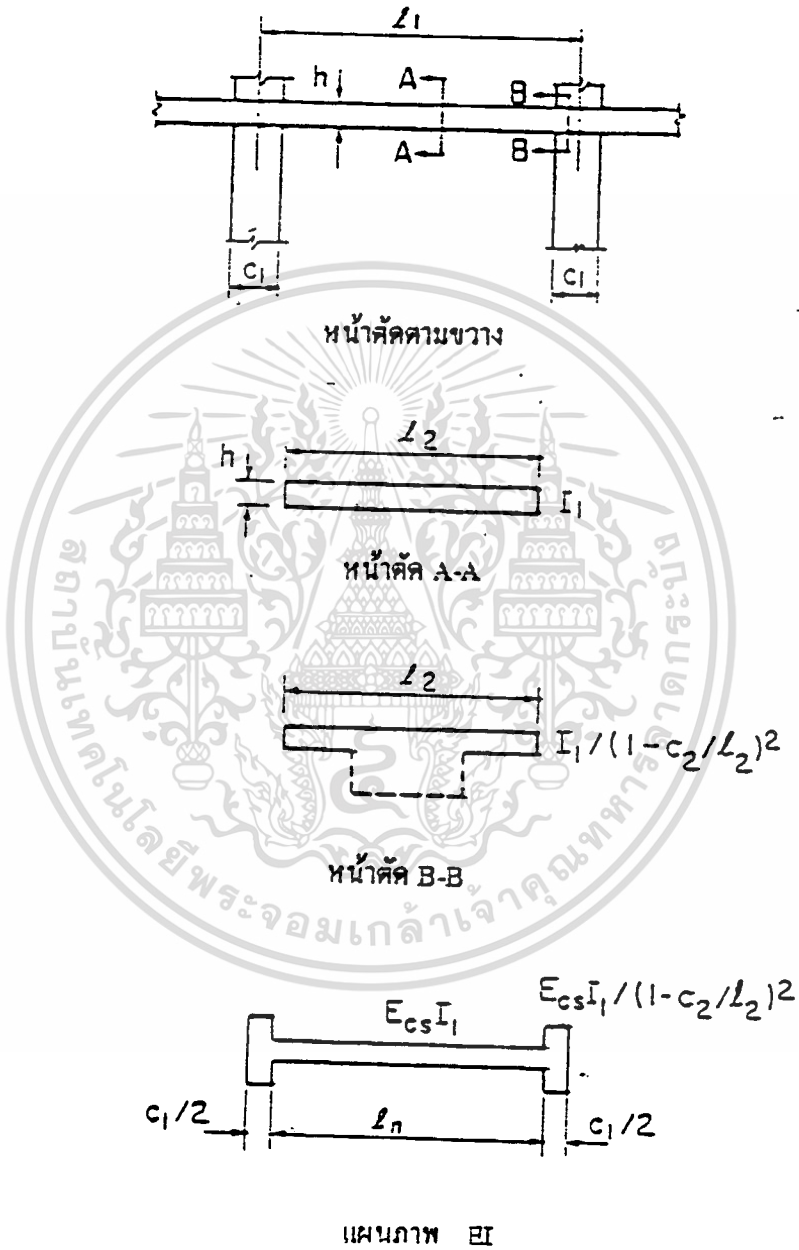
3.4.2 กรณีมีแป้นรองรับหัวเสาแต่ไม่มีหัวเสา



3.4.3 กรณีมีแป้นรองรับหัวเสาและหัวเสา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

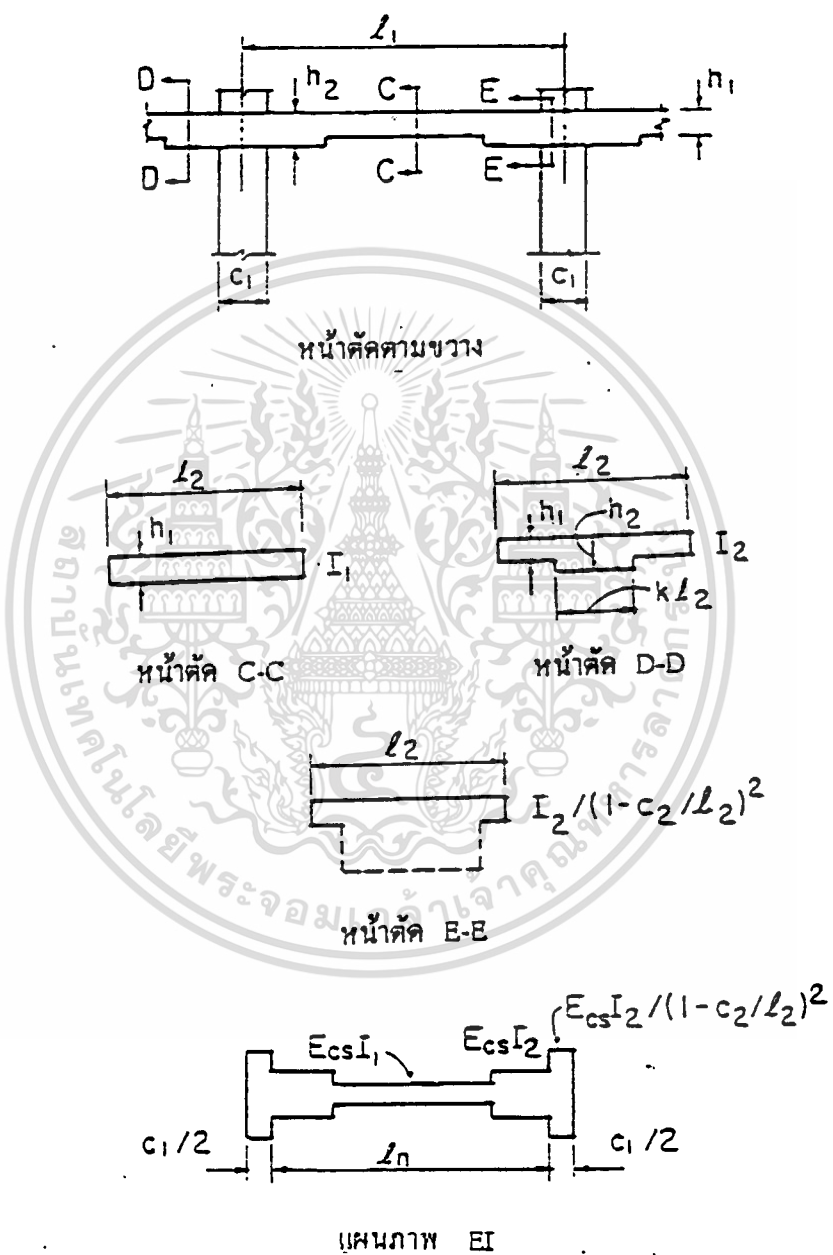
รูปที่ 3.5 แผนภาพประกอบการหาค่าสัมประสิทธิ์และตัวแปรของแผ่นพื้น



3.5.1 กรณีไม่มีแป้นรองรับหัวเสาและหัวเสา

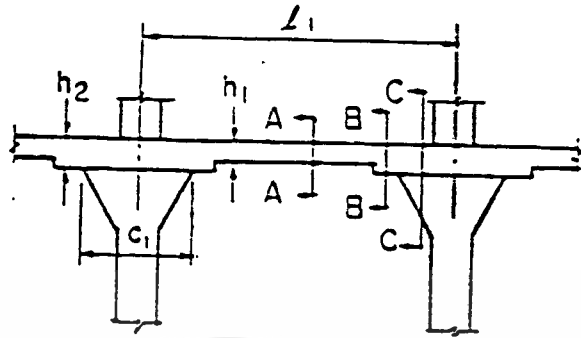


รูปที่ 3.5 แผนภาพประกอบการหาค่าสัมประสิทธิ์และตัวแปรของแผ่นพื้น

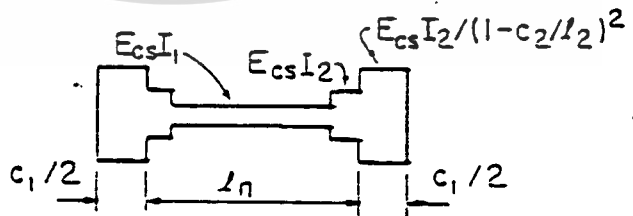
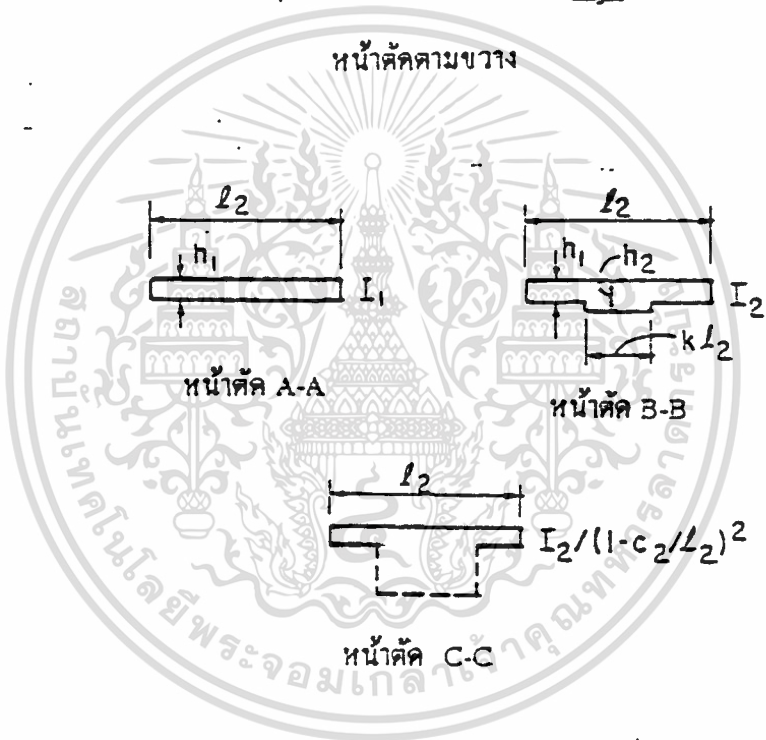


3.5.2 กรณีมีแป้นรองรับหัวเสา แต่ไม่มีหัวเสา

รูปที่ 3.5 แผนภาพประกอบการศึกษาหาค่าสัมประสิทธิ์และตัวแปรของพื้น



หน้าตัดตามขวาง



แผนภาพ EI

3.5.3 กรณีมีแป้นรองรับหัวเสาและหัวเสา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1 การหาค่า fixed end moment

จากสมการ $FEM = FEMc \cdot W_u \cdot L_2 \cdot L_1^2$

$FEMc$ เป็นค่าสัมประสิทธิ์ที่ได้จากการวิเคราะห์ในขั้นตอนที่ผ่านมา

W_u เป็นน้ำหนักที่ใช้ในการออกแบบ

โดยพิจารณาตาม load pattern (ACI 13.7.6.1) ที่มีอยู่ 5 แบบดังรูป 3.6.1 - 3.6.5

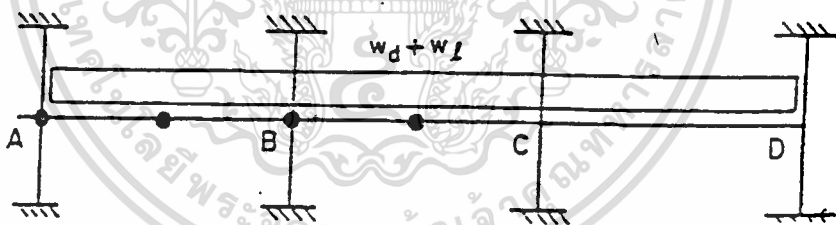
โดยทำการอธิบายเพียงแค่ 3 ช่วง

ในกรณีที่อัตราส่วนระหว่างน้ำหนักจรกับน้ำหนักคงที่มีค่าไม่เกิน 0.75

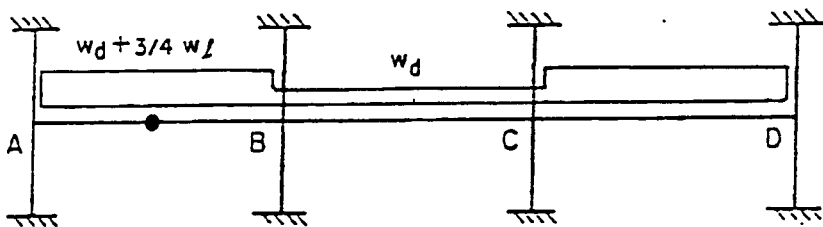
ให้ใช้เพียงแค่ load pattern 1 เท่านั้น (ACI 13.7.6.2)

ส่วนในกรณีที่มากกว่า 0.75 ให้ใช้น้ำหนักที่ได้ในแต่ละแบบจากแนวทางของ influenceline (ACI 13.7.6.3) ตามรูปที่ 3.6.1 - 3.6.5 โดยใช้ค่าโมเมนต์ที่มากที่สุดจาก pattern ที่มี (ACI 13.7.6.4) และกรณีที่ทราบน้ำหนักกระทำที่แน่นอนสามารถพิจารณาเพียงแค่ load pattern 1-3 เท่านั้น

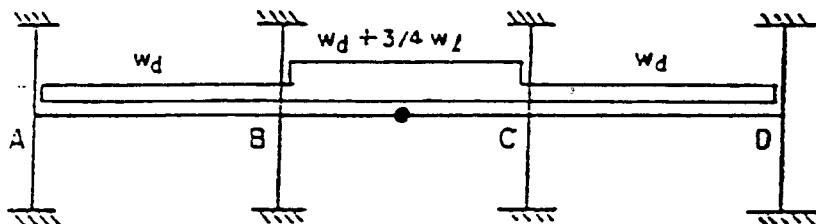
ทั้งนี้อัตราส่วนระหว่างน้ำหนักจรกับน้ำหนักคงที่ ต้องไม่เกินว่า 3 เพื่อป้องกันการแตกร้าวของพื้น (ACI 13.6.1)



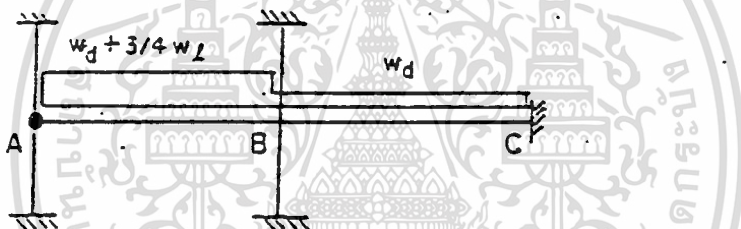
รูปที่ 3.6.1 load pattern 1 สำหรับการออกแบบโมเมนต์บวกและลบทุกช่วง



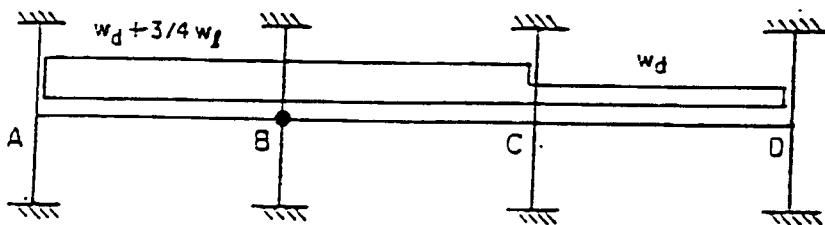
รูปที่ 3.6.2 load pattern 2 สำหรับการหาค่าโมเมนต์บวกในช่วง AB, CD



3.6.3 load pattern 3 สำหรับการหาค่าโมเมนต์บวกในช่วง BC



3.6.4 load pattern 4 สำหรับการหาค่าโมเมนต์ลบที่ ที่รองรับ A โดยทำการพิจารณาแค่ 2 ช่วงโดยให้สมมุติว่าจุด C เป็นจุดยึดแน่น เหมือนปลายเสาบน และ ล่าง



3.6.5 load pattern 5 สำหรับการหาค่าโมเมนต์ลบที่ ที่รองรับ B

3.3.2 การหาค่าตัวแปรการกระจายโมเมนต์, DF มีขั้นตอนดังนี้

1. การหาค่าความแข็งเกร็งของเสา, K_c

พิจารณาจากสมการ

$$K_c = k_{cc} E_{cc} I_c / l_c$$

โดยที่

k_{cc} คือค่าสัมประสิทธิ์ที่ได้จากวิเคราะห์ในหัวข้อ 3.3

I_c คือค่าโมเมนต์อินเนอร์เซียของเสา

E_{cc} คือค่าโมดูลัสแห่งความยืดหยุ่นของคอนกรีตของเสา

l_c คือความสูงแต่ละชั้นโดยวัดจากกึ่งกลางของพื้น

2. การหาค่าความแข็งเกร็งบิด (torsion stiffness), K_t

พิจารณาจากสมการ

$$K_t = 9 E_{cs} C / l_2 [1 - (c_2 / l_2)]^3$$

โดย

E_{cs} คือค่าโมดูลัสแห่งความยืดหยุ่นของคอนกรีตของพื้น

$$C = \sum [1 - 0.63(x/y)] x^3 y / 3$$

โดยที่

x = ขนาดหน้าตัดด้านสั้น

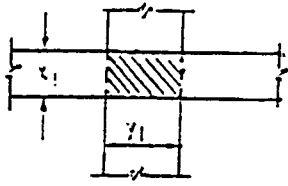
y = ขนาดหน้าตัดด้านยาว

โดยพิจารณาหน้าตัดที่รับแรงบิดที่เกิดขึ้น ดังในรูป 3.7

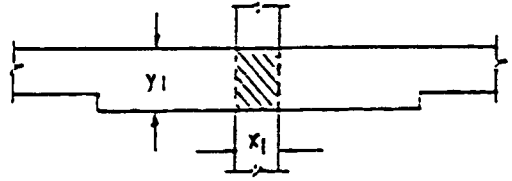
สำหรับกรณีที่หน้าตัดประกอบไปด้วยหน้าตัดรูปสี่เหลี่ยมผืนผ้าหลายรูป

ให้ทำการหาค่า C หลายๆแบบ แล้วใช้ค่าที่มากที่สุดมาใช้ในการหาค่า K_t ต่อไป

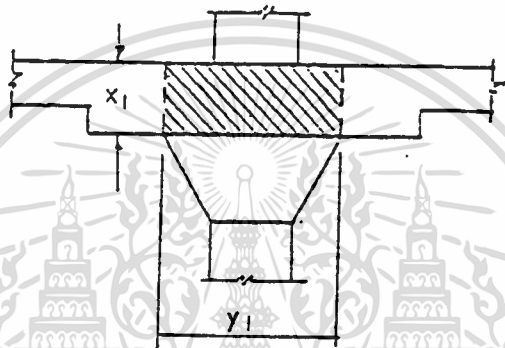
รูปที่ 3.7 หน้าตัดในการหาค่า C



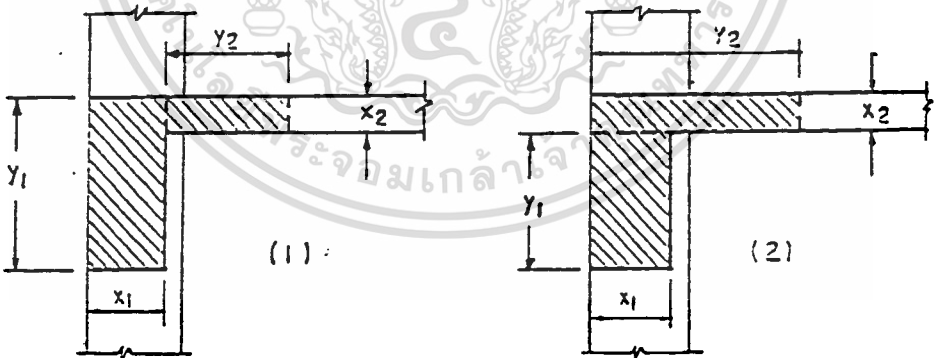
กรณีมีเฉพาะแผ่นพื้น



กรณีมีเป็นรองรับ



กรณีมีเป็นรองรับและหัวเสา



กรณีคานขอบ ใช้ค่ามากที่สุดจากทั้ง 2 รูป

3. หาค่าความแข็งเกร็งของเสาเปรียบเทียบ , K_{ec}

หลังจากวิเคราะห์ค่าความแข็งเกร็งของเสา, K_c และความแข็งเกร็งบิด, K_t ที่บริเวณรอยต่อระหว่างเสาและแผ่นพื้นแล้ว นำค่าที่หาได้มารวมเข้าด้วยกันดังต่อไปนี้

$$\frac{1}{K_{ec}} = \frac{1}{\sum K_c} + \frac{1}{\sum K_t}$$

หรือจะได้ว่า

$$K_{ec} = \frac{(\sum K_c \cdot \sum K_t)}{(\sum K_c + \sum K_t)}$$

4. หาค่าความแข็งเกร็งของแผ่นพื้น, K_s

พิจารณาสมการ

$$K_s = k_{ss} E_{cc} I_s / I_1$$

โดย

k_{ss} คือค่าสัมประสิทธิ์ที่ได้จากการวิเคราะห์ในหัวข้อ 3.3

I_s คือค่าโมเมนต์อินเนอร์เซียของแผ่นพื้น

$$= I_2 h^3 / 12$$

I_1 คือความกว้างของแถบพื้นที่พิจารณาจากศูนย์กลางเสาถึงศูนย์กลางเสา

I_2 คือความกว้างของแถบพื้นที่พิจารณาในแนวขวาง

จากขั้นตอนที่ 1-4 ที่ผ่านมาจะได้ค่าความความแข็งเกร็งของแผ่นพื้น และความแข็งเกร็งของเสาเปรียบเทียบ ซึ่งจะนำมาใช้ในการหาค่า DF เพื่อใช้ในการกระจายค่าโมเมนต์ที่เกิดขึ้นในแต่ละ node

ค่า DF ได้จากสมการต่อไปนี้

$$DF \text{ ที่จุดรองรับภายนอก} = \frac{K_s}{K_s + \sum K_{ec}}$$

$$DF \text{ ที่จุดรองรับภายใน} = \frac{K_s}{\sum K_s + \sum K_{ec}}$$

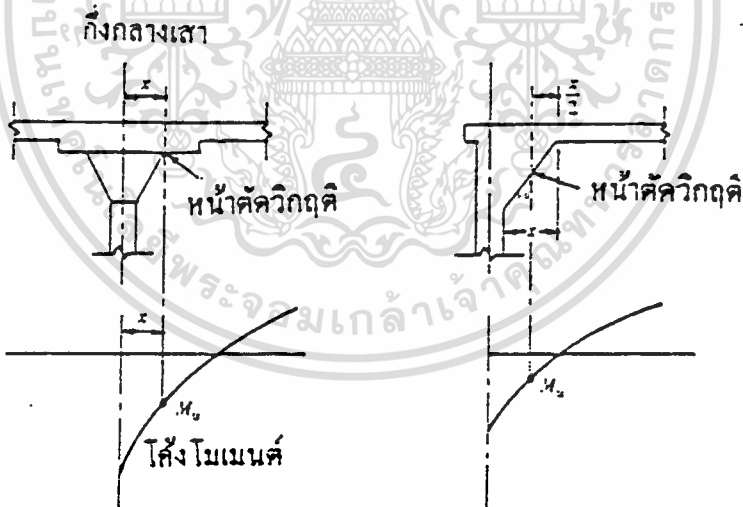
3.3.3 การหาค่า carry over factor, COF

Carry over factor คือตัวแปรที่ใช้ในการส่งถ่ายโมเมนต์จากปลายข้างหนึ่งไปยังอีกข้างหนึ่ง ซึ่งใช้ค่าที่ได้จากการวิเคราะห์ในขั้นตอนที่ผ่านมาในหัวข้อ 3.3

3.4 การหาโมเมนต์ในการออกแบบ

3.4.1 การหาค่าโมเมนต์ลบ

ค่าโมเมนต์ลบที่ได้จากการกระจายโมเมนต์นั้นเป็นค่าที่กึ่งกลางของเสาแต่จากข้อกำหนดของ ACI 13.7.7.1 ค่าโมเมนต์ที่นำมาใช้ในการออกแบบสามารถใช้โมเมนต์ที่เกิดขึ้นที่ขอบเสาหรือขอบหัวเสาได้ แต่ระยะดังกล่าวต้องไม่มากกว่า 0.175 เท่าของช่วงเสา และถ้าเสาด้านนอกมีหัวเสาให้ใช้ได้เพียงครั้งหนึ่งของระยะระหว่างขอบหัวเสาและขอบเสา ซึ่งเรียกว่า "ระยะวิกฤตสำหรับการหาโมเมนต์" ดังในรูป



รูปที่ 3.8 แสดงระยะวิกฤตสำหรับการหาโมเมนต์

สมการที่ใช้ในการหาค่าโมเมนต์ที่ใช้ในการออกแบบ เขียนได้ดังนี้

$$M = M_o + Vx - W_u \cdot L_2 \cdot x^2 / 2$$

โดยที่

M_o = โมเมนต์ลบที่เกิดขึ้นที่กึ่งกลางเสามีค่าเป็นลบเสมอ

V = แรงเฉือนเปรียบเทียบเป็นผลรวมของแรงเฉือนปกติบวกกับแรงเฉือนอันเกิดจากโมเมนต์ที่กระทำต่อแผ่นพื้นช่วงนั้น

x = ระยะห่างจากกึ่งกลางเสาถึงขอบเสาหรือขอบหัวเสาแต่ต้องมีค่าน้อยกว่า $0.175 l_1$ หรือระยะวิกฤติ

l_2 = แลบของแผ่นพื้นที่พิจารณา

3.4.2 การหาโมเมนต์บวกที่กึ่งกลางช่วงพื้น

จาก ACI-13.6.2 การหาโมเมนต์บวกที่กึ่งกลางช่วงพื้นให้ใช้สูตรดังต่อไปนี้

$$M \text{ ที่กึ่งกลาง} = M_s - (M_L + M_R) / 2$$

โดยที่

M_s คือค่าโมเมนต์ที่กึ่งกลางช่วงของ simple support

มีค่าเท่ากับ $W_u l_2^2 / 8$

M_L คือค่าโมเมนต์ลบที่กึ่งกลางเสาด้านซ้าย

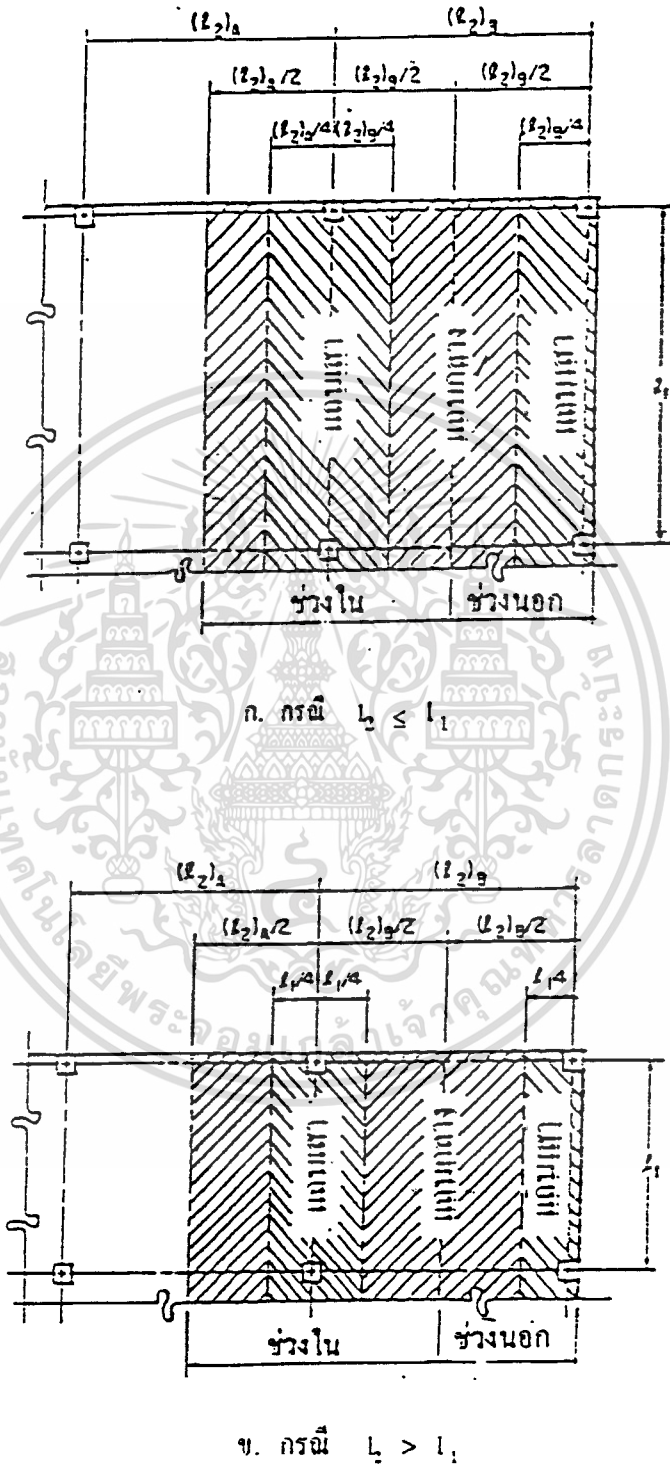
M_R คือค่าโมเมนต์ลบที่กึ่งกลางเสาด้านขวา

3.5 การกระจายโมเมนต์เข้าสู่แถบเสา และ แถบกลาง

โมเมนต์บวกและลบ ที่ได้จากการวิเคราะห์มาจนถึงลำดับนี้แล้ว ต้องมีการกระจายเข้าสู่แถบเสา(Column strip) และแถบกลาง(Middle strip) เพื่อนำมาคำนวณหาเหล็กเสริมตามข้อกำหนด ACI 13.7.7.6 ดังนี้

| พื้นที่ช่วงนอก | แถบเสา | แถบกลาง |
|--------------------|--------|---------|
| โมเมนต์ลบที่เสานอก | 100 % | |
| โมเมนต์ลบที่เสาใน | 75 % | 25 % |
| โมเมนต์บวก | 60 % | 40 % |
| พื้นที่ช่วงใน | | |
| โมเมนต์ลบ | 75 % | 25 % |
| โมเมนต์บวก | 60 % | 40 % |

ตารางที่ 3.1 การกระจายโมเมนต์เข้าสู่แถบเสาและแถบกลาง



รูปที่ 3.10 แสดงแถบเสา (column strip) และแถบล่าง (middle strip)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การหาปริมาณเหล็กเสริม

4.1 กรณีเหล็กเสริมรับแรงดึง

การออกแบบเหล็กเสริมในแผ่นพื้นไร้คาน มักจะมีการเสริมเหล็กเพื่อรับแรงดึงเพียงอย่างเดียวไม่นิยมออกแบบให้เสริมเหล็กรับแรงอัด ตามขั้นตอนต่อไปนี้

$$M_n = M_u / \phi$$

โดยที่ M_n คือ โมเมนต์ที่ใช้ในการหาเหล็กเสริม

M_u คือ โมเมนต์ที่ได้จากการวิเคราะห์

$$\phi = 0.90 \quad (\text{ACI-9.3})$$

จากผลที่สภาวะสมดุลย์

$$C = T$$

และ $M_n = (C \text{ หรือ } T)(d - a/2)$

แรงอัดรวมที่เกิดในคอนกรีต, C คือ

$$C = 0.85 f_c' b a$$

แรงดึงรวมในเหล็กเสริม, T คือ

$$T = A_s f_y = P b d f_y$$

จาก (4-2)

$$0.85 f_c' b a = P b d f_y$$

$$a = P \left(\frac{f_y}{0.85 f_c} \right) d$$

แทนค่า a ในสมการ จะได้

$$M_n = P b d f_y \left[d - \frac{P}{2} \left(\frac{f_y}{0.85 f_c} \right) d \right]$$

กำหนด R_v เป็นค่าสัมประสิทธิ์ต้านทาน (Coefficient of resistance)

มีค่าเท่ากับ $M_n / b d^2$

$$\text{ให้ } m = \frac{f_y}{0.85 f_c}$$

$$\text{ดังนั้น } R_u = \frac{M_n}{b d^2} = P f_y (1 - 1/2 P m)$$

$$\text{หรือ } P = \frac{1}{m} \left(1 - \sqrt{1 - \frac{2mRu}{f_y}} \right)$$

โดยค่า P คือ เปอร์เซนต์พื้นที่หน้าตัดเหล็กเสริมต่อพื้นที่หน้าตัดของแผ่นพื้น ซึ่งมีข้อกำหนดดังนี้

$$P_{\min} = 14/f_y$$

$$P_{\max} = 0.75 P_b$$

$$\text{โดยที่ } P_b = \frac{0.85 \beta_1 f_c'}{f_y} \left(\frac{6117}{6117 + f_y} \right)$$

โดยมี

$$\beta_1 = 0.85 \text{ สำหรับ } f_c' \leq 280 \text{ ksc}$$

$$\beta_1 = \frac{0.85 - 0.05 (f_c' - 280)}{70} \text{ ksc}$$

$$\text{สำหรับ } 280 \text{ ksc} < f_c' < 560 \text{ ksc}$$

$$\beta_1 = 0.65 \text{ สำหรับ } f_c' > 560$$

หาปริมาณเหล็กเสริม จากค่า P ที่หาได้

$$A_s = Pbd$$

โดยที่ค่า b คือ ความกว้างของแถบเสา สำหรับโมเมนต์ลบ

คือ ความกว้างของแถบกลางสำหรับโมเมนต์บวก

d คือ ระยะลึกประสิทธิผลของแผ่นพื้น

4.2 กรณีเหล็กเสริมรับแรงอัด

ในกรณีที่ค่าโมเมนต์ด้านของหน้าตัดที่คำนวณได้น้อยกว่าค่าโมเมนต์กระทำแสดงว่าหน้าตัดมีความจำเป็นต้องเพิ่มเหล็กรับแรงอัดเพื่อช่วยรับกำลังอัดบางส่วนที่เกินกว่าคอนกรีตจะรับไว้ได้ เหล็กเสริมรับแรงดึง, A_s และเหล็กเสริมรับแรงอัด, A_s' จะหาได้จากสมการของความสมดุลสองสมการ ดังนี้

$$C_c + C_s = T$$

$$\text{และ } M_n = C_c (d - \beta_1 X/2) + C_s (d - d')$$

โดยที่

$$T = P b d f_y$$

$$C_c = 0.85 f_c' \beta_1 X_b$$

$$C_s = (f_y - 0.85 f_c') P' b d$$

โดยตอนแรกสมมุติค่าให้ $f_s' = f_y$ ($\Delta's > \Delta y$)

ค่า X ที่ใช้ในการออกแบบทั่วไปประมาณ 0.375 X_b

$$X_b = \Delta c / (\Delta c + \Delta y) d$$

$$\Delta's = \Delta c (X - d') / X$$

เมื่อ

$$\Delta c = \text{ค่า strain ของคอนกรีต ปกติใช้ } 0.003$$

$$\Delta y = f_y / E_s$$

$$E_s = 2.04 \times 10^6 \text{ ksc}$$

$$d' = \text{ระยะระหว่างขอบเหล็กกับผิวคอนกรีตด้านบน}$$

$$\Delta's = \text{ค่า strain ของเหล็กรับแรงอัด}$$

ถ้าค่า $\Delta's < \Delta y$ ต้องพิจารณาค่า f_s' ใหม่โดย

$$f_s' = 6117 - (6117 + f_y) d' / d$$

จะได้เหล็กรับแรงดึง

$$A_s = P b d + A_s'$$

เหล็กรับแรงอัด

$$A_s' = \frac{M_u - \phi M_n}{\phi f_s' (d - d')}$$

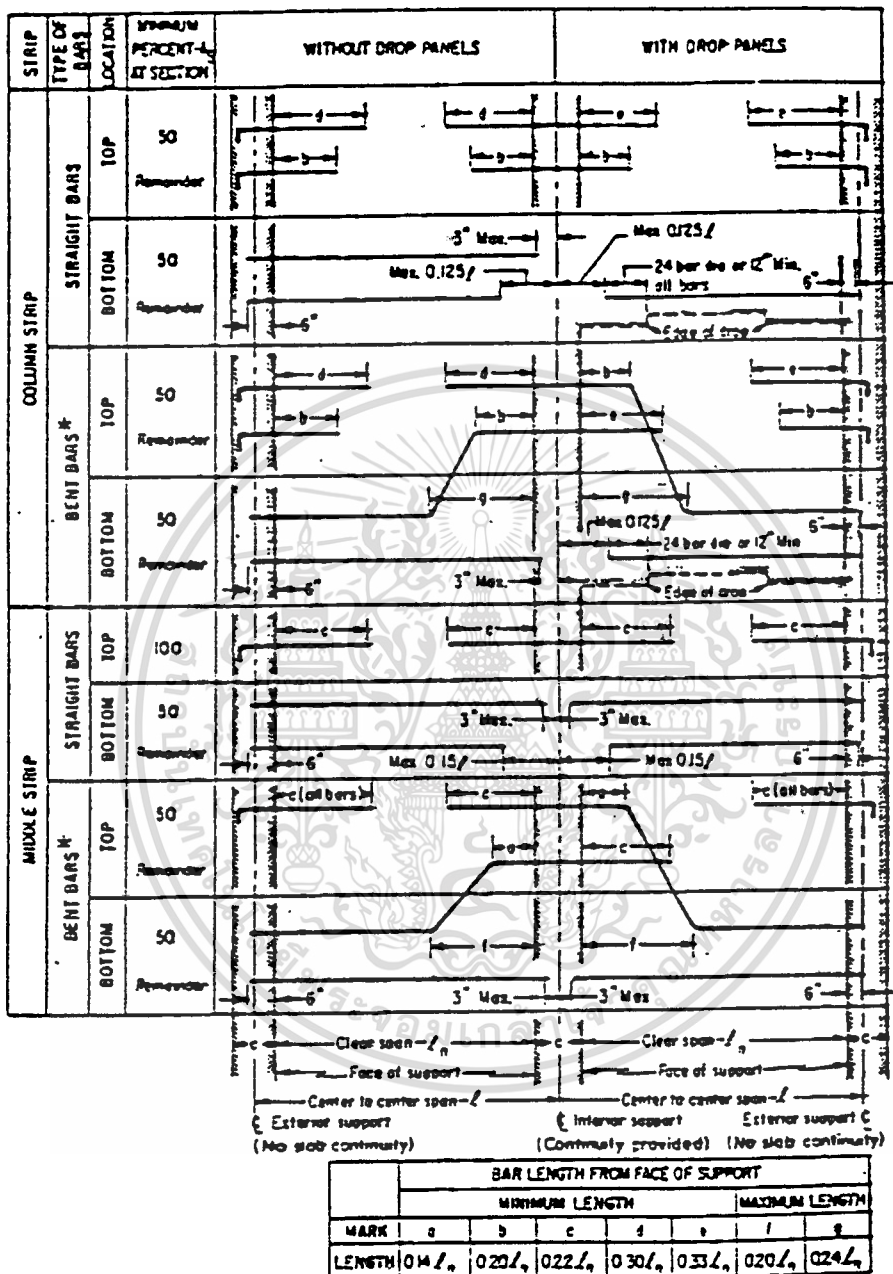
มาตรฐาน ว.ส.ท. ได้กำหนดเกณฑ์สำหรับเหล็กรับแรงอัด เพื่อความแน่ใจว่าแรงที่จุดประลัยมีค่าถึง f_y ของเหล็กเสริมดังนี้

$$P - P' > 0.85 \beta_1 \frac{f_c' d'}{f_y d} \left(\frac{6117}{6117 - f_y} \right)$$

โดยที่ค่า $(P - P')$ ต้องไม่เกิน 0.75 P_b

4.3 การจัดเหล็กเสริมในแผ่นพื้นไร้คาน ให้ปฏิบัติดังนี้

1. ปริมาณเหล็กเสริม, A_s ที่จัดหามาเสริมในแผ่นพื้นจะต้องเพียงพอต่อการรับกำลังที่ได้คำนวณไว้ และอย่างน้อยจะต้องมีปริมาณเพียงพอต่อการต้านการขีดครุดเนื่องจากอุณหภูมิ
2. ที่หน้าตัดวิกฤติ ระยะเรียงของเหล็กเสริมต้องไม่เกิน 2 เท่าของความหนาแผ่นพื้น
3. ระยะต่ำสุดของเหล็กเสริมจะมีกฎเกณฑ์คล้าย ๆ กับในแผ่นพื้นทางเดียวโดยเฉพาะเหล็กเสริมบวที่เสริมตั้งฉากกับขอบแผ่นพื้นด้านที่ไม่มีการต่อเนื่อง ควรจะยื่นเข้าไปในคานขอบ ผนังหรือเสาไม่ต่ำกว่า 15 ซม. และเหล็กเสริมลบที่เสริมตั้งฉากกับขอบก็มีการจัดเหล็กในลักษณะเดียวกัน
4. ควรจัดระยะเรียงของเหล็กเสริมตลอดแถบของช่วงพื้นให้ห่างสม่ำเสมอกันตลอด ยกเว้น กรณีต่อไปนี้
 - ก. จะต้องจัดให้มีเหล็กเสริมจำนวนไม่น้อยกว่า 1/4 ของเหล็กเสริมลบในแถบเสา ผ่านระยะเส้นรอบรูป ซึ่งอยู่ห่างจากขอบเสาหรือขอบหัวเสา เป็นระยะ d
 - ข. ในกรณีที่มีแป้นหัวเสา อย่างน้อยครึ่งหนึ่งของเหล็กเสริมลบในแถบเสา จะต้องผ่านระยะแป้นนั้น
 - ค. สำหรับเหล็กเสริมที่เหลือในแถบเสาจากข้อ ก. หรือ ข. ให้จัดระยะเรียงลดหลั่นกันจนเท่ากับระยะเรียงเหล็กเสริมในแถบกลาง



รูปที่ 4.1 แสดงระยะทาบ และ ระยะการเสริมเหล็ก

ตอนที่ 2

คู่มือการใช้โปรแกรม LFLATS
(Manual of LFLATS)



คู่มือการใช้โปรแกรมฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมการก่อสร้าง
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือประกอบการใช้โปรแกรม LFLATS

(Manual of LFLATS)

บทนำ

โปรแกรม LFLATS เป็นซอฟต์แวร์สำเร็จรูปสำหรับคำนวณและออกแบบแผ่นพื้นไร้คาน โดยพัฒนาโปรแกรมจากภาษา Visual BASIC Version 3.0 แล้วทำการ Compile เป็น Execute file (ไฟล์ที่มีนามสกุลเป็น exe .exe) ซึ่งสามารถทำงานได้เหมือน Execute file ทั่วไป โดยไม่จำเป็นต้องใช้โปรแกรม Visual BASIC

อย่างไรก็ตาม Execute file ที่พัฒนาจาก Visual BASIC จะต้องใช้ windows ในการทำงาน ดังนั้น เครื่องคอมพิวเตอร์ที่สามารถจะใช้โปรแกรมนี้ได้ จะต้องมีโปรแกรม windows ด้วย (ซึ่งเครื่องโดยทั่วไปจะมีโปรแกรม windows อยู่แล้ว) ในกรณีที่ต้องการปรับปรุงโปรแกรม จะต้องใช้ source code และใช้ Visual BASIC ตั้งแต่ Version 3.0 ขึ้นไปในการพัฒนาโปรแกรม

การออกแบบโปรแกรม LFLATS นี้เน้นการใช้งานง่ายแบบโปรแกรม windows ทั่วไป โปรแกรมเน้นความถูกต้องในการคำนวณ ผู้พัฒนาโปรแกรมได้พัฒนาโปรแกรมได้ทำการออกแบบโปรแกรม LFLATS ให้สามารถพิมพ์ผลการคำนวณ (summary) และรายละเอียดในการคำนวณ (Detail) ได้ เพื่อความสะดวกของผู้ใช้ทั่วไปในการออกแบบในประเทศไทย คือหน่วยเมตริก (Metric Units)

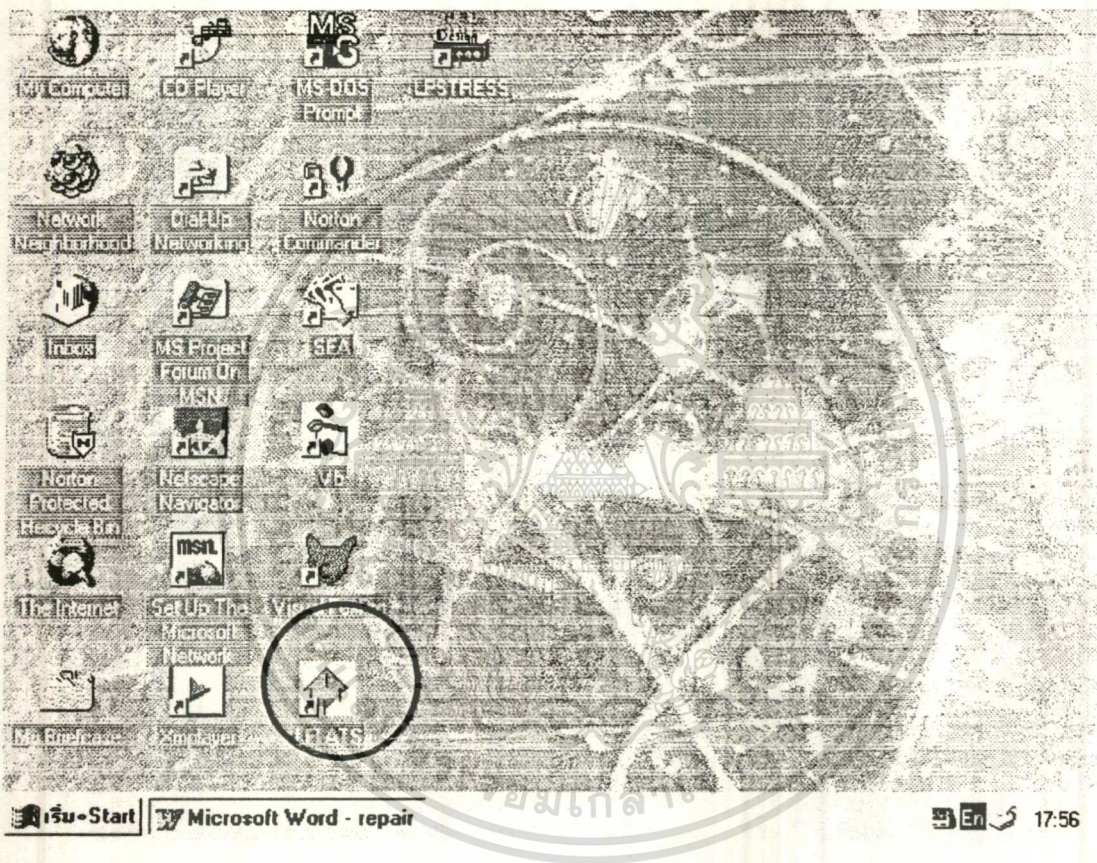
การใช้โปรแกรมในการออกแบบแผ่นพื้นไร้คาน นั้นผู้ใช้โปรแกรมควรมีความรู้ในการวิเคราะห์และออกแบบแผ่นพื้น จะทำให้สามารถใช้โปรแกรมได้ง่ายและเข้าใจโครงสร้างของโปรแกรมได้ดีขึ้น ฉะนั้นผู้ใช้โปรแกรมจึงควรที่จะศึกษาการออกแบบควบคู่ไปกับการเรียนรู้การใช้โปรแกรมจากคู่มือการใช้งานเล่มนี้

1. การเข้าสู่โปรแกรม

เนื่องจากโปรแกรม LFLATS เป็นโปรแกรมที่ run บนโปรแกรม windows ฉะนั้นการเข้าสู่โปรแกรมจึงต้องผ่าน windows

c:\win

การเข้าสู่โปรแกรมสามารถเข้าได้โดยใช้การใช้ Icon ที่ได้
ก่อนเข้าสู่โปรแกรมจะมี Icon ของโปรแกรม ปรากฏดังรูป



รูปที่1 แสดงไอคอนของโปรแกรม LFLATS

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

LFLATS

SLAB DESIGN IN FLAT SLAB AND FLAT PLATE SYSTEMS

BY

SIAM KUNKATE 35104449

PATCHARAWALAI MAITA 35104291

ADVISOR: MR. WIBOOL WUTIYAN

OK

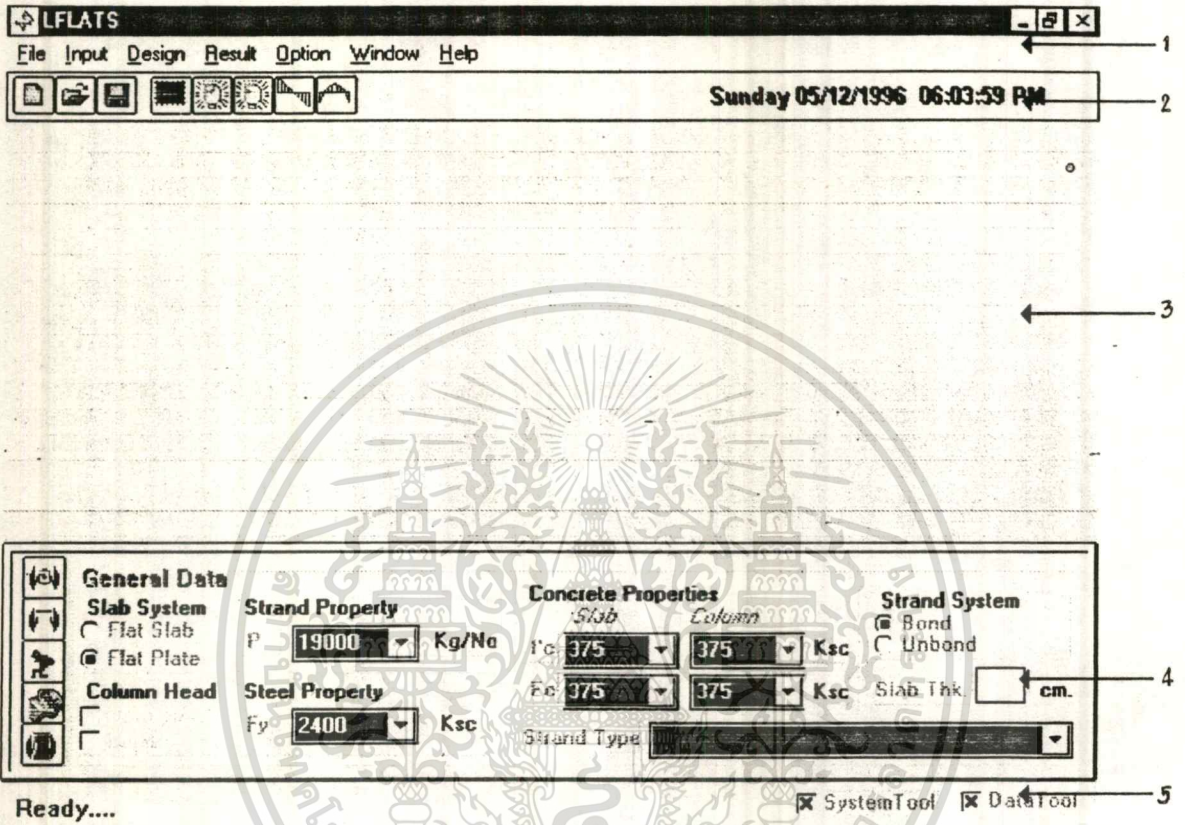
รูปที่ 2 เป็นหน้าจอที่แสดงการแนะนำโปรแกรมการออกแบบเมื่อเริ่มต้นใช้โปรแกรม
อุปกรณ์ที่ใช้และซอฟต์แวร์ที่ต้องใช้ควบคู่กับโปรแกรม LFLATS

1. ไมโครคอมพิวเตอร์ควรจะเป็น 486 DX2-66 ขึ้นไปและควรมี RAM อย่างน้อย 4 M byte เพื่อความรวดเร็วในการคำนวณ แนะนำให้ใช้ 8 M byte
2. Mouse
3. จอสี (โปรแกรมนี้ออกแบบมาสำหรับจอสี)
4. โปรแกรม Windows ควรเป็นรุ่น 3.11 ขึ้นไป
5. เครื่องพิมพ์

ลักษณะโดยรวมของหน้าจอแบ่งเป็น 5 ส่วนใหญ่ ๆ ดังนี้คือ

1. Pull Down Menu ประกอบด้วยเมนู 7 เมนู แต่ละเมนูจะประกอบด้วยเมนูย่อย ๆ ซึ่ง
จะอธิบายในภายหลัง
2. Icon Menu เป็นเมนูที่เป็นภาพที่เป็นภาพทำให้โปรแกรมใช้ได้สะดวกขึ้น
3. หน้าจอแสดง Output จะแสดงผลการคำนวณได้แก่ โมเมนต์ แรงเฉือน และอื่น ๆ ซึ่ง
จะแสดงเป็นกราฟิก หรือเป็นรายการ

4. หน้าจอแสดงส่วนที่ต้องใส่ข้อมูล ซึ่งจะมียู่หลายหน้าจอ สามารถจะเลือกโดยใช้ Icon ที่เป็นรูปภาพที่สื่อความหมายทำให้โปรแกรมใช้ได้สะดวกขึ้น
5. Status Bar เป็นส่วนที่อำนวยความสะดวกในการบอกสถานะของโปรแกรมในขณะนั้น



รูปที่ 3 แสดงส่วนต่าง ๆ บนหน้าจอของตัวโปรแกรม

ส่วนประกอบของโปรแกรมโดยละเอียด

1. Pull Down Menu ประกอบด้วยเมนู 7 เมนู

1.Menu file (Alt+F) เป็นเมนูที่จัดการเกี่ยวกับ file จะมีเมนูย่อย ๆ ให้เลือกจัดการกับ file ดังนี้

1.1 New (F1) เป็นการเปิด File ใหม่ทุกครั้งที่ต้องการทำโครงการใหม่

1.2 Open (F3) เป็นการเปิด file ที่มีอยู่แล้วขึ้นมาแก้ไข

1.3 Save (F2) เป็นการบันทึกข้อมูลที่กำลังทำงานอยู่โดยใช้ชื่อเดิม การบันทึกจะทำได้เมื่อ file ถูก New หรือ Open เท่านั้น

1.4 Save as เป็นการบันทึกข้อมูลที่กำลังทำงานอยู่โดยบันทึกโดยมีการตั้งชื่อใหม่ การบันทึกต้องมีการ New หรือ Open file เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 Print เป็นเมนูที่ใช้ในการพิมพ์ ประกอบด้วยเมนูย่อยดังนี้

1.5.1 Flat Slab & Flat Plate เป็นการพิมพ์ผลของการออกแบบใน ส่วนของแผ่นพื้นคอนกรีตอัดแรงได้แก่

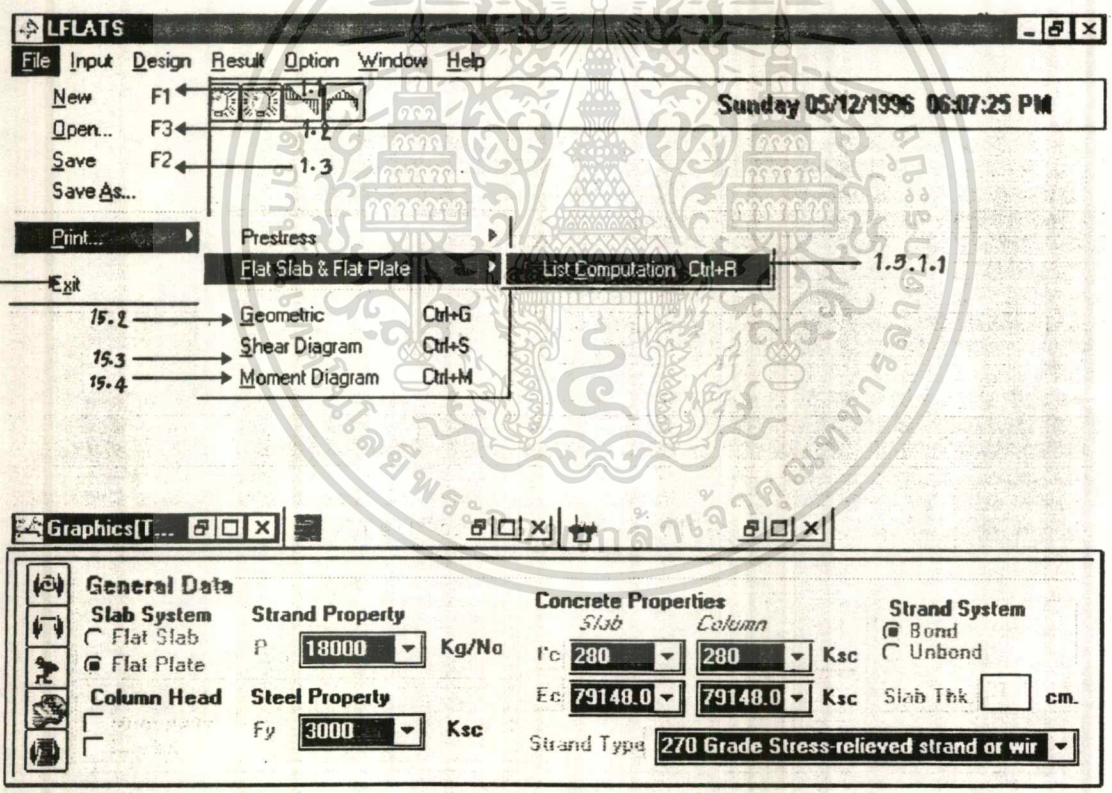
1.5.1.1 List Computation (Ctrl+R)เป็นการพิมพ์ผลของการคำนวณ ทั้งหมดออกมาเป็นรายการคำนวณ

1.5.2 Geometric (Ctrl+G) เป็นการพิมพ์ผลของการคำนวณในส่วน ของกราฟิกแสดงรูปร่าง ขนาด ตำแหน่งของแผ่นพื้น

1.5.3 Shear Diagram (Ctrl+S) เป็นการพิมพ์ผลการคำนวณแรงเฉือน ทางด้านกราฟิก

1.5.4 Moment Diagram (Ctrl+M) เป็นการพิมพ์ผลการคำนวณ โมเมนต์ทางด้านกราฟิก

1.6 Exit เพื่อออกจากโปรแกรม



รูปที่ 4 แสดงหน้าจอซึ่งแสดง Pull down Menu File และเมนูย่อยต่าง ๆ

2.Menu Input (Alt+I) เป็นเมนูสำหรับใส่ข้อมูลที่ใช้ในการ Design

2.1 Node (F4) สำหรับเปิด Panel ของการใส่ข้อมูลของ คุณสมบัติของเสาที่ใช้ในการออกแบบพื้น (อธิบายใน การใส่ข้อมูล Node)

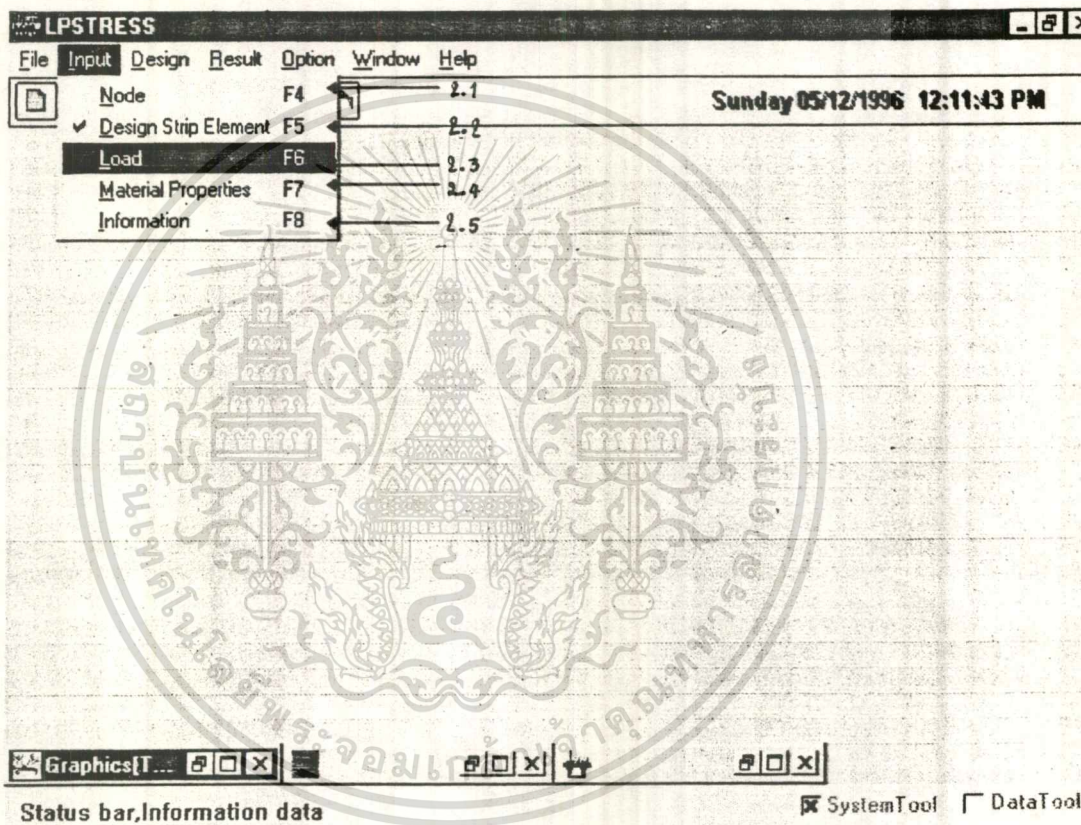
2.2 Design Strip Element (F5) สำหรับเปิด Panel แสดง Design Strip ในแนวแกน X และ แกน Y

2.3 Load (F6) สำหรับเปิด Panel สำหรับใส่ข้อมูลของ แรงกระทำกับพื้น

2.4 Material Properties (F7) สำหรับเปิด Panel สำหรับ ใส่ข้อมูลของวัสดุ

2.5 Information (F8) สำหรับเปิด Panel สำหรับ Information ซึ่งเป็นข้อมูลซึ่ง

ไม่มีผลต่อการคำนวณ

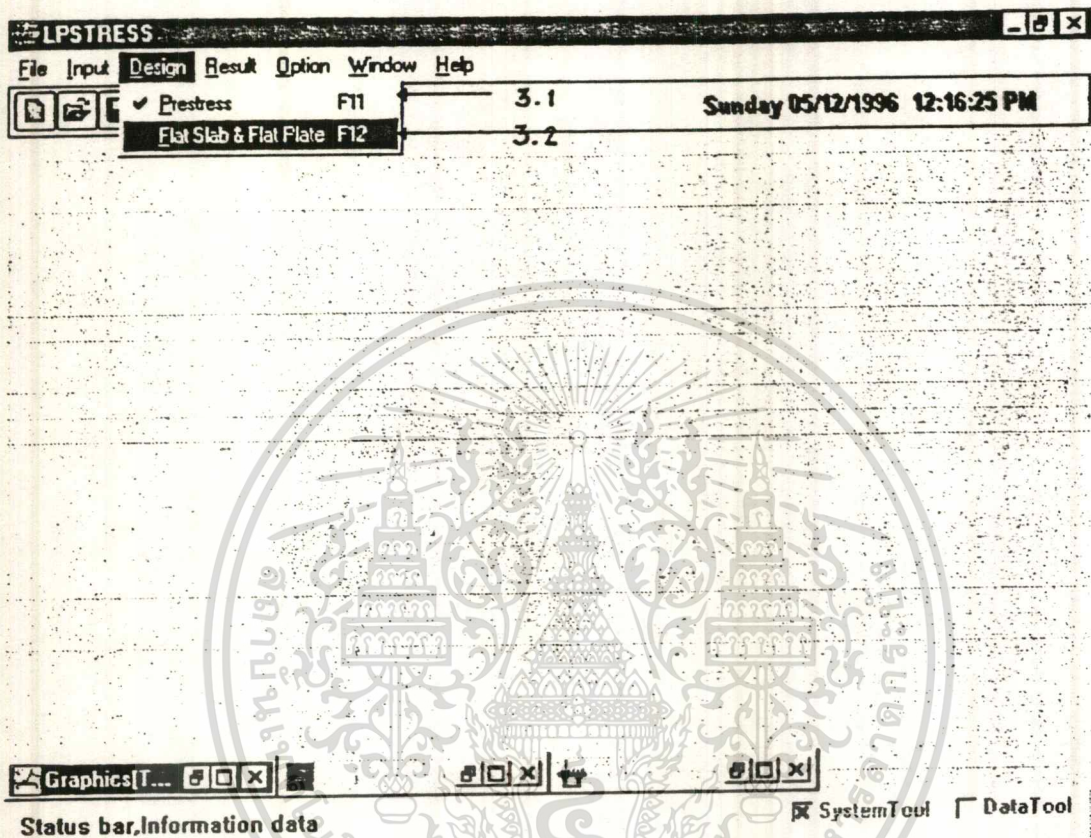


รูปที่ 5 แสดงหน้าจอซึ่งแสดง Pull down Menu Input และเมนูย่อยต่าง ๆ

3.Menu Design

3.1 Pre stress (F11) ใช้ในการออกแบบพื้นรูปแบบ Pre-stressed

3.2 Flat Slab & Flat Plate (F12) ใช้ในการออกแบบพื้นรูปแบบ Flat Slab & Flat Plate



รูปที่ 6 แสดงหน้าจอซึ่งแสดง Pull down Menu Design และเมนูย่อยต่าง ๆ

4.Menu Result

จะแสดงผลของของการ Design ในหน้าจอตั้งหัวข้อ 3
(แสดงรายละเอียดในจอแสดงผลอีกครั้งใน หัวข้อ 3)

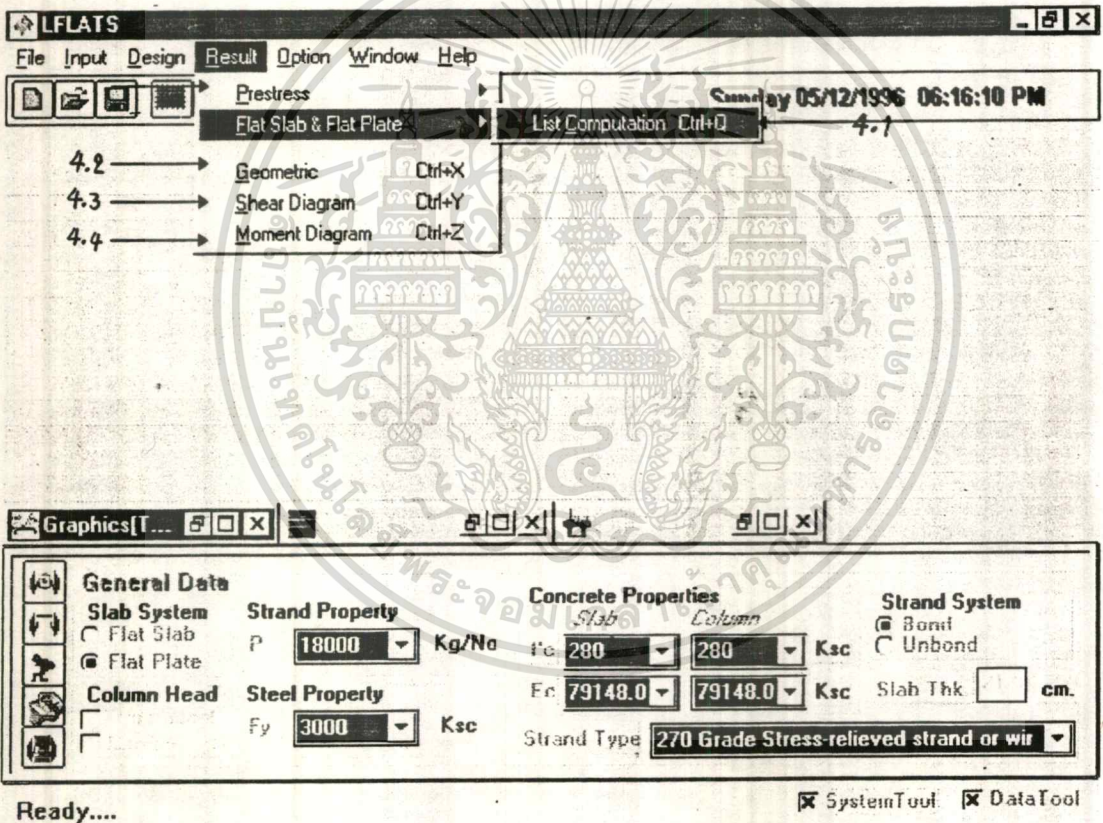
4.1 Flat Slab & Flat Plate

4.1.1 List Computation (Ctrl+Q) สำหรับแสดงพื้นที่หน้าตัดการเสริมเหล็กเสริม และ รายการคำนวณ จะ Active เมื่อ Design Flat Slab & Flat Plate ก่อน

4.2 Geometric (Ctrl+X) แสดงรูป Graphics ของลักษณะของแผ่นพื้น

4.3 Shear Diagram (Ctrl+Y)

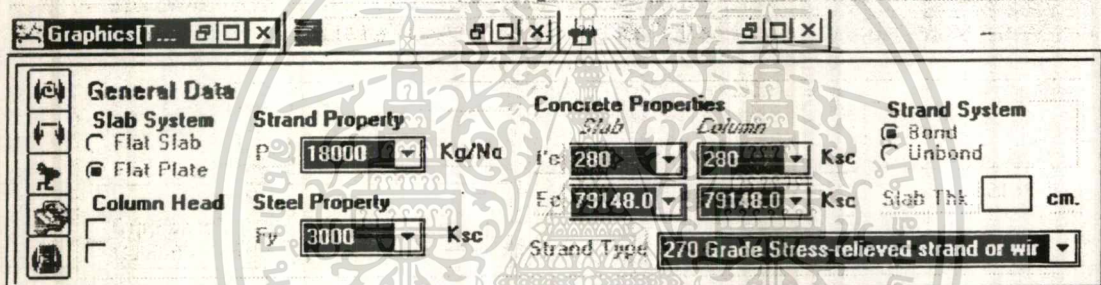
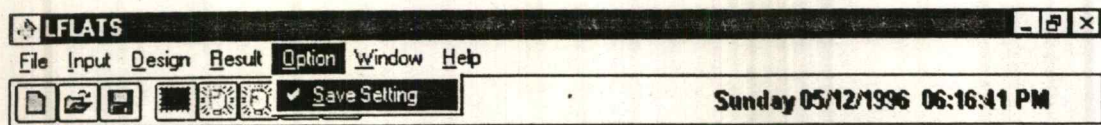
4.4 Moment Diagram (Ctrl+Z)



รูปที่ 7 แสดงหน้าจอซึ่งแสดง Pull down Menu Result และเมนูย่อยต่าง ๆ

5.Menu Option

5.1 Save setting เพื่อ เก็บค่าไว้ว่าแสดง Icon Menu หรือ Panel Input data ใน ครั้งต่อไปเพื่อแสดงหรือไม่แสดงเมื่อเข้าโปรแกรม



Status bar, Information data

SystemTool DataTool

รูปที่ 8 แสดงหน้าจอซึ่งแสดง Pull down Menu Save setting และเมนูย่อยต่าง ๆ

6.Menu Window

6.1 Arrange Icon (Ctrl+A) การเรียง Icon ให้เป็นระเบียบ

6.2 Close All Windows (Ctrl+W) การปิดหน้าต่าง Output 3 หน้าต่าง

LFLATS

File Input Design Result Option Window Help

Arrange Icon Ctrl+A

Close All Windows Ctrl+W

Sunday 05/12/1996 06:17:54 PM

Graphics[T...]

General Data

Slab System
 Flat Slab
 Flat Plate

Column Head

Strand Property
 P Kg/Na

Steel Property
 Fy Ksc

Concrete Properties

| | Slab | Column |
|-----|--------------------------------------|--------------------------------------|
| f'c | <input type="text" value="280"/> | <input type="text" value="280"/> |
| Fc | <input type="text" value="79148.0"/> | <input type="text" value="79148.0"/> |

Strand Type

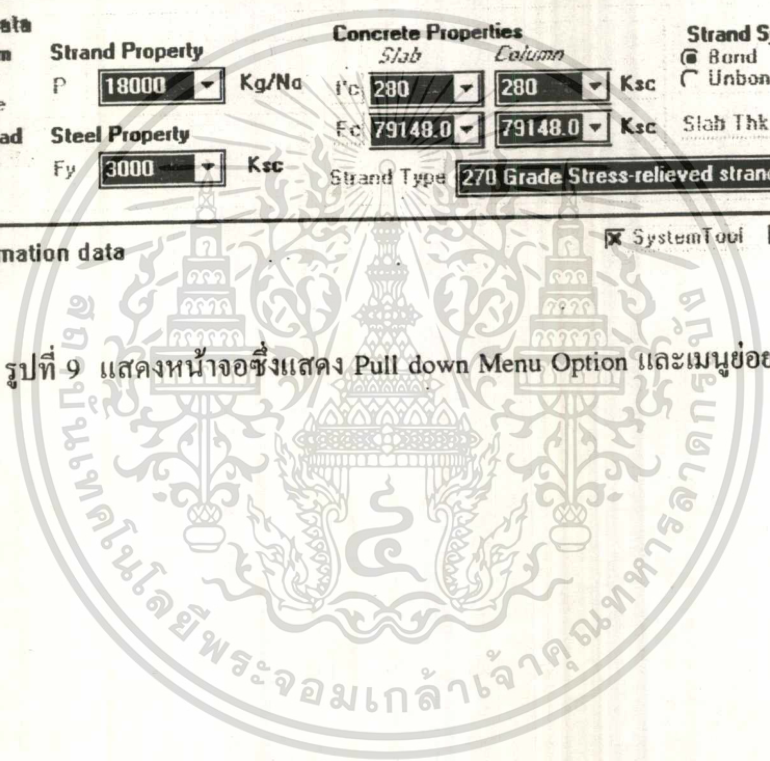
Strand System
 Bond
 Unbond

Slab Thk. cm.

Status bar, Information data

SystemTool DataTool

รูปที่ 9 แสดงหน้าจอซึ่งแสดง Pull down Menu Option และเมนูย่อยต่าง ๆ

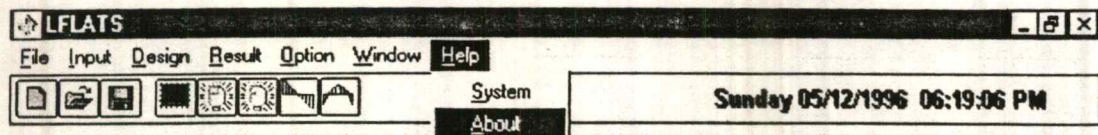


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.Menu Help

7.1 System แสดงว่าทรัพยากรของเครื่องนั้นเป็นอย่างไร

7.2 About แสดงหน้าจอเหมือนกับหน้าจอในรูปที่ 2

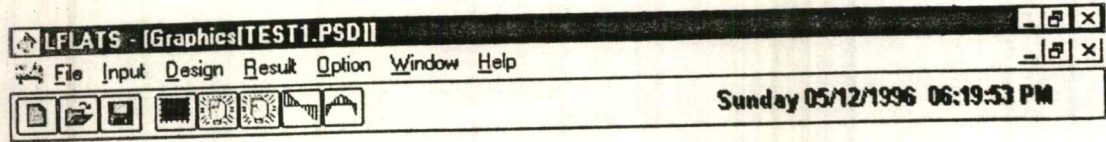


Properties of concrete in slab and column

 SystemTool DataTool

รูปที่ 10 แสดงหน้าจอซึ่งแสดง Pull down Menu About และเมนูย่อยต่าง ๆ

2.Icon Menu เป็นเมนูที่เป็นภาพที่เป็นภาพทำให้โปรแกรมใช้ได้สะดวกขึ้น แต่ต้องมีการ Check ที่ System Tool ใน Status bar ก่อน





Status bar, Information data

System Tool Data Tool


คลิกที่

รูปที่ 11 แสดงหน้าจอซึ่งแสดง การคลิก System Tool


 เป็น Icon Menu สำหรับเปิดไฟล์ใหม่ขึ้นมาทำ

 เป็น Icon Menu สำหรับเปิดไฟล์ที่มีอยู่ขึ้นมาทำการแก้ไขหรือทำงานต่อ


 เป็น Icon Menu สำหรับบันทึกข้อมูลโดยอัตโนมัติ

 เป็น Icon Menu สำหรับ แสดงรูปของพื้นที่ใช้ในการ Design แต่ต้องแน่ใจว่าใส่ข้อมูลของพื้นที่นั้นถูกต้อง จะ Active เมื่อมีการเปิด ไฟล์ใหม่ หรือเก่าขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

 เป็น Icon Menu สำหรับ Design Pre-stressed จะ Active เมื่อมีการเปิดไฟล์ใหม่ หรือ เก้าขึ้นมา

 เป็น Icon Menu สำหรับ Design Flat Slab & Flat Plate จะ Active เมื่อมีการเปิดไฟล์ใหม่ หรือ เก้าขึ้นมา

 เป็น Icon Menu สำหรับ แสดง Shear Force Diagram จะ Active เมื่อมีการ Design ใดๆ อย่างหนึ่ง

 เป็น Icon Menu สำหรับ แสดง Bending Moment Diagram จะ Active เมื่อมีการ Design ใดๆ อย่างหนึ่ง



3. หน้าจอแสดง Output จะแสดงผลการคำนวณได้แก่ โมเมนต์ แรงเฉือน และอื่น ๆ ซึ่งจะแสดงเป็นกราฟิก หรือเป็นรายการ

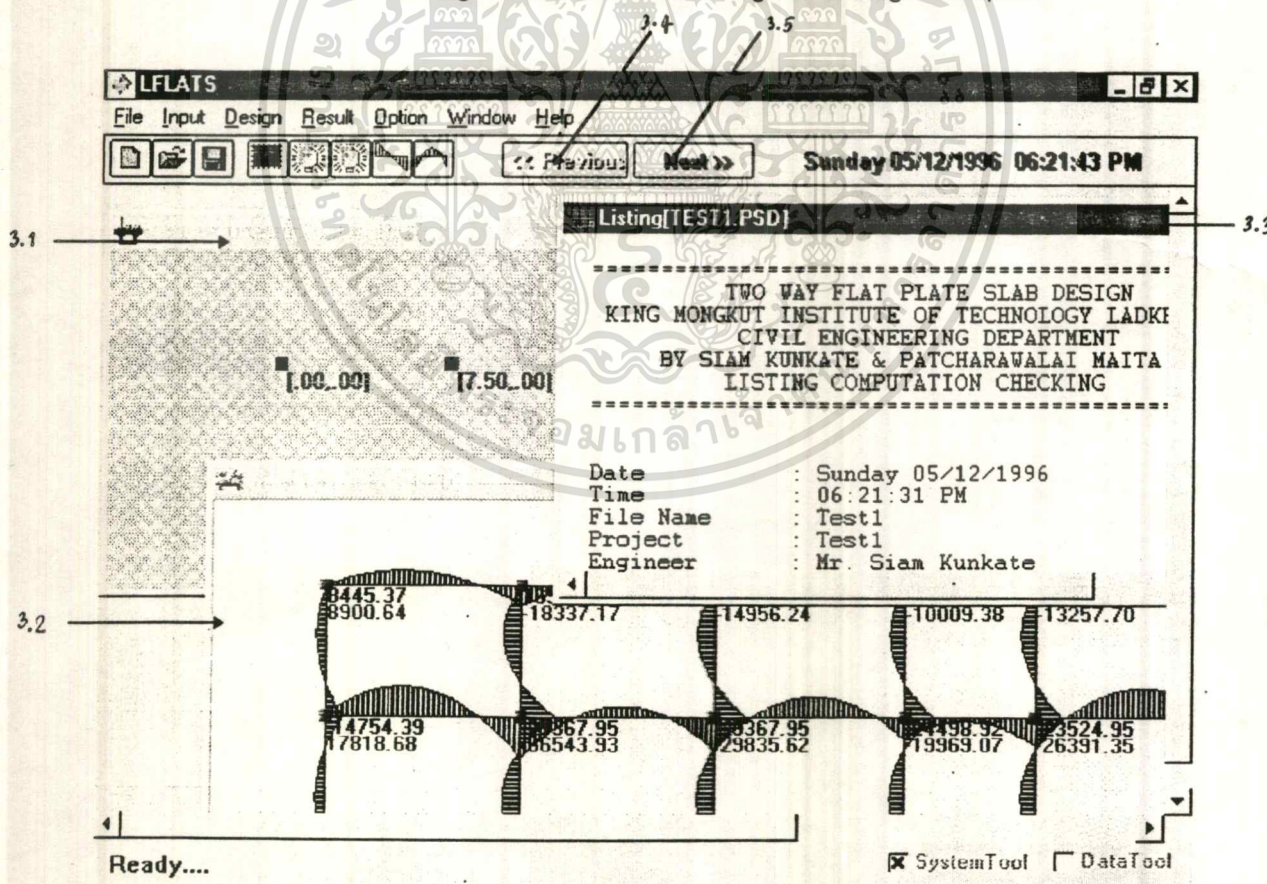
3.1 Slab To Design เป็นหน้าต่าง แสดงรูปของพื้นคอน Load ข้อมูลขึ้นมา จุดสีดำแสดงตำแหน่งเสา และในวงเล็บเป็น Coordinate ของเสา โดยแกน X วัดจากซ้ายไปขวา แกน Y วัดจากบนลงล่าง

3.2 Graphics เป็นหน้าต่าง แสดง Graphics ของ Shear Diagram ,Moment Diagram

3.3 Listing เป็นหน้าต่าง แสดง ผลที่เป็น Text ของการ Design

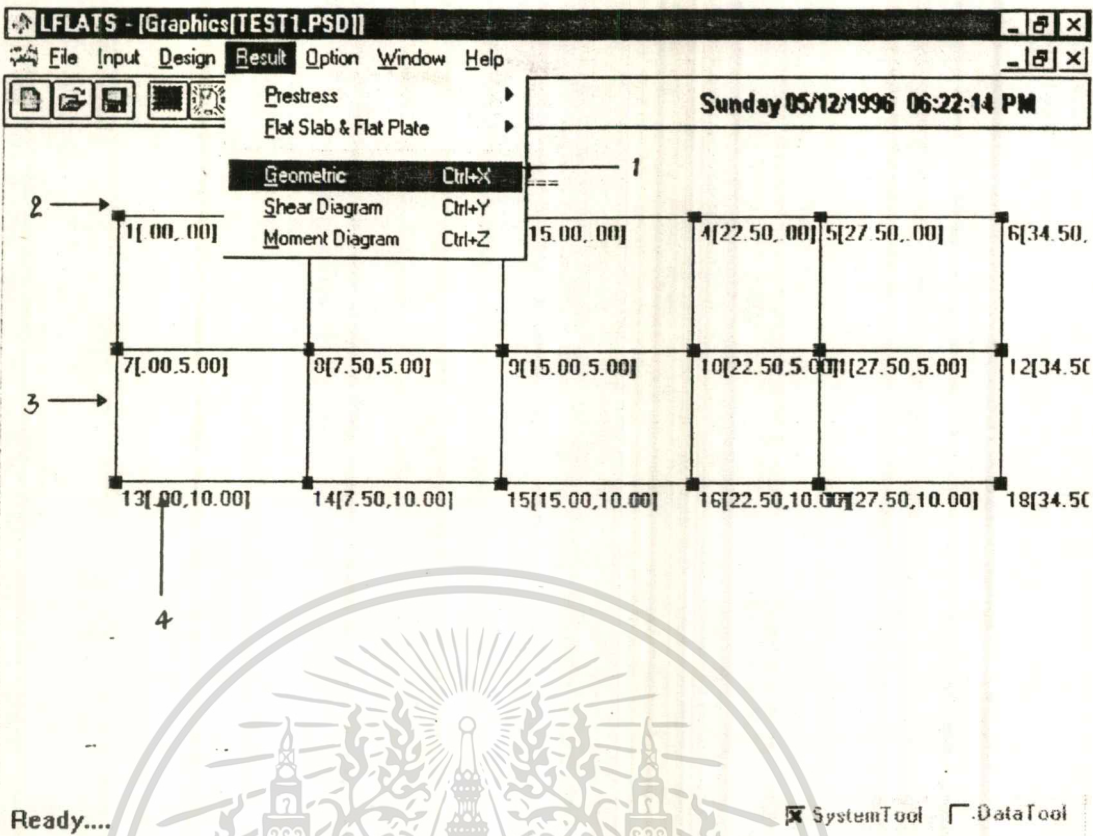
3.4 ปุ่ม Previous จะปรากฏเมื่อหน้าต่าง Listing ปรากฏเท่านั้น สำหรับเปิด Listing ย้อนกลับไปหน้าที่ผ่านมาแล้วของ Listing เมื่อ Listing มีหลายหน้า

3.5 ปุ่ม Next จะปรากฏเมื่อหน้าต่าง Listing ปรากฏเท่านั้น สำหรับเปิด Listing ไปหน้าต่อไปของ Listing เมื่อ Listing มีหลายหน้า



รูปที่ 12 แสดงหน้าจอซึ่งแสดง หน้าต่าง output ของการ Design

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 13 แสดงหน้าจอซึ่งแสดง หน้าต่าง output ของ Slab To Design
อธิบายเพิ่มเติม

1. เมื่อทำการ Design แล้วคลิกที่ เมนูหมายเลข 1 จะแสดง Output เป็นรูปแบบของพื้นที่ออกแบบ
2. แสดงตำแหน่งเสา
3. แสดง Element ที่เชื่อมต่อระหว่างเสาสองเสา
4. ตัวเลขหน้าวงเล็บบอกหมายเลขเสา ตัวเลขในวงเล็บบอก Co-ordinate x,y ตามลำดับ

| LFLATS - [Listing TEST1.PSD]] | | | | | | |
|---|---|---------------------|--------------|--------|-------------|-------------------------------|
| File Input Design Result Option Window Help | | | | | | |
| | | | | | | Sunday 05/12/1996 06:25:14 PM |
| 1 | 2 | | | Bottom | -3.1762E+03 | 0.00E+00 |
| 1 | 2 | Right Bay | | Top | -3.1762E+03 | 4.52E+00 |
| 1 | 2 | | | Bottom | -3.1762E+03 | 4.51E+00 |
| 1 | 2 | Middle Column Strip | | Top | 1.3184E+04 | 1.87E+01 |
| 1 | 2 | | | Bottom | 1.3184E+04 | 1.87E+01 |
| 1 | 2 | Left Bay | | Top | -5.0820E+03 | 0.00E+00 |
| 1 | 2 | | | Bottom | -5.0820E+03 | 0.00E+00 |
| 1 | 2 | Right Bay | | Top | 8.7896E+03 | 1.25E+01 |
| 1 | 2 | | | Bottom | 8.7896E+03 | 1.25E+01 |
| 1 | 2 | Bottom Column Strip | | Top | 1.1642E+03 | 1.65E+00 |
| 1 | 2 | | | Bottom | 1.1642E+03 | 1.66E+00 |
| 1 | 2 | Left Bay | | Top | 0.0000E+00 | 0.00E+00 |
| 1 | 2 | | | Bottom | 0.0000E+00 | 0.00E+00 |
| 1 | 2 | Right Bay | | Top | 0.0000E+00 | 0.00E+00 |
| 1 | 2 | | | Bottom | 0.0000E+00 | 1.62E-03 |
| ----- | | | | | | |
| 2 | 1 | Top | Column Strip | Top | 2.3327E+03 | 3.30E+00 |
| 2 | 1 | | | Bottom | 2.3327E+03 | 3.33E+00 |
| 2 | 1 | Left Bay | | Top | 0.0000E+00 | 0.00E+00 |
| 2 | 1 | | | Bottom | 0.0000E+00 | 1.62E-03 |
| 2 | 1 | Right Bay | | Top | 0.0000E+00 | 0.00E+00 |
| 2 | 1 | | | Bottom | 0.0000E+00 | 1.62E-03 |
| 2 | 1 | Middle Column Strip | | Top | 8.7059E+03 | 1.24E+01 |
| 2 | 1 | | | Bottom | 8.7059E+03 | 1.24E+01 |
| 2 | 1 | Left Bay | | Top | 2.9020E+03 | 4.12E+00 |
| 2 | 1 | | | Bottom | 2.9020E+03 | 4.13E+00 |
| 2 | 1 | Right Bay | | Top | 2.9020E+03 | 4.12E+00 |
| 2 | 1 | | | Bottom | 2.9020E+03 | 4.13E+00 |

Status bar, Information data

SystemTool DataTool

รูปที่ 15 แสดง พื้นที่เหล็กเสริม

| LFLATS - [Listing TEST1.PSD]] | | | | | | |
|---|---|--|--------|-------------------------|-------|-------------------------------|
| File Input Design Result Option Window Help | | | | | | |
| | | | | | | Sunday 05/12/1996 06:24:23 PM |
| | | Prestress | | Menu | | |
| | | Flat Slab & Flat Plate | | List Computation Ctrl+D | | |
| TWO | | Geometric | Ctrl+X | SIGN | | |
| KING MONCKUT | | Shear Diagram | Ctrl+Y | Y LADKRABANG | | |
| CIV | | Moment Diagram | Ctrl+Z | ENT | | |
| BY SIAM K | | | | | MAITA | |
| LISTING COMPUTATION CHECKING | | | | | | |
| ----- | | | | | | |
| Date | : | Sunday 05/12/1996 | | | | |
| Time | : | 06:24:09 PM | | | | |
| File Name | : | Test1 | | | | |
| Project | : | Test1 | | | | |
| Engineer | : | Mr. Siam Kunkate | | | | |
| Address | : | Civil Engineer KMIT'L | | | | |
| Information | : | Design Slab with prestress system & Flat slab or Flat plat | | | | |
| Length Unit | : | Meter(m) | | | | |
| Load Unit | : | Kilogram(Kg) | | | | |
| fc' | : | 280.00 ksc | | | | |
| fy (Reinforce) | : | 3000.00 ksc | | | | |
| Ec (Slab) | : | 79148.04 ksc | | | | |
| Ec (Column) | : | 79148.04 ksc | | | | |
| COLUMN DATA | | | | | | |
| ----- | | | | | | |

Ready....

SystemTool DataTool

รูปที่ 16 แสดง Output รายการคำนวณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. หน้าจอแสดงส่วนที่ต้องใส่ข้อมูล ซึ่งจะมีอยู่หลายหน้าจอ สามารถจะเลือกโดยใช้ Icon ที่เป็นรูปภาพที่สื่อความหมายทำให้โปรแกรมใช้ได้สะดวกขึ้น แต่ต้อง Check ที่ Data Tool ใน Status Bar ก่อน จะปรากฏ Panel ของการใส่ข้อมูลในการ Design ปรากฏขึ้นมา

The screenshot shows the LFLATS software interface. The title bar reads "LFLATS - [Graphics[TEST1.PSD]]". The menu bar includes "File Input Design Result Option Window Help". The status bar at the bottom shows "Sunday 05/12/1996 06:26:36 PM".

The "Node Data" panel is visible, containing the following information:

| Node Data | | Edge X | | Edge Y | |
|-----------|---------|----------|--|----------|--|
| Position | | Interior | | Interior | |
| No. | | Left | | Bottom | |
| X | 0 m. | Right | | Top | |
| Y | 0 m. | | | | |
| Arr X:Y | 1 : 1 | | | | |
| X:Y | 0 : 0 | | | | |
| X Dim | 0.30 m. | | | | |
| Y Dim | 0.30 m. | | | | |
| Length | 3 m. | | | | |

The "Edge" table is also visible:

| No. | X | Y | Edge | Edge | Supp | Width | De |
|-----|-------|------|------|------|------|-------|-----|
| 1 | 0.00 | 0.00 | 1 | 1 | 0 | 0.30 | 0.1 |
| 2 | 7.50 | 0.00 | 0 | 1 | 0 | 0.30 | 0.1 |
| 3 | 15.01 | 0.00 | 0 | 1 | 0 | 0.30 | 0.1 |
| 4 | 22.51 | 0.00 | 0 | 1 | 0 | 0.30 | 0.1 |

The Status Bar at the bottom shows "SystemTool" and "DataTool" (checked). An arrow points to the "DataTool" checkbox with the text "คลิกที่" (Click).

รูปที่ 17 แสดงหน้าจอซึ่งแสดงการคลิก Data Tool

คลิกที่

4.1 Node Data

LFLATS - [Graphics[TEST1.PSD]]

File Input Design Result Option Window Help

Node F4

Design Strip Element F5

Load F6

Material Properties F7

Information F8

Sunday 05/12/1996 06:28:03 PM

เส้นที่ 1

เส้นที่ 2

เส้นที่ 3

1[0.00,0.00] 2[7.50,0.00] 3[15.00,0.00] 4[22.50,0.00] 5[27.50,0.00] 6[34.50,0.00]

7[0.00,5.00] 8[7.50,5.00] 9[15.00,5.00] 10[22.50,5.00] 11[27.50,5.00] 12[34.50,5.00]

13[0.00,10.00] 14[7.50,10.00] 15[15.00,10.00] 16[22.50,10.00] 17[27.50,10.00] 18[34.50,10.00]

Node Data

Position

No. 1

X 0 m.

Y 0 m.

Arr X:Y 1

X:Y 0

X Dim 0.30 m.

Y Dim 0.30 m.

Length 3 m.

Edge_X

Interior

Left

Right

Edge_Y

Interior

Bottom

Top

| No. | X | Y | Edge | Edge | Supp | Width | De |
|-----|-------|------|------|------|------|-------|-----|
| 1 | 0.00 | 0.00 | 1 | 1 | 0 | 0.30 | 0.1 |
| 2 | 7.50 | 0.00 | 0 | 1 | 0 | 0.30 | 0.1 |
| 3 | 15.00 | 0.00 | 0 | 1 | 0 | 0.30 | 0.1 |
| 4 | 22.50 | 0.00 | 0 | 1 | 0 | 0.30 | 0.1 |

Status bar, Information data

SystemTool DataTool

รูปที่ 18 แสดงหน้าจอซึ่งแสดง Panel การใส่ข้อมูลของพื้น

อธิบายการใส่ข้อมูลตามรูปตามตัวเลขที่ปรากฏ

1,2 คลิกที่นี้จะมี Panel ใส่ข้อมูลปรากฏดังรูป

3 หมายเลขของเสา เป็นจำนวนเต็มบวกและต้องเรียงเช่น ข้อมูลเสามี 8 ต้น เมื่อต้องการเพิ่มเสาอีกก็ต้องกรอกข้อมูลเลข 9 เป็นเสาที่ 9

4. Coordinate ของเสาในแกน X

5. Coordinate ของเสาในแกน Y

6. ขนาดของเสาในแกน X

7. ขนาดของเสาในแกน Y

8. ความยาวของเสา

9. ลักษณะความเป็นขอบในแกน X ของเสา มี 3 แบบ คือ Interior, Left, Right เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเสาที่ 1 Edge X เป็น Interior

ตัวอย่างเสาที่ 2 Edge X เป็น Right

ตัวอย่างเสาที่ 3 Edge X เป็น Interior

10 ลักษณะความเป็นขอบในแกน Y ของเสา มี 3 แบบ คือ Interior, Top, Bottom เช่น

ตัวอย่างเสาที่ 1 Edge Y เป็น Top

ตัวอย่างเสาที่ 2 Edge Y เป็น Top

ตัวอย่างเสาที่ 3 Edge Y เป็น Interior

การใส่ข้อมูลเสาเป็นชุด

11 จำนวน Column ของเสาในแกน X ที่เรียงเป็น Array

12 จำนวน Row ของเสาในแกน Y ที่เรียงเป็น Array

13 Coordinate ในแกน X ของเสาในด้านขวาล่าง

14 Coordinate ในแกน Y ของเสาในด้านขวาล่าง

ตัวอย่างการใส่ข้อมูลเสาเป็นชุด

ดังในรูปที่ 13 สมมติว่าระยะห่างของทุกเสาเป็น 5 ทั้งหมด

จะใส่ No เป็น 1

X เป็น 0

Y เป็น 0

ArrX:Y แกน X เป็น 6 (มีเสาอยู่ 6 Column)

แกน Y เป็น 3 (มีเสาอยู่ 3 Row)

X:Y แกน X เป็น 25 ($0+5*5=25$)

แกน Y เป็น 15 ($0+3*5=15$)

ถ้าต้องการใส่ข้อมูล 1 Node ArrX:Y เป็น 1,1 X:Y เป็น 0,0

15 ปุ่ม OK กดเมื่อกรอกข้อมูลเรียบร้อยแล้ว

16 เป็นตารางแสดงข้อมูล Node ที่ใส่เรียบร้อยแล้ว

ช่อง 1 เป็นหมายเลขของ Node

ช่อง 2 เป็น Coordinate แกน X

ช่อง 3 เป็น Coordinate แกน Y

ช่อง 4 เป็นสถานะของขอบแกน X

0 เป็น Interior

1 เป็น Left

2 เป็น Right

ช่อง 5 เป็นสถานะของขอบแกน Y

0 เป็น Interior

1 เป็น Top

2 เป็น Bottom

ช่อง 6 ชนิดของฐานรองรับเป็น 0 แสดงว่ามีเสา

ช่อง 7 ขนาดเสาแกน X

ช่อง 8 ขนาดเสาแกน Y

การแก้ไขข้อมูล Node ที่ใส่ไปแล้ว

1 ลบโดยการเลือกที่ Node ที่ต้องการลบแล้วกด Back space

2 การแก้ไขข้อมูล โดยการเลือกที่ Node ที่ต้องการแก้ไขแล้ว

ข้อมูลทั้งหมดจะปรากฏที่ช่องใส่ข้อมูล แล้วไปแก้ไขที่ช่องใส่ข้อมูลแล้วกด OK การแก้ไขก็เสร็จสมบูรณ์



4.2 Element Data

LPSTRESS - [Graphics[TEST1.PSD]]

File Input Design Result Option Window Help

Sunday 05/12/1996 02:21:26 PM

Node F4

Design Strip Element F5

Load F6

Material Properties F7

Information F8

1[.00,.00] 2[7.50,.00] 3[15.00,.00] 4[22.50,.00] 5[27.50,.00] 6[34.50,.00]

7[.00,5.00] 8[7.50,5.00] 9[15.00,5.00] 10[22.50,5.00] 11[27.50,5.00] 12[34.50,5.00]

13[.00,10.00] 14[7.50,10.00] 15[15.00,10.00] 16[22.50,10.00] 17[27.50,10.00] 18[34.50,10.00]

3 4 5 6 7 8 9 10 11

| No. | Along | Line | Element | Column1 | Column2 | Length | Bay1 | Bay2 |
|-----|-------|------|---------|---------|---------|--------|------|------|
| 1 | X | 1 | 1 | 1 | 2 | 7.5 | 0 | 5 |
| 2 | X | 1 | 2 | 2 | 3 | 7.5 | 0 | 5 |
| 3 | X | 1 | 3 | 3 | 4 | 7.5 | 0 | 5 |
| 4 | X | 1 | 4 | 4 | 5 | 5 | 0 | 5 |
| 5 | X | 1 | 5 | 5 | 6 | 7 | 0 | 5 |

Ready....

SystemTool DataTool

รูปที่ 19 แสดงหน้าจอซึ่งแสดง Panel แสดง Design strip

อธิบายการส่งข้อมูลตามรูปตามตัวเลขที่ปรากฏ

1,2 คลิกที่นี้จะปรากฏ Panel ที่ใช้ในการแสดง Design strip Element จะปรากฏข้อมูลเมื่อมีการ Design ใดๆอย่างหนึ่งแล้ว

3 No of Element ของ Element ที่อยู่แกน x,y

4 Along เป็นตัวบอกว่า เป็น Strip ในแกน X หรือ Y

5 Line เป็นตัวแสดง Line ของเสา

6 Element เป็นตัวแสดงหมายเลขของ Element ใน Line นั้น

7 Column 1 เชื่อมระหว่าง Element

8 Column 2 เชื่อมระหว่าง Element

9 Length ความยาวของ Element

10 Bay 1 ความกว้างของช่วงพื้น

ถ้าเป็น Element ทางแกน x ก็คือความกว้างช่วงพื้นข้างบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าเป็น Element ทางแกน y ก็คือความกว้างช่วงพื้นด้านซ้าย

11 Bay 2 ความกว้างของช่วงพื้น

ถ้าเป็น Element ทางแกน x ก็คือความกว้างช่วงพื้นด้านล่าง

ถ้าเป็น Element ทางแกน y ก็คือความกว้างช่วงพื้นด้านขวา

Along แสดงว่าอยู่ในแนวแกน x หรือ y

Line แสดง Line ของเสา

Column 1 เชื่อม Element

Column 2 เชื่อม Element

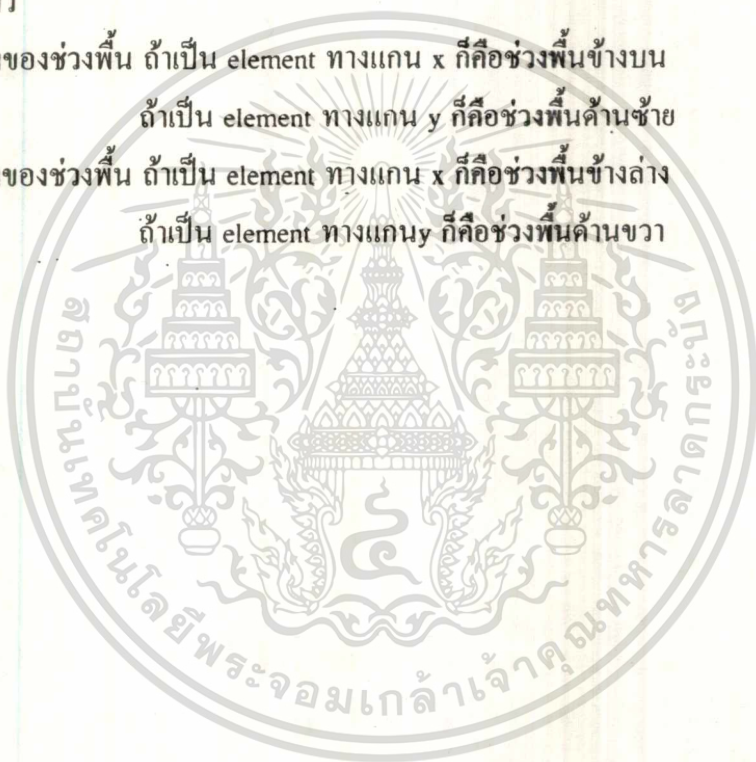
Length ความยาว

Bay 1 ความกว้างของช่วงพื้น ถ้าเป็น element ทางแกน x ก็คือช่วงพื้นข้างบน

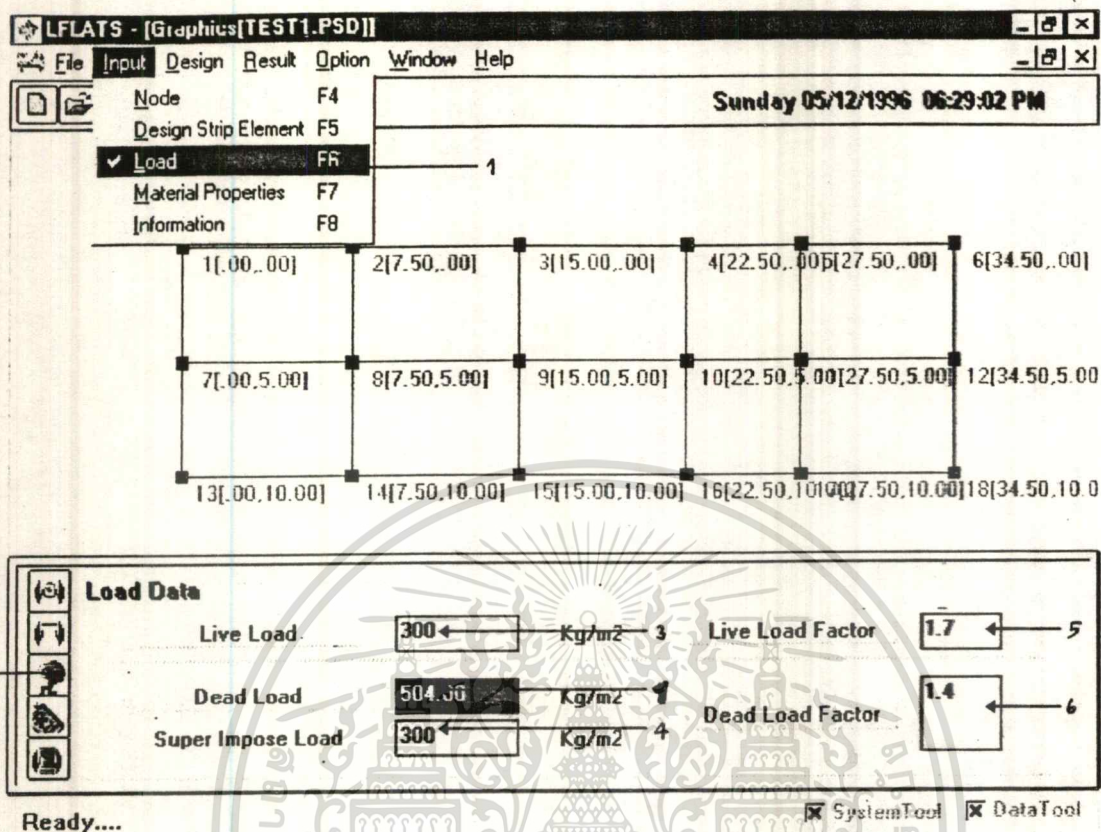
ถ้าเป็น element ทางแกน y ก็คือช่วงพื้นด้านซ้าย

Bay 2 ความกว้างของช่วงพื้น ถ้าเป็น element ทางแกน x ก็คือช่วงพื้นข้างล่าง

ถ้าเป็น element ทางแกน y ก็คือช่วงพื้นด้านขวา



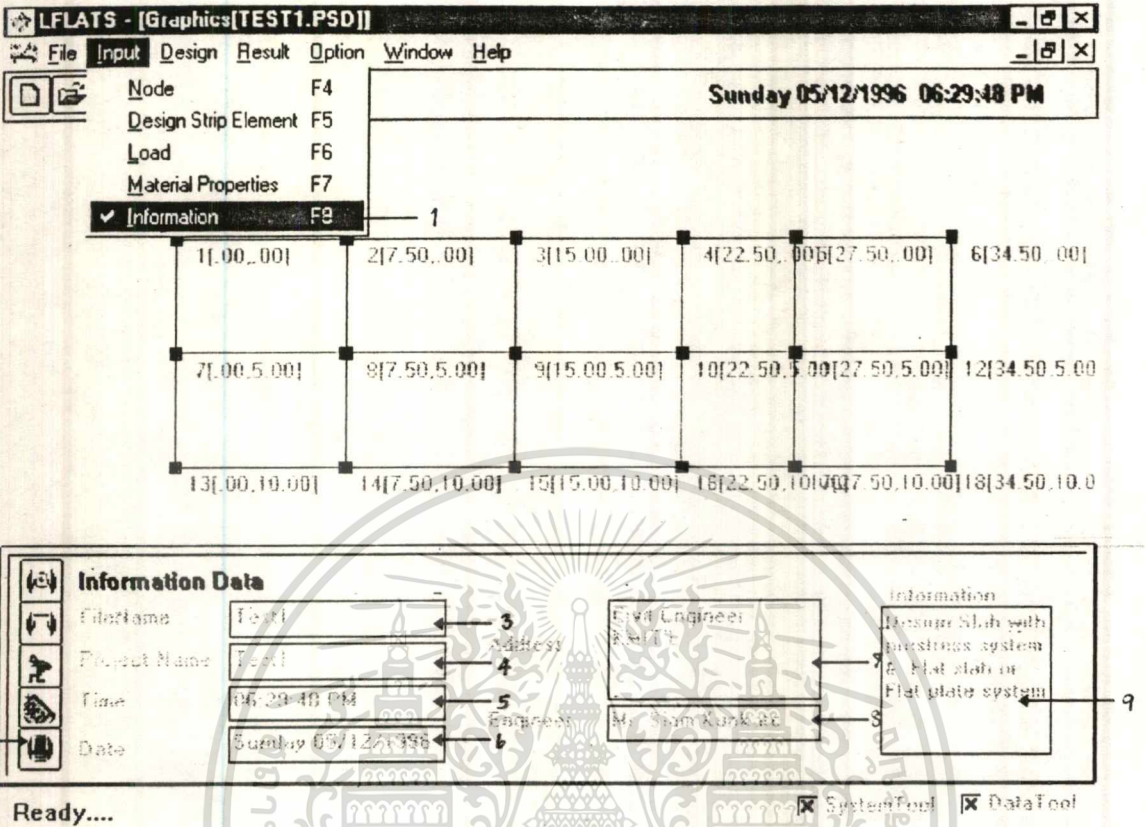
4.3 Load Data



รูปที่ 20 แสดงหน้าจอซึ่งแสดง Panel การใส่ข้อมูลของแรงกระทำบนพื้นและ Factor อธิบายรูปเพิ่มเติม

- เมื่อคลิกที่นี่ จะแสดง Panel ดังแสดงในรูป
- ใส่ข้อมูลน้ำหนักจร
- ใส่ข้อมูล Super Impose Load ใช้สำหรับการออกแบบคอนกรีตอัดแรงเท่านั้น
- ใส่ข้อมูล Load Factor สำหรับน้ำหนักจร (ตามมาตรฐาน ACI Default ไว้ที่ 1.7)
- ใส่ข้อมูล Load Factor สำหรับน้ำหนักตาย (ตามมาตรฐาน ACI Default ไว้ที่ 1.4)
- เป็น Dead Load ที่เป็น Output

4.5 Information Data

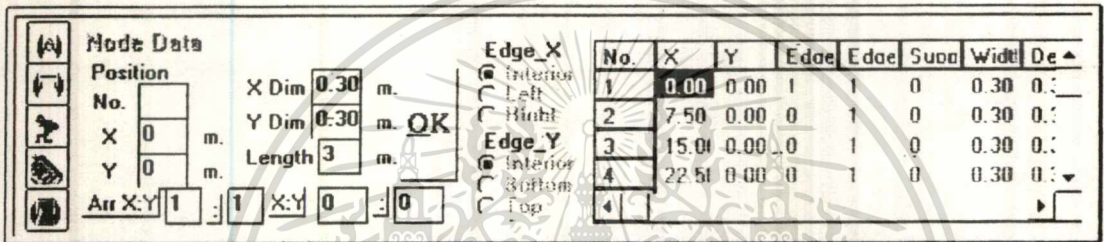
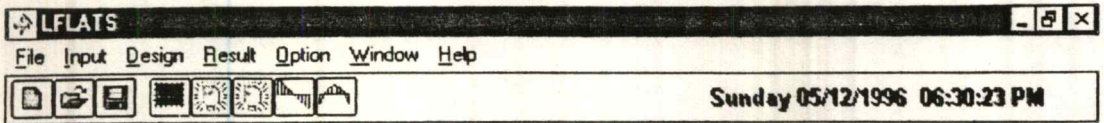


รูปที่ 21 แสดงหน้าจอซึ่งแสดง Panel การใส่ข้อมูล Information

อธิบายรูปเพิ่มเติม

- 1.2 คลิกที่นี้จะปรากฏ Panel ใส่ข้อมูล Information
- 3 ใส่ข้อมูลเป็นการตั้งชื่อไฟล์
4. ใส่ชื่อ โครงการงาน
5. เป็นการบอกเวลาที่ใช้โปรแกรมออกแบบแผ่นพื้น
6. เป็นการบอกวัน เดือน ปี ที่ทำการออกแบบโครงการงาน
7. เป็นการใส่ที่อยู่ตามที่ผู้ใช้งานต้องการ
8. เป็นการใส่ชื่อผู้ใช้โปรแกรม
9. เป็นข้อมูลเพิ่มเติมบางประการที่ผู้ใช้งานต้องการระบุ

5. Status Bar เป็นส่วนที่อำนวยความสะดวกในการบอกสถานะของโปรแกรมในขณะนั้น



Display column data such as coordinate, length, column size, etc.

System Tool

Data Tool

รูปที่ 22 แสดงหน้าจอซึ่งแสดง Panel การใส่ข้อมูล Information

อธิบายรูปเพิ่มเติม

1 แสดง Status Bar

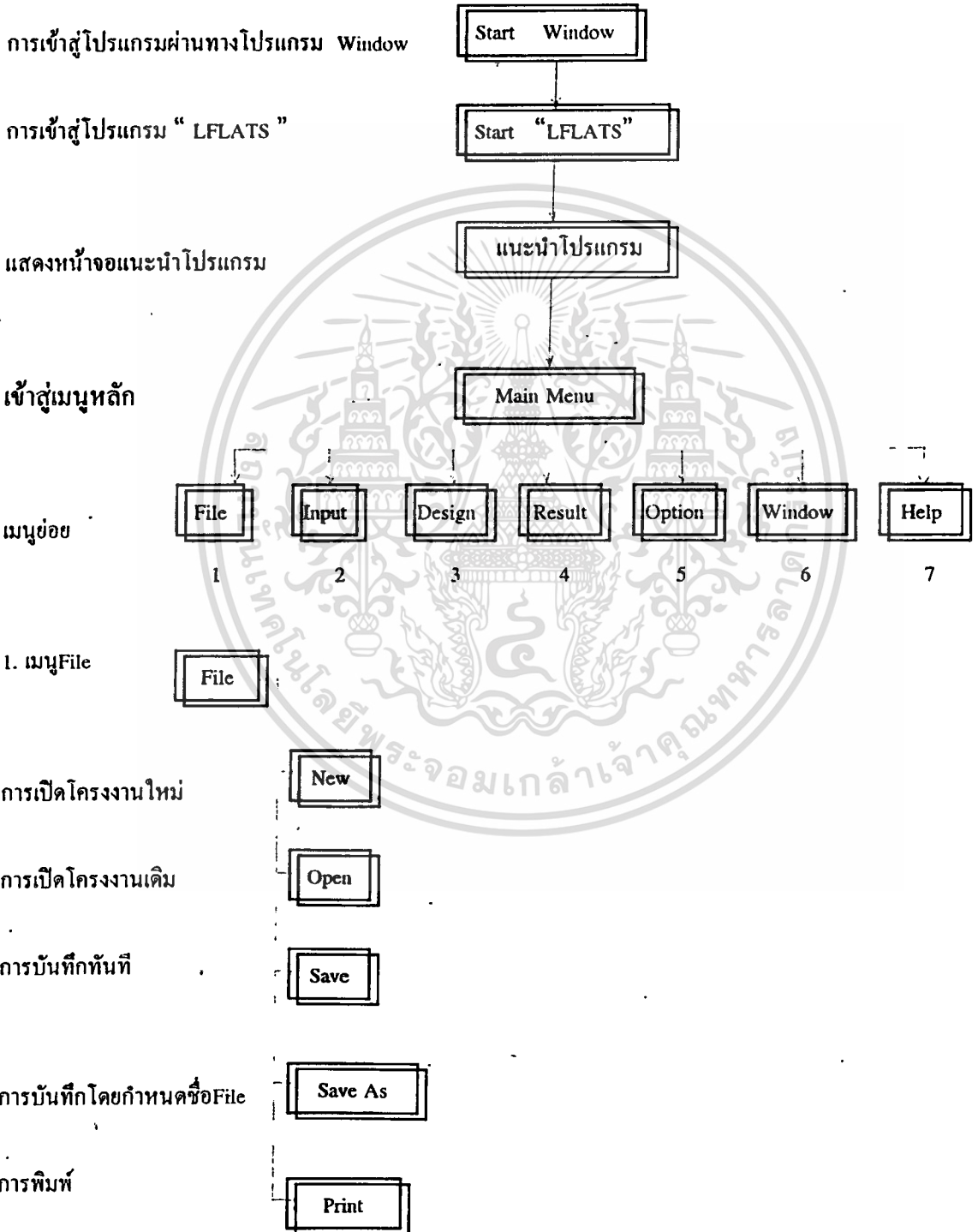
2 System Tool เป็นส่วนที่ใช้แสดงสถานะของโปรแกรมขณะที่ทำงานอยู่ว่าแสดง Menu Icon ที่เป็นรูปภาพปรากฏอยู่ด้านบนไว้เพื่ออำนวยความสะดวกในการทำงาน

3 Data Tool เป็นส่วนที่ใช้แสดงสถานะของโปรแกรมขณะที่ทำงานอยู่ว่าแสดง Panel สำหรับใส่ข้อมูล

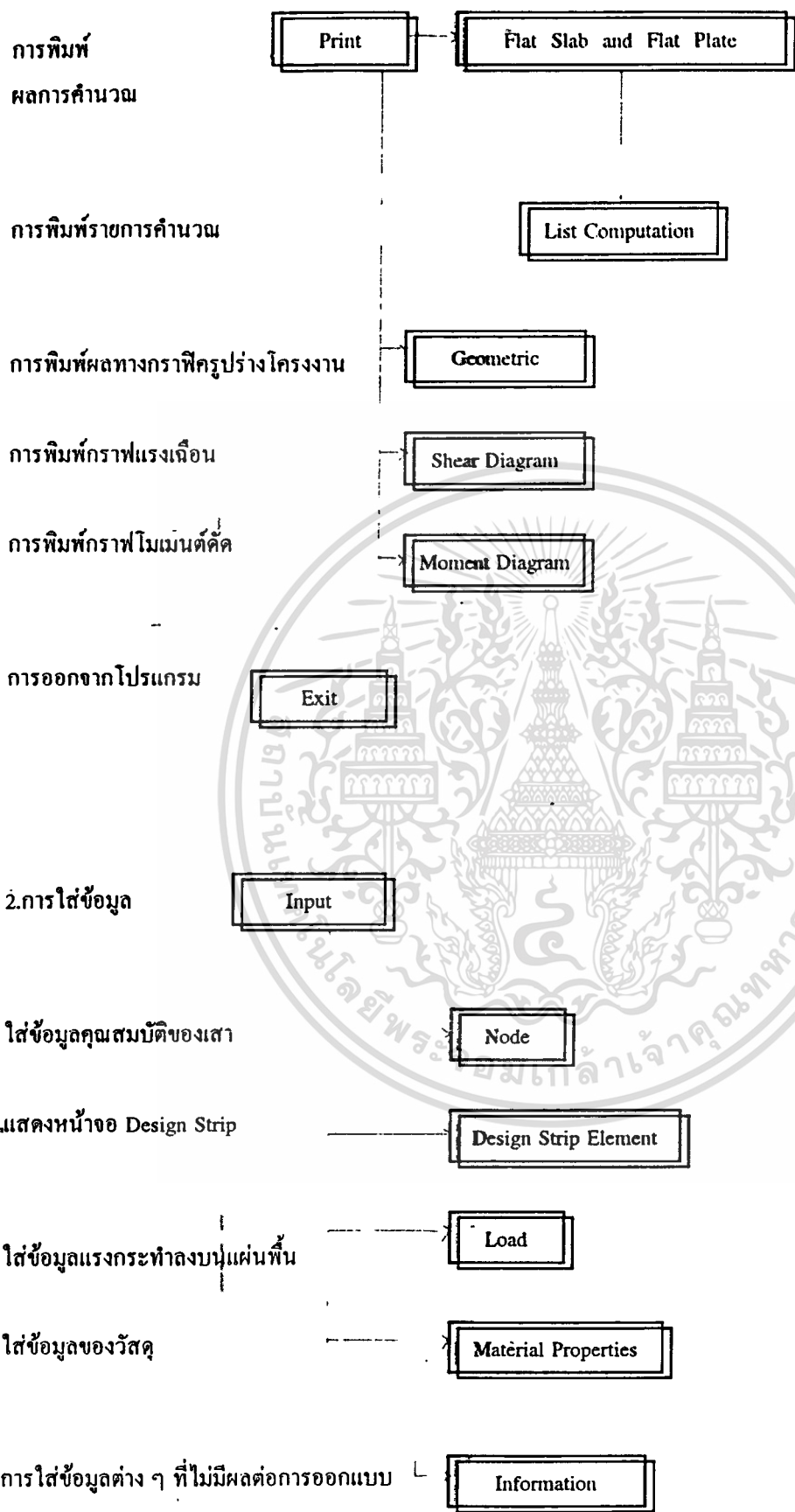
4 นอกจากนี้จะมีข้อความที่แสดงสถานะของโปรแกรมว่าขณะนี้โปรแกรมกำลังแสดงส่วนใดของโปรแกรมอยู่

ลำดับขั้นตอนการใช้งานของโปรแกรม “LFLATS”

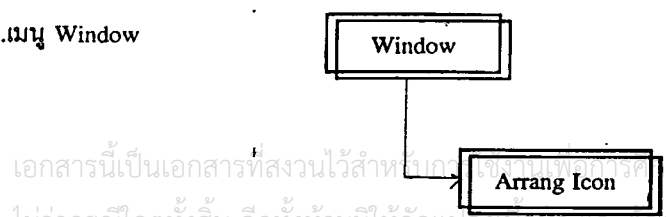
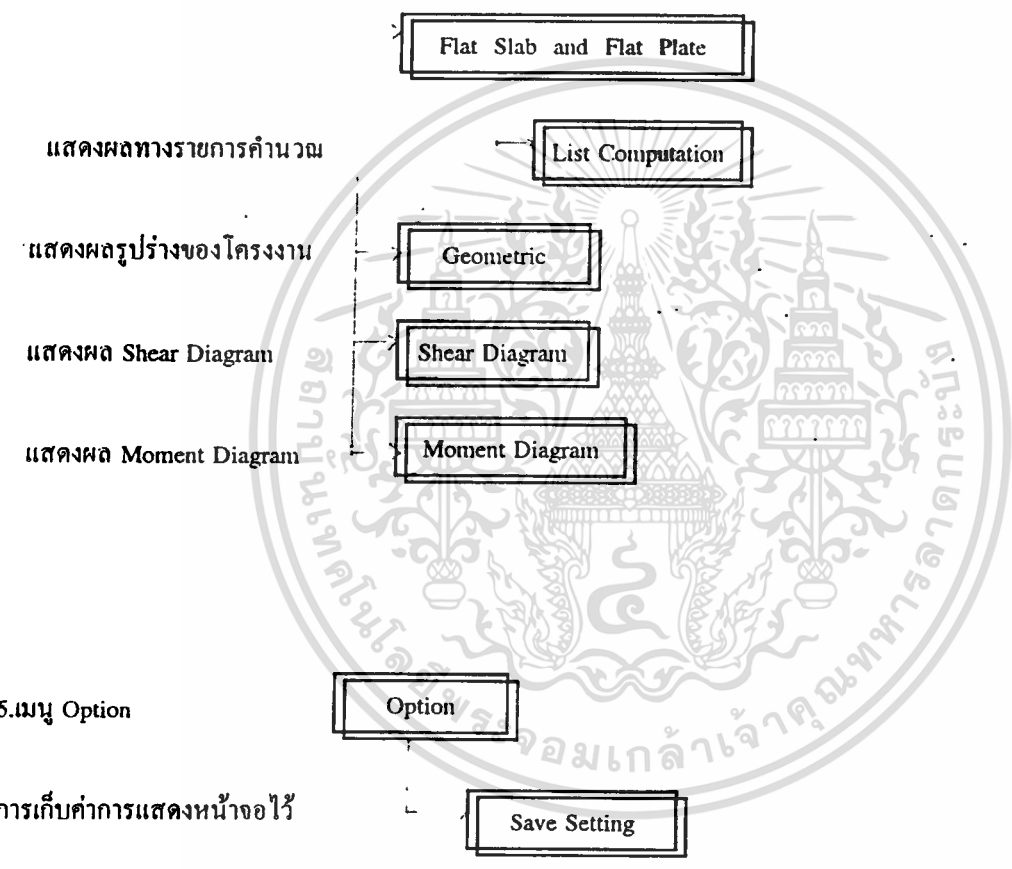
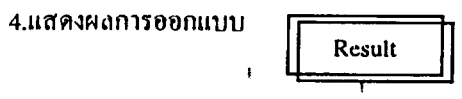
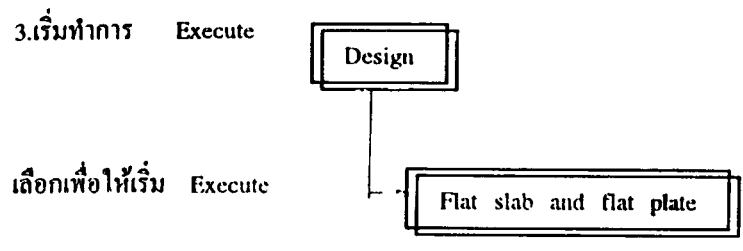
โปรแกรม “LFLATS” เป็นโปรแกรมที่สร้างขึ้นให้ใช้งานได้ง่าย โดยบนหน้าจอของเครื่องคอมพิวเตอร์ จะปรากฏทั้งหน้าจอในส่วนของการใส่ข้อมูลและหน้าจอในส่วนของการแสดงผลในเวลาเดียวกันโดยไม่ต้อง เปลี่ยนหน้าจอเรื่อย ๆ ไปที่หน้าจอจนทำให้ล้าหรือการทำงานไม่สะดวก ฉะนั้นผู้ใช้งานจึงทำงานได้ภายในจอภาพ เดียว ซึ่งจะเป็นไปในลักษณะของโปรแกรมที่อยู่บน Windows ทั้งหมดนั่นเอง



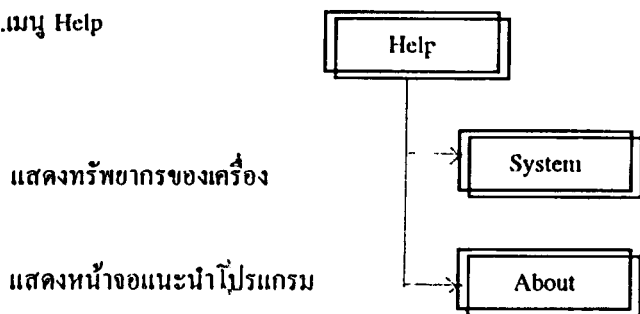
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



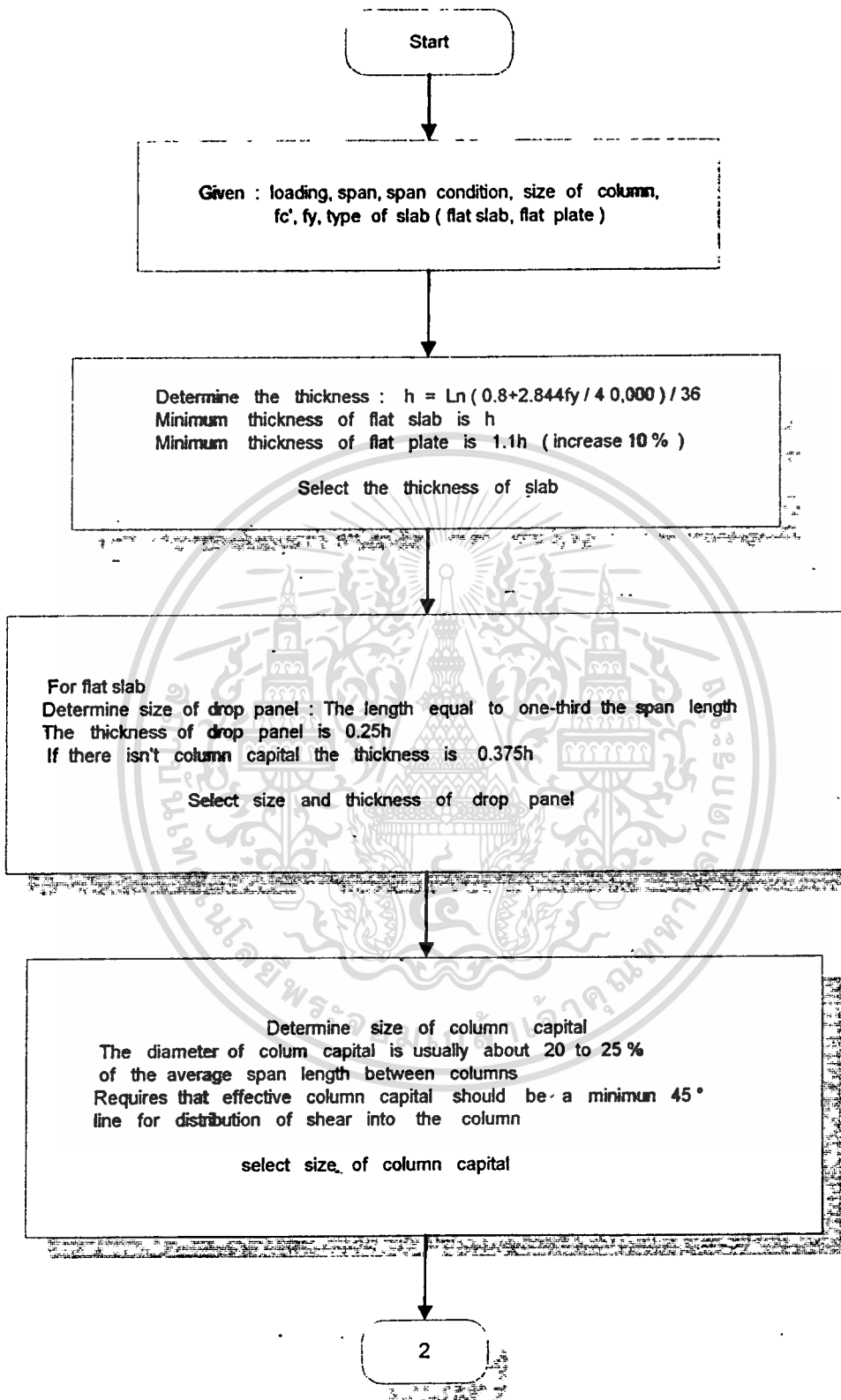
7.เมนู Help



ตอนที่ 3

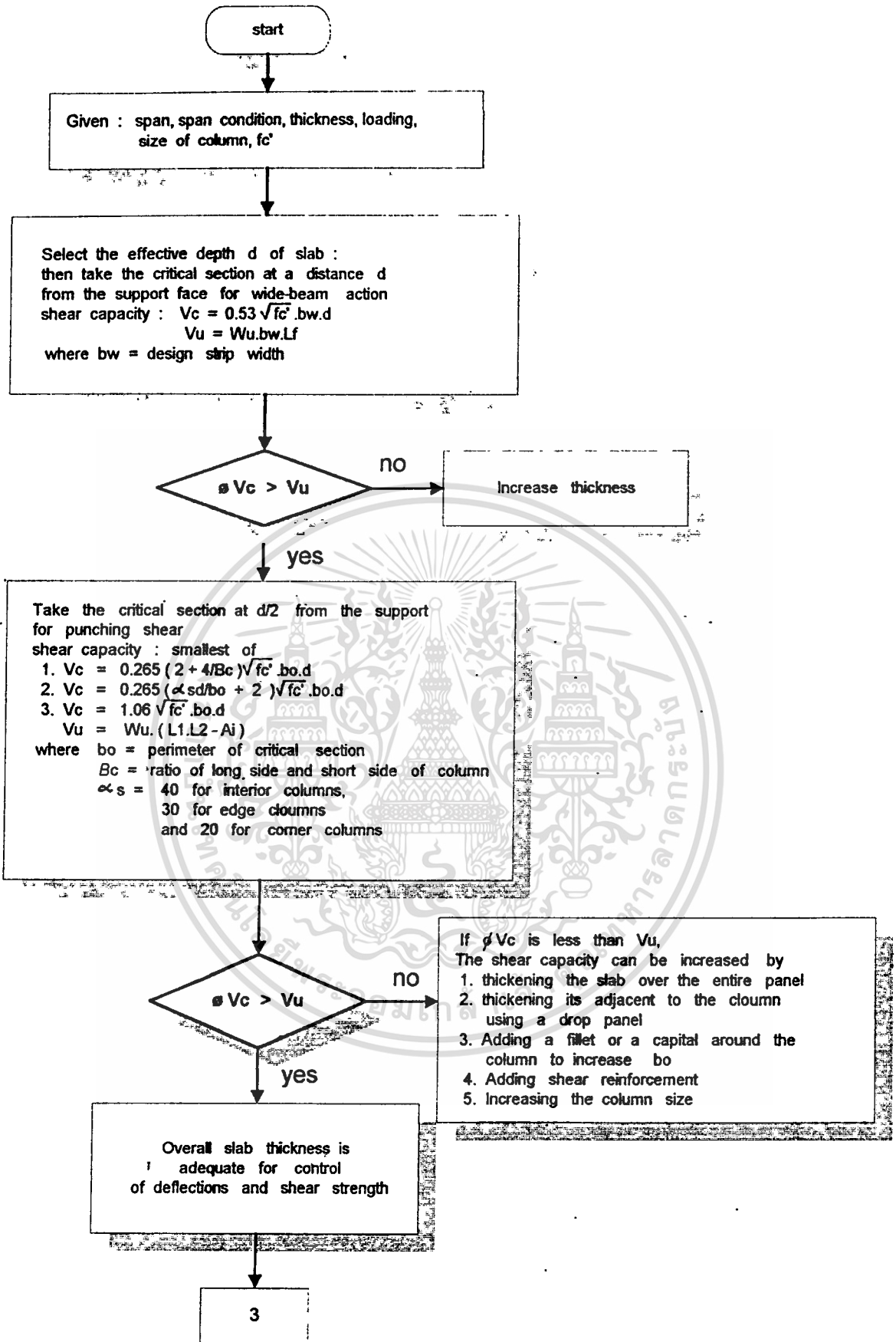
FLOW CHART สำหรับการเขียนโปรแกรม





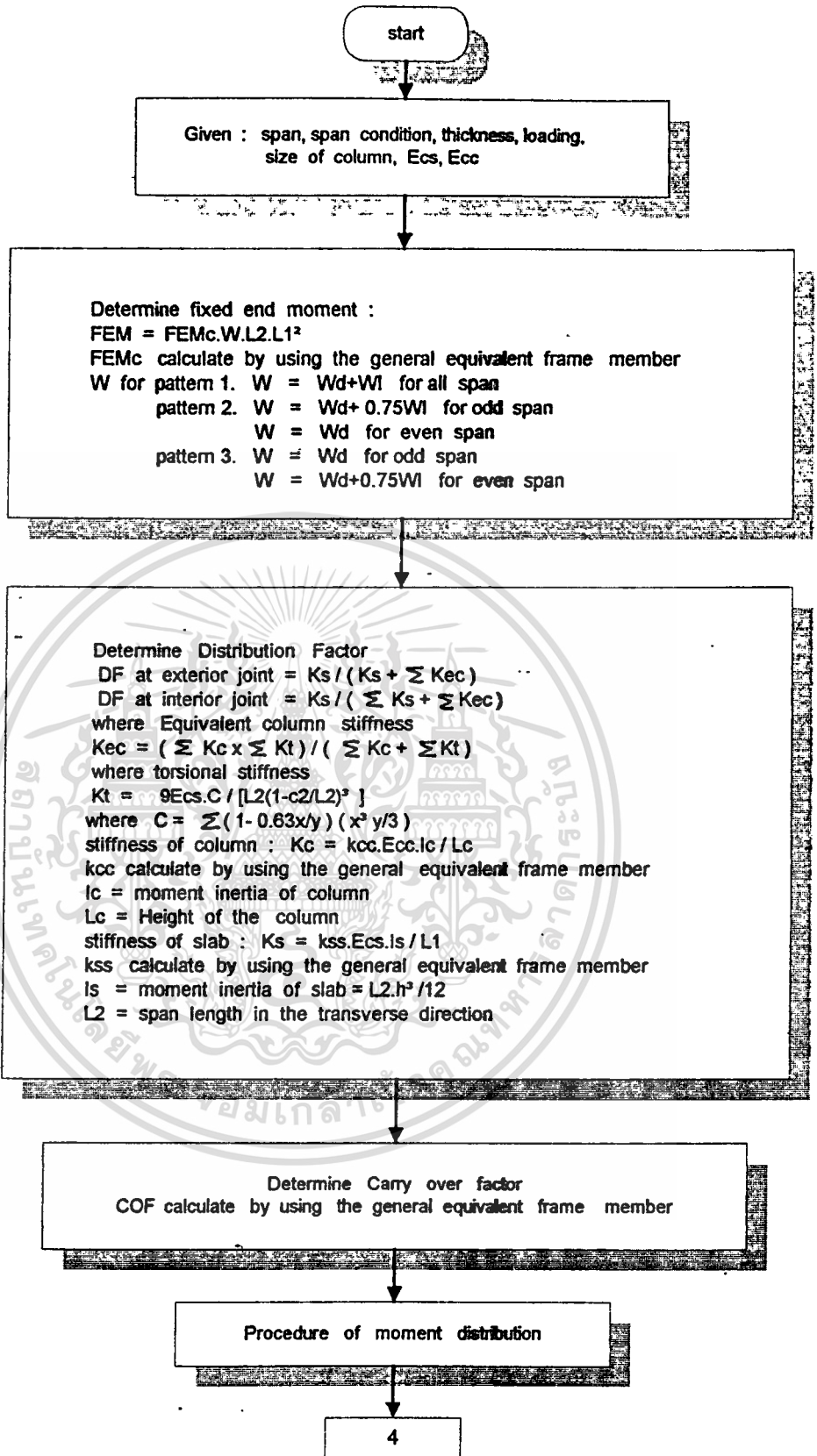
1. FLOW CHART FOR DETERMINE THE THICKNESS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



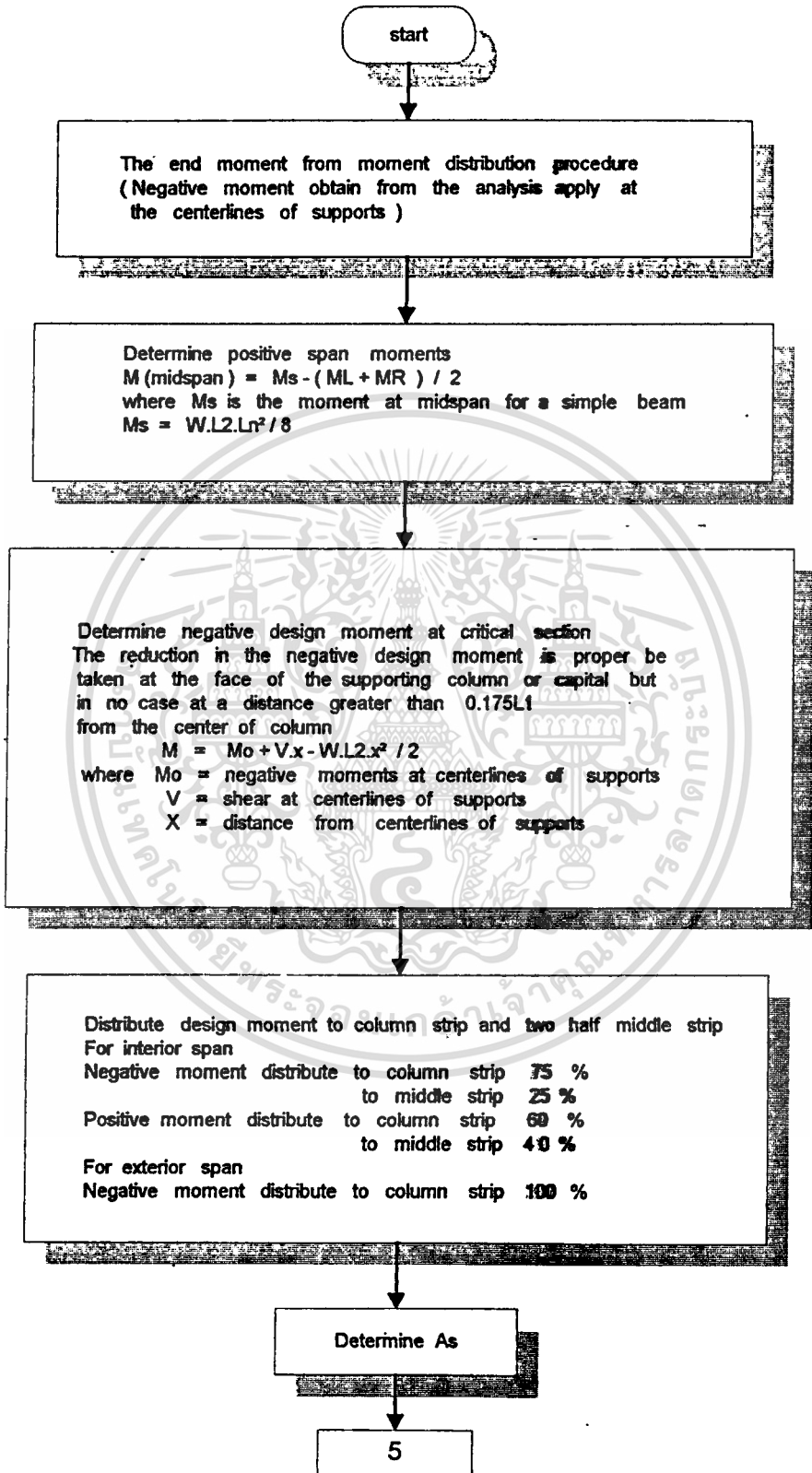
2. FLOW CHART FOR CHECKING SHEAR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



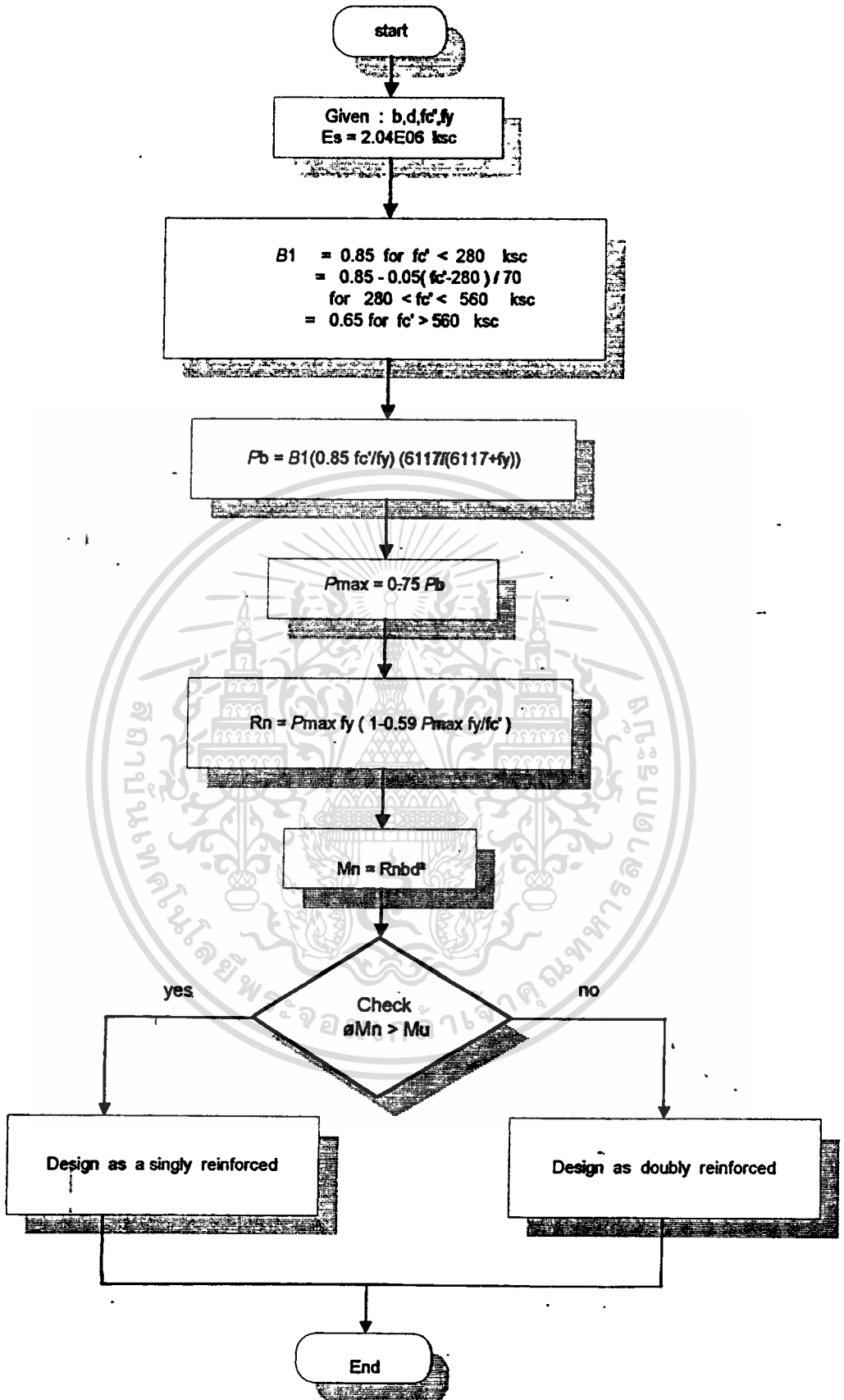
3. FLOW CHART FOR SEQUENCE OF OPERATIONS FOR DETERMINE THE MOMENT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



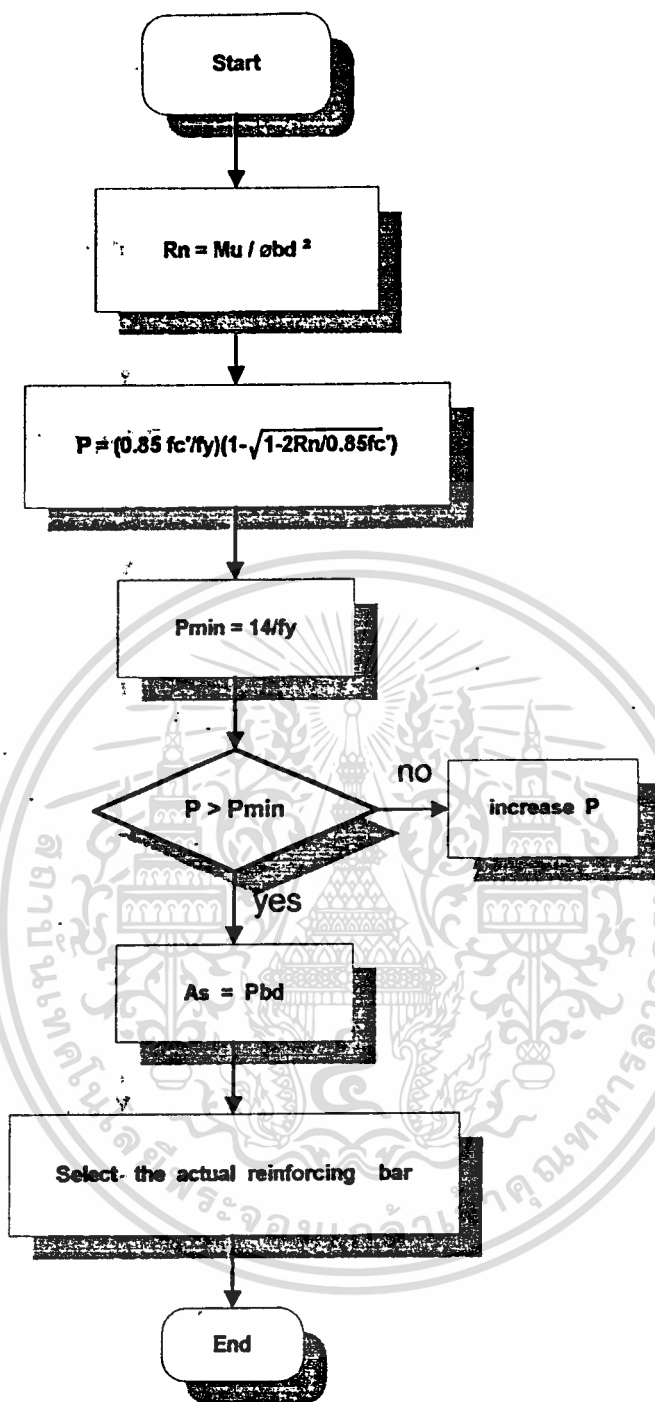
4. FLOW CHART FOR DISTRIBUTION THE DESIGN MOMENT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น เมื่อผู้ใดเห็นประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



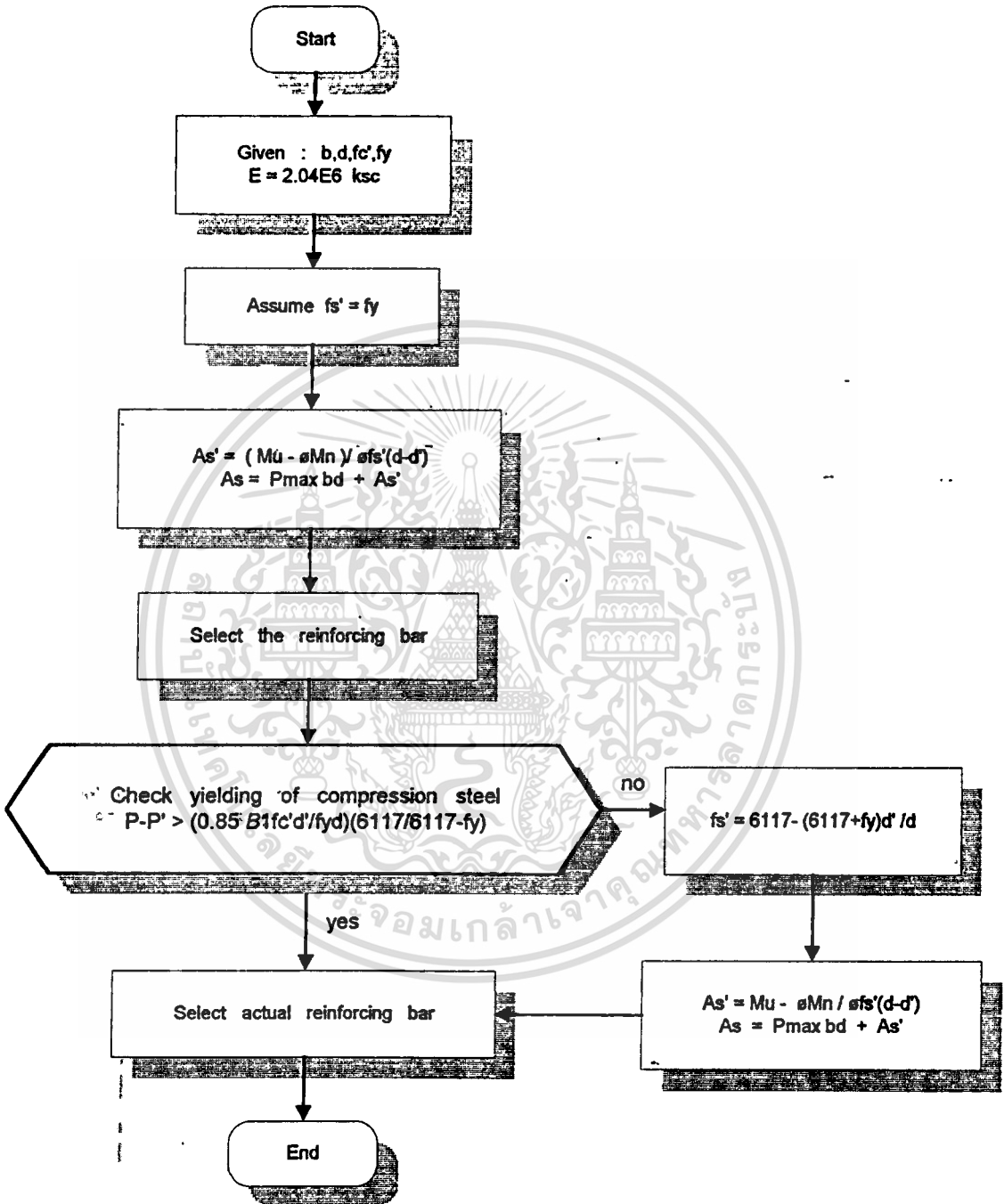
5. FLOW CHART FOR SEQUENCE OF OPERATIONS FOR THE DESIGN OF REINFORCED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



6. FLOW CHART FOR ANALYSIS SINGLY REINFORCED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



7. FLOW CHART FOR ANALYSIS OF DOUBLY REINFORCED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอนที่ 4

ตัวอย่างการคำนวณ

=====

TWO WAY FLAT PLATE SLAB DESIGN

KING MONGLUT INSTITUTE OF TECHNOLOGY LADKRABANG

CML ENGINEERING DEPARTMENT

BY SIAM KUNKATE & PATCHARAWALAI MAITA

LISTING COMPUTATION CHECKING

=====

Data : Sunday 03/31/1996 :: Time : 10:12:45 PM

File Name : Test

Project : Test

Engineer : Dr. Siam Kunkate

Address : 21/11 Moo.3 Ladkrabang Ladkrabang Bangkok 10520

Information : For Test Program

Length Unit : Meter(m) :: Load Unit : Kilogram(Kg)

fc' : 280.00 ksc

fy (Reinforce) : 3000.00 ksc

Ec (Slab) : 79148.04 ksc

Ec (Column) : 79148.04 ksc

Column Data

| Column No. | Co_X(m) | Co_Y(m) | Length(m) |
|------------|---------|---------|-----------|
|------------|---------|---------|-----------|

| | | | |
|---|-------|------|------|
| 1 | 0.00 | 0.00 | 3.00 |
| 2 | 6.00 | 0.00 | 3.00 |
| 3 | 12.00 | 0.00 | 3.00 |
| 4 | 18.00 | 0.00 | 3.00 |
| 5 | 0.00 | 6.00 | 3.00 |
| 6 | 6.00 | 6.00 | 3.00 |
| 7 | 12.00 | 6.00 | 3.00 |
| 8 | 18.00 | 6.00 | 3.00 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Row Element Data

| Row | Element No. | Start Column | End Column | Length(m) | Top-Bay(m) | Bottom-Bay(m) |
|-----|-------------|--------------|------------|-----------|------------|---------------|
| 1 | 1 | 1 | 2 | 6.00 | 0.00 | 6.00 |
| 1 | 2 | 2 | 3 | 6.00 | 0.00 | 6.00 |
| 1 | 3 | 3 | 4 | 6.00 | 0.00 | 6.00 |
| 2 | 1 | 5 | 6 | 6.00 | 6.00 | 0.00 |
| 2 | 2 | 6 | 7 | 6.00 | 6.00 | 0.00 |
| 2 | 3 | 7 | 8 | 6.00 | 6.00 | 0.00 |

Row Element Data

| Column | Element No. | Start Column | End Column | Length(m) | Left-Bay(m) | Right-Bay(m) |
|--------|-------------|--------------|------------|-----------|-------------|--------------|
| 1 | 1 | 1 | 5 | 6.00 | 0.00 | 6.00 |
| 2 | 1 | 2 | 6 | 6.00 | 6.00 | 6.00 |
| 3 | 1 | 3 | 7 | 6.00 | 6.00 | 6.00 |
| 4 | 1 | 4 | 8 | 6.00 | 6.00 | 0.00 |

SLAB SYSTEM : Flat Plate

Overall Slab Thickness = 17.00 cm

Cover of concrete = 2.50 cm

Load factor :::

 $W_u = 1.4DL + 1.7LL$ Unfactored Dead Load = 408.00 kg/m²Unfactored Live Load = 300.00 kg/m²

| Column No. | Punching at Column | | Punching at Drop Panel | | Punching at Capital | | Diagonal shear | |
|------------|--------------------|--------|------------------------|--------|---------------------|--------|----------------|--------|
| | Vc(kg) | Vn(kg) | Vc(kg) | Vn(kg) | Vc(kg) | Vn(kg) | Vc(kg) | Vn(kg) |

| | | | | | | | | |
|---|------------|------------|------------|------------|------------|------------|------------|------------|
| 1 | 1.9161E+04 | 1.1272E+04 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 1.2859E+04 | 2.8684E+04 |
|---|------------|------------|------------|------------|------------|------------|------------|------------|

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่เว้นกรณีใดๆทั้งสิ้น ยกเว้นที่มิมีเหตุที่แบบลงเนื้อหา และต้องจ่ายสิ่งของเงินของเอกสารทุกครั้งที่มีการนำไปใช้

| | | | | | | | | |
|---|------------|------------|------------|------------|------------|------------|------------|------------|
| 2 | 3.0606E+04 | 2.2685E+04 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 1.2859E+04 | 3.2500E+03 |
| 3 | 3.0606E+04 | 2.2685E+04 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 1.2859E+04 | 3.2500E+03 |
| 4 | 1.9161E+04 | 1.1272E+04 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 1.2859E+04 | 2.8684E+03 |
| 5 | 1.9161E+04 | 1.1272E+04 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 1.2859E+04 | 2.8684E+03 |
| 6 | 3.0606E+04 | 2.2685E+04 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 1.2859E+04 | 3.2500E+03 |
| 7 | 3.0606E+04 | 2.2685E+04 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 1.2859E+04 | 3.2500E+03 |
| 8 | 1.9161E+04 | 1.1272E+04 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 1.2859E+04 | 2.8684E+03 |

REINFORCEMENT OF SLAB SEPARATE IN COLUMN STRIP & HALF MIDDLE STRIP
DESIGN MOMENT AT CRITICAL SECTIONS FROM SUPPORTS

Reinforce in Row

| Row No. | Section | Design M(kg-m) | As(cm ²) |
|---------|-------------------------|----------------|----------------------|
| 1 | Left Column Strip Top | 1.1205E+03 | 2.28E+00 |
| 1 | Bottom | 1.1205E+03 | 2.29E+00 |
| 1 | Top Bay Top | 0.0000E+00 | 0.00E+00 |
| 1 | Bottom | 0.0000E+00 | 0.00E+00 |
| 1 | Bottom Bay Top | 0.0000E+00 | 0.00E+00 |
| 1 | Bottom | 0.0000E+00 | 1.01E-03 |
| 1 | Middle Column Strip Top | 5.3694E+03 | 1.09E+01 |
| 1 | Bottom | 5.3694E+03 | 1.09E+01 |
| 1 | Top Bay Top | 3.5796E+03 | 0.00E+00 |
| 1 | Bottom | 3.5796E+03 | 0.00E+00 |
| 1 | Bottom Bay Top | 3.5796E+03 | 7.29E+00 |
| 1 | Bottom | 3.5796E+03 | 7.30E+00 |
| 1 | Right Column Strip Top | -7.8115E+03 | 1.59E+01 |
| 1 | Bottom | -7.8115E+03 | 1.59E+01 |
| 1 | Top Bay Top | -2.6036E+03 | 0.00E+00 |
| 1 | Bottom | -2.6036E+03 | 0.00E+00 |
| 1 | Bottom Bay Top | -2.6036E+03 | 5.31E+00 |
| 1 | Bottom | -2.6036E+03 | 5.30E+00 |

เอกสารนี้เป็นเอกสารที่ส่งไปไว้สำหรับการใช้งานเพื่อกรณีฉุกเฉินเท่านั้น ไม่ควรใช้สำหรับประกอบการตัดสินใจในการดำเนินการใดๆ
หากกรณีใดๆที่เริ่มมีอาการผิดปกติให้ติดต่อช่างเทคนิค และดำเนินการแก้ไขทันที และดำเนินการบันทึกประวัติการดำเนินงานทุกครั้งที่มีการนำไปใช้

1 2 Left Column Strip Top -8.0111E+03 1.63E+01

| | | | | | |
|-------|---|---------------------|--------|-------------|----------|
| 1 | 2 | | Bottom | -8.0111E+03 | 1.63E+01 |
| 1 | 2 | Top Bay | Top | 0.0000E+00 | 0.00E+00 |
| 1 | 2 | | Bottom | 0.0000E+00 | 0.00E+00 |
| 1 | 2 | Bottom Bay | Top | 0.0000E+00 | 0.00E+00 |
| 1 | 2 | | Bottom | 0.0000E+00 | 1.01E-03 |
| 1 | 2 | Middle Column Strip | Top | 9.0007E+03 | 1.63E+01 |
| 1 | 2 | | Bottom | 9.0007E+03 | 1.63E+01 |
| 1 | 2 | Top Bay | Top | 0.0000E+00 | 0.00E+00 |
| 1 | 2 | | Bottom | 0.0000E+00 | 0.00E+00 |
| 1 | 2 | Bottom Bay | Top | 6.0005E+03 | 1.22E+01 |
| 1 | 2 | | Bottom | 6.0005E+03 | 1.22E+01 |
| 1 | 2 | Right Column Strip | Top | -8.0364E+03 | 1.64E+01 |
| 1 | 2 | | Bottom | -8.0364E+03 | 1.64E+01 |
| 1 | 2 | Top Bay | Top | -2.6795E+03 | 0.00E+00 |
| 1 | 2 | | Bottom | -2.6795E+03 | 0.00E+00 |
| 1 | 2 | Bottom Bay | Top | -2.6795E+03 | 5.46E+00 |
| 1 | 2 | | Bottom | -2.6795E+03 | 5.45E+00 |
| ===== | | | | | |
| 1 | 3 | Left Column Strip | Top | -7.7641E+03 | 1.59E+01 |
| 1 | 3 | | Bottom | -7.7641E+03 | 1.59E+01 |
| 1 | 3 | Top Bay | Top | 0.0000E+00 | 0.00E+00 |
| 1 | 3 | | Bottom | 0.0000E+00 | 0.00E+00 |
| 1 | 3 | Bottom Bay | Top | 0.0000E+00 | 0.00E+00 |
| 1 | 3 | | Bottom | 0.0000E+00 | 1.01E-03 |
| 1 | 3 | Middle Column Strip | Top | 1.2632E+04 | 2.57E-01 |
| 1 | 3 | | Bottom | 1.2632E+04 | 2.57E+01 |
| 1 | 3 | Top Bay | Top | 0.0000E+00 | 0.00E+00 |
| 1 | 3 | | Bottom | 0.0000E+00 | 0.00E+00 |
| 1 | 3 | Bottom Bay | Top | 8.4214E+03 | 1.72E+01 |
| 1 | 3 | | Bottom | 8.4214E+03 | 1.72E+01 |
| 1 | 3 | Right Column Strip | Top | 1.0640E+03 | 2.20E+00 |
| 1 | 3 | | Bottom | 1.0640E+03 | 2.21E+00 |
| 1 | 3 | Top Bay | Top | 0.0000E+00 | 0.00E+00 |
| 1 | 3 | | Bottom | 0.0000E+00 | 0.00E+00 |
| 1 | 3 | Bottom Bay | Top | 0.0000E+00 | 0.00E+00 |
| 1 | 3 | | Bottom | 0.0000E+00 | 1.01E-03 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ควรกรณใดๆทั้งสิ้น อีกทั้งห้ามเผยแพร่เปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | | | | | |
|-------|---|--------|--------------|--------|-------------|----------|
| 2 | 1 | Left | Column Strip | Top | 1.1205E+03 | 2.28E+00 |
| 2 | 1 | | | Bottom | 1.1205E+03 | 2.29E+00 |
| 2 | 1 | | Top Bay | Top | 0.0000E+00 | 0.00E+00 |
| 2 | 1 | | | Bottom | 0.0000E+00 | 1.01E-03 |
| 2 | 1 | | Bottom Bay | Top | 0.0000E+00 | 0.00E+00 |
| 2 | 1 | | | Bottom | 0.0000E+00 | 0.00E+00 |
| 2 | 1 | Middle | Column Strip | Top | 5.3694E+03 | 1.09E+01 |
| 2 | 1 | | | Bottom | 5.3694E+03 | 1.09E+01 |
| 2 | 1 | | Top Bay | Top | 3.5796E+03 | 7.29E+00 |
| 2 | 1 | | | Bottom | 3.5796E+03 | 7.30E+00 |
| 2 | 1 | | Bottom Bay | Top | 3.5796E+03 | 0.00E+00 |
| 2 | 1 | | | Bottom | 3.5796E+03 | 0.00E+00 |
| 2 | 1 | Right | Column Strip | Top | -7.8115E+03 | 1.59E+01 |
| 2 | 1 | | | Bottom | -7.8115E+03 | 1.59E+01 |
| 2 | 1 | | Top Bay | Top | -2.6038E+03 | 5.31E+00 |
| 2 | 1 | | | Bottom | -2.6038E+03 | 5.30E+00 |
| 2 | 1 | | Bottom Bay | Top | -2.6038E+03 | 0.00E+00 |
| 2 | 1 | | | Bottom | -2.6038E+03 | 0.00E+00 |
| ===== | | | | | | |
| 2 | 2 | Left | Column Strip | Top | -8.0111E+03 | 1.63E+01 |
| 2 | 2 | | | Bottom | -8.0111E+03 | 1.63E+01 |
| 2 | 2 | | Top Bay | Top | 0.0000E+00 | 0.00E+00 |
| 2 | 2 | | | Bottom | 0.0000E+00 | 1.01E-03 |
| 2 | 2 | | Bottom Bay | Top | 0.0000E+00 | 0.00E+00 |
| 2 | 2 | | | Bottom | 0.0000E+00 | 0.00E+00 |
| 2 | 2 | Middle | Column Strip | Top | 9.0007E+03 | 1.83E+01 |
| 2 | 2 | | | Bottom | 9.0007E+03 | 1.83E+01 |
| 2 | 2 | | Top Bay | Top | 0.0000E+00 | 0.00E+00 |
| 2 | 2 | | | Bottom | 0.0000E+00 | 1.01E-03 |
| 2 | 2 | | Bottom Bay | Top | 6.0005E+03 | 0.00E+00 |
| 2 | 2 | | | Bottom | 6.0005E+03 | 0.00E+00 |
| 2 | 2 | Right | Column Strip | Top | -5.0384E+03 | 1.64E+01 |
| 2 | 2 | | | Bottom | -5.0384E+03 | 1.64E+01 |
| 2 | 2 | | Top Bay | Top | -2.6795E+03 | 5.46E+00 |
| 2 | 2 | | | Bottom | -2.6795E+03 | 5.45E+00 |
| 2 | 2 | | Bottom Bay | Top | -2.6795E+03 | 0.00E+00 |
| 2 | 2 | | | Bottom | -2.6795E+03 | 0.00E+00 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ทางการศึกษาเท่านั้น ไม่ควรนำข้อมูลนี้ไปใช้ประโยชน์ด้านการค้า
 ไม่ควรนำข้อมูลนี้ไปใช้เพื่อการค้าหรือเพื่อประโยชน์อื่นใดโดยไม่ได้รับอนุญาต

| | | | | | | |
|---|---|--------|--------------|--------|-------------|----------|
| 2 | 3 | Left | Column Strip | Top | -7.7841E+03 | 1.59E+01 |
| 2 | 3 | | | Bottom | -7.7841E+03 | 1.59E+01 |
| 2 | 3 | | Top Bay | Top | 0.0000E+00 | 0.00E+00 |
| 2 | 3 | | | Bottom | 0.0000E+00 | 1.01E-03 |
| 2 | 3 | | Bottom Bay | Top | 0.0000E+00 | 0.00E+00 |
| 2 | 3 | | | Bottom | 0.0000E+00 | 0.00E+00 |
| 2 | 3 | Middle | Column Strip | Top | 1.2632E+04 | 2.57E+01 |
| 2 | 3 | | | Bottom | 1.2632E+04 | 2.57E+01 |
| 2 | 3 | | Top Bay | Top | 0.0000E+00 | 0.00E+00 |
| 2 | 3 | | | Bottom | 0.0000E+00 | 1.01E-03 |
| 2 | 3 | | Bottom Bay | Top | 8.4214E+03 | 0.00E+00 |
| 2 | 3 | | | Bottom | 8.4214E+03 | 0.00E+00 |
| 2 | 3 | Right | Column Strip | Top | 1.0840E+03 | 2.20E+00 |
| 2 | 3 | | | Bottom | 1.0840E+03 | 2.21E+00 |
| 2 | 3 | | Top Bay | Top | 0.0000E+00 | 0.00E+00 |
| 2 | 3 | | | Bottom | 0.0000E+00 | 1.01E-03 |
| 2 | 3 | | Bottom Bay | Top | 0.0000E+00 | 0.00E+00 |
| 2 | 3 | | | Bottom | 0.0000E+00 | 0.00E+00 |

Reinforce in Column

| Row | No. | Section | Design M(kg-m) | As(cm ²) | | |
|-----|-----|---------|----------------|----------------------|------------|----------|
| 1 | 1 | Left | Column Strip | Top | 1.3666E+03 | 2.82E+00 |
| 1 | 1 | | | Bottom | 1.3666E+03 | 2.83E+00 |
| 1 | 1 | | Left Bay | Top | 0.0000E+00 | 0.00E+00 |
| 1 | 1 | | | Bottom | 0.0000E+00 | 0.00E+00 |
| 1 | 1 | | Right Bay | Top | 0.0000E+00 | 0.00E+00 |
| 1 | 1 | | | Bottom | 0.0000E+00 | 1.01E-03 |
| 1 | 1 | Middle | Column Strip | Top | 9.0007E+03 | 1.83E+01 |
| 1 | 1 | | | Bottom | 9.0007E+03 | 1.83E+01 |
| 1 | 1 | | Left Bay | Top | 6.0005E+03 | 0.00E+00 |
| 1 | 1 | | | Bottom | 6.0005E+03 | 0.00E+00 |
| 1 | 1 | | Right Bay | Top | 6.0005E+03 | 1.22E+01 |
| 1 | 1 | | | Bottom | 6.0005E+03 | 1.22E+01 |
| 1 | 1 | Right | Column Strip | Top | 1.3666E+03 | 2.82E+00 |
| 1 | 1 | | | Bottom | 1.3666E+03 | 2.83E+00 |
| 1 | 1 | | Left Bay | Top | 0.0000E+00 | 0.00E+00 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ของนักศึกษาที่เข้าศึกษาที่มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ห้ามนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังให้ตัดแปลงเนื้อหา และต่อ 0.0000E+00 0.00E+00 การทุกครั้งที่มีการนำไปใช้

| | | | | | |
|---|---|-----------|--------|------------|----------|
| 1 | 1 | | Bottom | 0.0000E+00 | 0.00E+00 |
| 1 | 1 | Right Bay | Top | 0.0000E+00 | 0.00E+00 |
| 1 | 1 | | Bottom | 0.0000E+00 | 1.01E-03 |

| | | | | | |
|---|---|---------------------|--------|------------|----------|
| 2 | 1 | Left Column Strip | Top | 2.7733E+03 | 5.64E+00 |
| 2 | 1 | | Bottom | 2.7733E+03 | 5.66E+00 |
| 2 | 1 | Left Bay | Top | 0.0000E+00 | 0.00E+00 |
| 2 | 1 | | Bottom | 0.0000E+00 | 1.01E-03 |
| 2 | 1 | Right Bay | Top | 0.0000E+00 | 0.00E+00 |
| 2 | 1 | | Bottom | 0.0000E+00 | 1.01E-03 |
| 2 | 1 | Middle Column Strip | Top | 1.8001E+04 | 3.67E+01 |
| 2 | 1 | | Bottom | 1.8001E+04 | 3.67E+01 |
| 2 | 1 | Left Bay | Top | 6.0005E+03 | 1.22E+01 |
| 2 | 1 | | Bottom | 6.0005E+03 | 1.22E+01 |
| 2 | 1 | Right Bay | Top | 6.0005E+03 | 1.22E+01 |
| 2 | 1 | | Bottom | 6.0005E+03 | 1.22E+01 |
| 2 | 1 | Right Column Strip | Top | 2.7733E+03 | 5.64E+00 |
| 2 | 1 | | Bottom | 2.7733E+03 | 5.66E+00 |
| 2 | 1 | Left Bay | Top | 0.0000E+00 | 0.00E+00 |
| 2 | 1 | | Bottom | 0.0000E+00 | 1.01E-03 |
| 2 | 1 | Right Bay | Top | 0.0000E+00 | 0.00E+00 |
| 2 | 1 | | Bottom | 0.0000E+00 | 1.01E-03 |

| | | | | | |
|---|---|---------------------|--------|------------|----------|
| 3 | 1 | Left Column Strip | Top | 2.7733E+03 | 5.64E+00 |
| 3 | 1 | | Bottom | 2.7733E+03 | 5.66E+00 |
| 3 | 1 | Left Bay | Top | 0.0000E+00 | 0.00E+00 |
| 3 | 1 | | Bottom | 0.0000E+00 | 1.01E-03 |
| 3 | 1 | Right Bay | Top | 0.0000E+00 | 0.00E+00 |
| 3 | 1 | | Bottom | 0.0000E+00 | 1.01E-03 |
| 3 | 1 | Middle Column Strip | Top | 1.8001E+04 | 3.67E+01 |
| 3 | 1 | | Bottom | 1.8001E+04 | 3.67E+01 |
| 3 | 1 | Left Bay | Top | 6.0005E+03 | 1.22E+01 |
| 3 | 1 | | Bottom | 6.0005E+03 | 1.22E+01 |
| 3 | 1 | Right Bay | Top | 6.0005E+03 | 1.22E+01 |
| 3 | 1 | | Bottom | 6.0005E+03 | 1.22E+01 |
| 3 | 1 | Right Column Strip | Top | 2.7733E+03 | 5.64E+00 |
| 3 | 1 | | Bottom | 2.7733E+03 | 5.66E+00 |
| 3 | 1 | Left Bay | Top | 0.0000E+00 | 0.00E+00 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรรมใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงข้อมูลเอกสารทุกครั้งที่มีการนำไปใช้

| | | | | | |
|---|---|-----------|--------|------------|----------|
| 3 | 1 | | Bottom | 0.0000E+00 | 1.01E-03 |
| 3 | 1 | Right Bay | Top | 0.0000E+00 | 0.00E+00 |
| 3 | 1 | | Bottom | 0.0000E+00 | 1.01E-03 |

| | | | | | |
|---|---|---------------------|--------|------------|----------|
| 4 | 1 | Left Column Strip | Top | 1.3666E+03 | 2.82E+00 |
| 4 | 1 | | Bottom | 1.3666E+03 | 2.83E+00 |
| 4 | 1 | Left Bay | Top | 0.0000E+00 | 0.00E+00 |
| 4 | 1 | | Bottom | 0.0000E+00 | 1.01E-03 |
| 4 | 1 | Right Bay | Top | 0.0000E+00 | 0.00E+00 |
| 4 | 1 | | Bottom | 0.0000E+00 | 0.00E+00 |
| 4 | 1 | Middle Column Strip | Top | 9.0007E+03 | 1.83E+01 |
| 4 | 1 | | Bottom | 9.0007E+03 | 1.83E+01 |
| 4 | 1 | Left Bay | Top | 6.0005E+03 | 1.22E+01 |
| 4 | 1 | | Bottom | 6.0005E+03 | 1.22E+01 |
| 4 | 1 | Right Bay | Top | 6.0005E+03 | 0.00E+00 |
| 4 | 1 | | Bottom | 6.0005E+03 | 0.00E+00 |
| 4 | 1 | Right Column Strip | Top | 1.3666E+03 | 2.82E+00 |
| 4 | 1 | | Bottom | 1.3666E+03 | 2.83E+00 |
| 4 | 1 | Left Bay | Top | 0.0000E+00 | 0.00E+00 |
| 4 | 1 | | Bottom | 0.0000E+00 | 1.01E-03 |
| 4 | 1 | Right Bay | Top | 0.0000E+00 | 0.00E+00 |
| 4 | 1 | | Bottom | 0.0000E+00 | 0.00E+00 |

DIMENSION OF DROP PANEL & COLUMN CAPITAL

| No. | Drop Width | Drop Length | Drop Thick | Capital Width | Capital Length |
|-----|------------|-------------|------------|---------------|----------------|
| 1 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 |
| 2 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 |
| 3 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 |
| 4 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 |
| 5 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | | | | |
|---|------------|------------|------------|------------|------------|
| 6 | 0 0000E+00 | 0 0000E+00 | 0 0000E+00 | 0 0000E+00 | 0 0000E+00 |
| 7 | 0 0000E+00 | 0.0000E+00 | 0 0000E+00 | 0 0000E+00 | 0 0000E+00 |
| 8 | 0 0000E+00 | 0.0000E+00 | 0 0000E+00 | 0 0000E+00 | 0 0000E+00 |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทสรุปและวิจารณ์

เนื่องจากการออกแบบพื้นระบบแผ่นพื้นไร้คานเป็นงานที่ค่อนข้างจะซับซ้อนมาก และผู้ที่มีความรู้ความเชี่ยวชาญทางด้านนี้ยังมีอยู่น้อย ดังนั้นวัตถุประสงค์ในการทำโครงการครั้งนี้ ก็เพื่อรวบรวมข้อมูลในการออกแบบที่ถูกต้องนำมาเขียนเป็นโปรแกรมคอมพิวเตอร์โดยอาศัยโปรแกรมที่มีประสิทธิภาพและมีความยืดหยุ่นในการทำงาน เพื่อให้การออกแบบสามารถทำได้ง่ายและสะดวกขึ้น โดยทางกลุ่มโครงการจะจำกัดขอบเขตการทำงานของโปรแกรมเพื่อความเหมาะสมในการทำงาน และคำนึงถึงการออกแบบในรูปแบบการใช้งานที่สามารัลดความยุ่งยากในการใช้งานจริง

โครงการนี้พัฒนาโปรแกรมวิเคราะห์โครงสร้างบนระบบปฏิบัติการ Windows ซึ่งเป็นระบบปฏิบัติการที่นิยมใช้กันอย่างกว้างขวางในปัจจุบัน - และถ้าหากต้องการ โปรแกรมที่มีความละเอียดในการคำนวณที่มากกว่านี้ก็สามารถนำโครงการนี้ไปพัฒนาขั้นตอนในการคำนวณได้ แต่ก็คงต้องใช้เวลาและความสามารถในการศึกษาเรื่องการเขียนโปรแกรมด้วยภาษา Visual BASIC เพิ่มเติม ทั้งนี้ผู้ทำการออกแบบจะต้องมีพื้นฐานความรู้และประสบการณ์ในการออกแบบพอสมควรด้วย

ข้อดีของโปรแกรมนี้คือสามารถแสดงผลเป็นกราฟิก และรายการแสดงผลซึ่งจะทำให้สามารถเข้าใจพฤติกรรมของแผ่นพื้นและเห็นภาพเป็นรูปธรรมมากยิ่งขึ้น

สำหรับข้อบกพร่องของโปรแกรมคือ ระยะช่วงพื้นที่ออกแบบควรจะมีค่าแตกต่างกันไม่มาก เพื่อให้ความหนาของแผ่นพื้นที่ได้จากการวิเคราะห์ มีค่าใกล้เคียงกันหรือเท่ากันตลอดช่วงพื้น เพื่อให้การวิเคราะห์ในส่วนอื่นๆคือการตรวจสอบแรงเฉือนและการหาค่าโมเมนต์ได้ผลลัพธ์ที่ถูกต้องมากที่สุด และเนื่องจากการวิเคราะห์โดยวิธีโครงเปรียบเทียบถือว่าเป็นการวิเคราะห์โครงข้อแข็งที่มีความต่อเนื่องดังนั้นแผ่นต้องมีความต่อเนื่องอย่างน้อยสามช่วงติดต่อกันในแต่ละทิศทาง

ปัญหาที่เกิดขึ้นในการทำโครงการนี้คือ ปัญหาของตัวภาษาที่ใช้เขียน เนื่องจากภาษา Visual BASIC นั้นยังไม่เป็นที่แพร่หลายมากนัก จึงมีแหล่งข้อมูลให้ศึกษาน้อยและต้องใช้เวลาศึกษานานพอสมควร และเนื่องจากเวลาในการทำโครงการนี้มีจำกัดเป็นอย่างมากจึงทำให้ไม่สามารถทำการวางแผนการเขียนโปรแกรมให้เป็นไปโดยมีระบบและระเบียบตามที่ควรจะเป็น ทำให้การเขียนโปรแกรมเป็นไปโดยยากลำบาก การตรวจสอบและแก้ไขโปรแกรมเป็นไปด้วยความยากลำบากเนื่องจากไม่สามารถตรวจสอบโปรแกรมทีละขั้นตอนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และจากการที่ตัวโปรแกรมมีขนาดใหญ่ มีตัวแปรมาก ทำให้ใช้หน่วยความจำสูงตามไปด้วย

นอกจากปัญหาในการเขียนโปรแกรมการคำนวณแล้ว ยังมีปัญหาในเรื่องของการออกแบบทางโยธาด้วย เพราะการออกแบบส่วนใหญ่ใช้การตัดสินใจของวิศวกรในการเลือกว่าต้องการจะออกแบบแบบใด ยกตัวอย่างเช่นการเลือกเหล็กเสริม โดยปกติแล้วผู้ออกแบบจะต้องเป็นผู้ตัดสินใจว่าจะต้องใช้เหล็กขนาดใดและระยะห่างเท่าใดจึงจะเหมาะสม

ทั้งหมดนี้คือข้อสรุปและข้อเสนอแนะสำหรับผู้ที่มีความสนใจ ซึ่งถ้าหากมีความรู้ความสามารถทางด้านคอมพิวเตอร์จะได้พัฒนาโปรแกรมให้มีความสมบูรณ์ต่อไป



ภาคผนวก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAM LISTING



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMABOUT.FRM - 1

```
Sub cmdOK_Click ()  
    frmAbout.Visible = False  
    Unload frmAbout  
End Sub
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMABOUT.FRM - 1

VERSION 2.00

Begin Form frmAbout

```

AutoRedraw      = -1 'True
BackColor       = &H00C0C0C0&
BorderStyle    = 1 'Fixed Single
Caption        = "About Program"
ClientHeight   = 6930
ClientLeft     = 1665
ClientTop      = 735
ClientWidth    = 9570
ClipControls   = 0 'False
ControlBox     = 0 'False
Height         = 7335
Left           = 1605
LinkTopic      = "Form1"
ScaleHeight    = 6930
ScaleWidth     = 9570
Top            = 390
Width          = 9690

```

Begin SSPanel Panel3D5

```

BevelOuter     = 0 'None
Caption        = "PATCHARAWALAI MAITA 35104"
Font3D         = 3 'Inset w/light shading
ForeColor      = &H00FF0000&
Height         = 255
Left           = 240
TabIndex       = 8
Top            = 5160
Width          = 9135

```

End

Begin SSPanel Panel3D4

```

BevelOuter     = 0 'None
Caption        = "ADVISOR: MR. WIBOOL VOOTIYAN"
Font3D         = 3 'Inset w/light shading
ForeColor      = &H00FF0000&
Height         = 495
Left           = 240
TabIndex       = 7
Top            = 5640
Width          = 9255

```

End

Begin SSPanel Panel3D3

```

BevelOuter     = 0 'None
Caption        = "SOMPONG KONGKRAPAN 35104443"
Font3D         = 3 'Inset w/light shading
ForeColor      = &H00FF0000&
Height         = 255
Left           = 240
TabIndex       = 6
Top            = 4680
Width          = 9135

```

End

Begin SSPanel Panel3D2

```

BevelOuter     = 0 'None
Caption        = "SIAM KUNKATE 35104449"
Font3D         = 3 'Inset w/light shading
ForeColor      = &H00FF0000&

```

FRMABOUT.FRM - 2

```

Height      = 495
Left        = 240
TabIndex    = 5
Top         = 4080
Width       = 9135

```

End

Begin SSPanel Panel3D1

```

BevelOuter  = 0 'None
Caption     = "DIRECTED BY"
Font3D      = 3 'Inset w/light shading
FontBold    = -1 'True
FontItalic  = 0 'False
FontName    = "MS Sans Serif"
FontSize    = 8.25
FontStrikethru = 0 'False
FontUnderline = 0 'False
ForeColor   = &H00FF0000&
Height      = 495
Index       = 3
Left        = 360
TabIndex    = 4
Top         = 3480
Width       = 8895

```

End

Begin SSPanel Panel3D1

```

BevelOuter  = 0 'None
Caption     = "SLAB DESIGN IN PRE-STRESSED && FLAT SLAB AND FLA
Font3D      = 3 'Inset w/light shading
FontBold    = -1 'True
FontItalic  = 0 'False
FontName    = "MS Sans Serif"
FontSize    = 12
FontStrikethru = 0 'False
FontUnderline = 0 'False
ForeColor   = &H00FF0000&
Height      = 975
Index       = 2
Left        = 240
TabIndex    = 3
Top         = 2400
Width       = 8895

```

End

Begin SSPanel Panel3D1

```

BevelOuter  = 0 'None
Caption     = "PFES"
Font3D      = 3 'Inset w/light shading
FontBold    = -1 'True
FontItalic  = 0 'False
FontName    = "Courier"
FontSize    = 18.75
FontStrikethru = 0 'False
FontUnderline = 0 'False
ForeColor   = &H000000FF&
Height      = 735
Index       = 1
Left        = 240
TabIndex    = 2

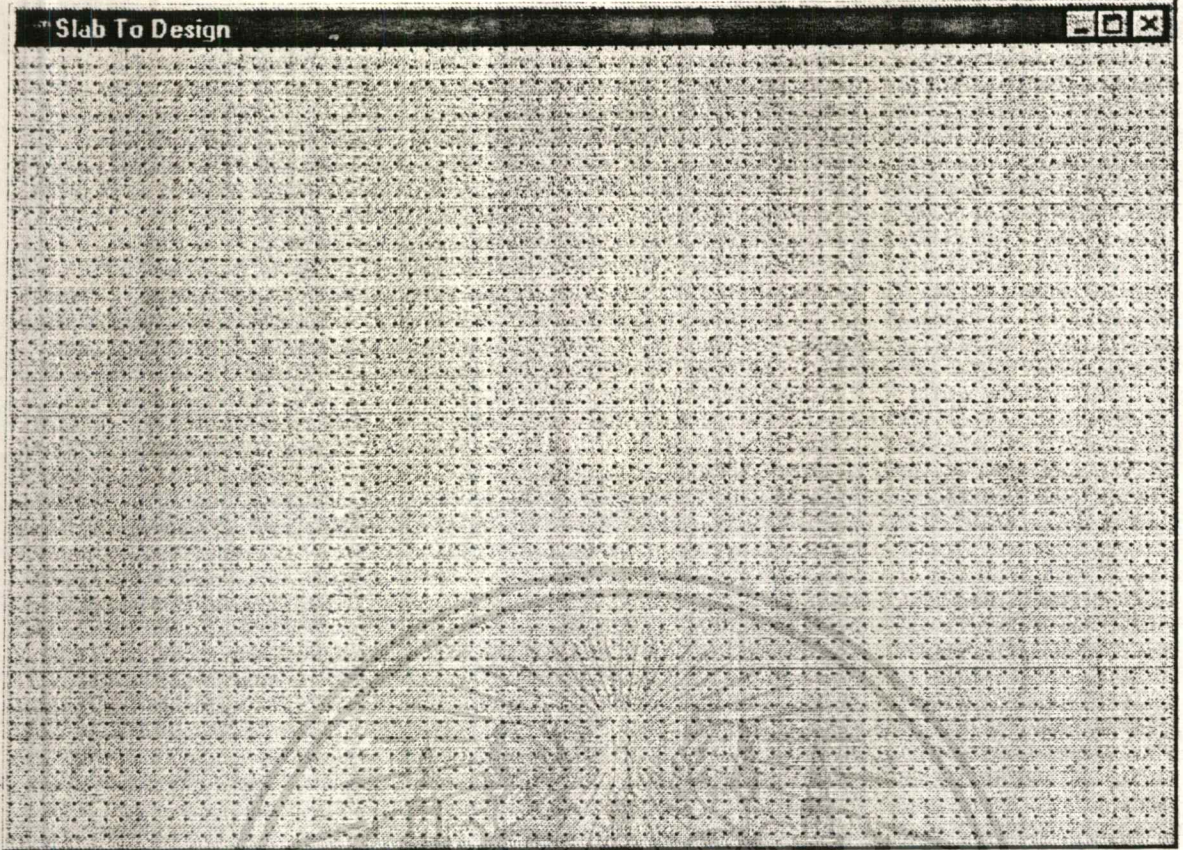
```

FRMABOUT.FRM - 3

```

Top           = 1440
Width        = 8895
End
Begin SSPanel Panel3D1
  BevelOuter  = 0 'None
  Caption     = "KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABAN"
  Font3D     = 3 'Inset w/light shading
  FontBold   = -1 'True
  FontItalic  = 0 'False
  FontName    = "MS Sans Serif"
  FontSize   = 14.25
  FontStrikethru = 0 'False
  FontUnderline = 0 'False
  ForeColor   = &H00FF0000&
  Height     = 975
  Index      = 0
  Left       = 360
  TabIndex   = 1
  Top       = 240
  Width     = 8895
End
Begin SSCommand cmdOK
  Caption     = "&OK"
  FontBold   = -1 'True
  FontItalic  = 0 'False
  FontName    = "Arial"
  FontSize   = 12
  FontStrikethru = 0 'False
  FontUnderline = 0 'False
  ForeColor   = &H00FF0000&
  Height     = 495
  Left       = 4080
  Picture     = (none)
  TabIndex   = 0
  Top       = 6360
  Width     = 1575
End
End

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMDATAN.FRM - 1

VERSION 2.00

Begin Form frmDataInfo

```

AutoRedraw      = -1 'True
BackColor       = &H0080FF80&
Caption        = "Slab To Design"
ClientHeight    = 4485
ClientLeft     = 2070
ClientTop      = 2325
ClientWidth    = 7140
Height         = 4890
Icon           = (Icon)
Left          = 2010
LinkTopic     = "frmDataName"
MDIChild      = -1 'True
ScaleHeight   = 4485
ScaleWidth    = 7140
Top           = 1980
Width         = 7260
WindowState   = 2 'Maximized

```

End



FRMDATAN.FRM - 1

```
Sub Form_QueryUnload (Cancel As Integer, UnloadMode As Integer)
  Ready_ExitProject
End Sub
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Civil Design

File Input Design Result Option Window Help

Saturday 05/11/1996 11:28:00 PM

General Data

Slab System
 Flat Slab
 Flat Plate

Column Head
 Drop panel
 Capital

Strand Property
 f_t 19000 Kg/Mg

Steel Property
 F_y 2400 Ksc

Concrete Properties

| | Slab | Column | |
|----------------|------|--------|-----|
| f _c | 375 | 375 | Ksc |
| E _c | 375 | 375 | Ksc |

Strand System
 Bond
 Unbond

Slab Thk. cm

Strand Type

Ready....

SystemTool DataTool

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 1

VERSION 2.00

```

Begin MDIForm frmMain
  Caption      = "Civil Design"
  ClientHeight = 2655
  ClientLeft   = -90
  ClientTop    = 3915
  ClientWidth  = 9600
  Height       = 3345
  Icon         = (Icon)
  Left         = -150
  LinkTopic    = "frmMain"
  Top          = 3285
  Width        = 9720
  WindowState  = 2 'Maximized
Begin SSPanel pnlTextInputData
  Align        = 2 'Align Bottom
  BevelInner   = 1 'Inset
  ForeColor    = &H000000FF&
  Height       = 2070
  Left         = 0
  Outline      = -1 'True
  ShadowColor  = 1 'Black
  TabIndex     = 12
  Top          = -330
  Width        = 9600
Begin SSFrame fmeTextInputData
  Caption      = "Node Data"
  Font3D       = 2 'Raised w/heavy shading
  FontBold     = -1 'True
  FontItalic   = 0 'False
  FontName     = "MS Sans Serif"
  FontSize     = 9.75
  FontStrikethru = 0 'False
  FontUnderline = 0 'False
  ForeColor    = &H00008000&
  Height       = 1815
  Index        = 0
  Left         = 9480
  ShadowStyle  = 1 'Raised
  TabIndex     = 102
  Top          = 300
  Visible      = 0 'False
  Width        = 8955
Begin SSPanel pnlLengthUnit
  BevelInner   = 2 'Raised
  BevelOuter   = 0 'None
  Caption      = "m."
  Font3D       = 1 'Raised w/light shading
  ForeColor    = &H00000080&
  Height       = 315
  Index        = 8
  Left         = 2580
  TabIndex     = 109
  Top          = 1020
  Width        = 315
End
Begin TextBox txtNodeLength

```

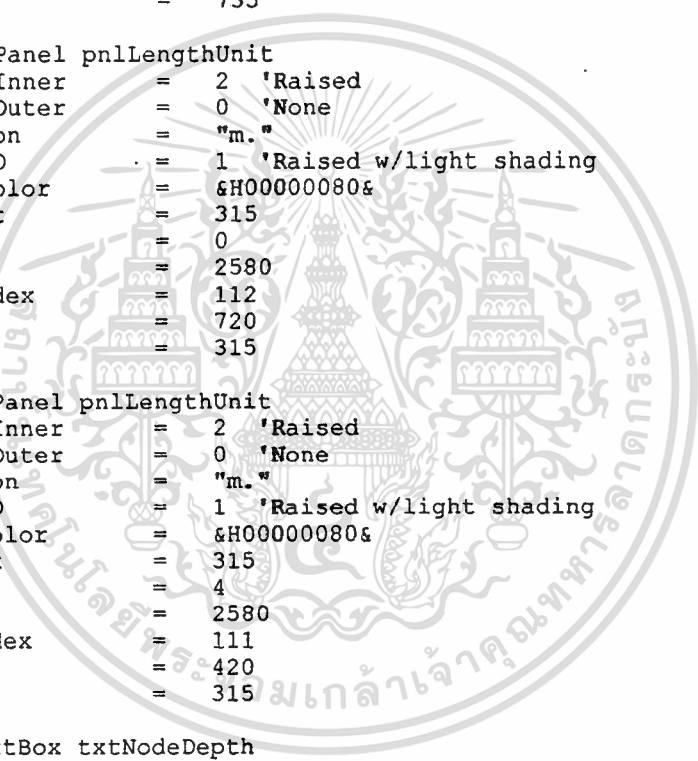
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 2

```

Height          = 315
Left            = 2160
TabIndex       = 110
Text           = "3"
Top            = 1020
Width          = 375
End
Begin SSPanel  pnlColumnLength
  BevelInner    = 2 'Raised
  BevelOuter    = 0 'None
  Caption       = "Length"
  Font3D        = 1 'Raised w/light shading
  ForeColor     = &H00000080&
  Height        = 315
  Left          = 1440
  TabIndex      = 108
  Top           = 1020
  Width         = 735
End
Begin SSPanel  pnlLengthUnit
  BevelInner    = 2 'Raised
  BevelOuter    = 0 'None
  Caption       = "m."
  Font3D        = 1 'Raised w/light shading
  ForeColor     = &H00000080&
  Height        = 315
  Index         = 0
  Left          = 2580
  TabIndex      = 112
  Top           = 720
  Width         = 315
End
Begin SSPanel  pnlLengthUnit
  BevelInner    = 2 'Raised
  BevelOuter    = 0 'None
  Caption       = "m."
  Font3D        = 1 'Raised w/light shading
  ForeColor     = &H00000080&
  Height        = 315
  Index         = 4
  Left          = 2580
  TabIndex      = 111
  Top           = 420
  Width         = 315
End
Begin TextBox  txtNodeDepth
  Alignment     = 2 'Center
  Height        = 315
  Left          = 2100
  TabIndex      = 115
  Text          = "0.30"
  Top           = 720
  Width         = 435
End
Begin TextBox  txtNodeWidth
  Alignment     = 2 'Center
  Height        = 315

```



FRMMAIN.FRM - 3

```

    Left           = 2100
    TabIndex      = 116
    Text          = "0.30"
    Top           = 420
    Width         = 435
End
Begin SSPanel Panel3D3
    BevelInner    = 2 'Raised
    BevelOuter    = 0 'None
    Caption       = "Y Dim"
    Font3D        = 1 'Raised w/light shading
    ForeColor     = &H00000080&
    Height        = 315
    Left          = 1440
    TabIndex      = 113
    Top           = 720
    Width         = 675
End
Begin SSPanel Panel3D2
    BevelInner    = 2 'Raised
    BevelOuter    = 0 'None
    Caption       = "X Dim"
    Font3D        = 1 'Raised w/light shading
    ForeColor     = &H00000080&
    Height        = 315
    Left          = 1440
    TabIndex      = 114
    Top           = 420
    Width         = 675
End
Begin TextBox txtGrid
    BorderStyle   = 0 'None
    Height        = 435
    Left          = 5040
    TabIndex      = 143
    Top           = 2280
    Visible       = 0 'False
    Width         = 1335
End
Begin SSFrame fmeEdgeYIndex
    Caption       = "Edge_Y"
    Font3D        = 1 'Raised w/light shading
    ForeColor     = &H00FF0000&
    Height        = 855
    Left          = 3480
    TabIndex      = 139
    Top           = 900
    Width         = 1035
Begin SSOption optEdgeY
    Caption       = "Interior"
    Font3D        = 3 'Inset w/light shading
    ForeColor     = &H00808000&
    Height        = 210
    Index         = 0
    Left          = 60
    TabIndex      = 142
    Top           = 240

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 4

```

        Value      = -1 'True
        Width      = 915
    End
    Begin SSOption optEdgeY
        Caption     = "Top"
        Font3D      = 3 'Inset w/light shading
        ForeColor   = &H00808000&
        Height      = 210
        Index       = 1
        Left        = 60
        TabIndex    = 141
        TabStop     = 0 'False
        Top         = 600
        Width       = 915
    End
    Begin SSOption optEdgeY
        Caption     = "Bottom"
        Font3D      = 3 'Inset w/light shading
        ForeColor   = &H00808000&
        Height      = 210
        Index       = 2
        Left        = 60
        TabIndex    = 140
        TabStop     = 0 'False
        Top         = 420
        Width       = 915
    End
End
Begin SSFrame fmeNode
    Caption       = "Position"
    Font3D        = 1 'Raised w/light shading
    ForeColor     = &H00FF0000&
    Height        = 1455
    Left          = 60
    TabIndex      = 117
    Top           = 300
    Width         = 3375
    Begin SSPanel pnlGen
        Caption    = "Arr X:Y"
        Font3D     = 1 'Raised w/light shading
        ForeColor  = &H00000080&
        Height     = 255
        Index      = 0
        Left       = 60
        ShadowColor = 1 'Black
        TabIndex   = 127
        Top        = 1140
        Width      = 675
    End
    Begin TextBox txtNode
        Alignment  = 2 'Center
        Height     = 315
        Index      = 4
        Left       = 780
        TabIndex   = 129
        Text       = "1"
        Top        = 1140
    End

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 5

```

        Width          = 315
    End
    Begin CommandButton cmdNodeOK
        BackColor      = &H00FF00FF&
        Caption        = "&OK"
        FontBold       = -1 'True
        FontItalic     = 0 'False
        FontName       = "MS Serif"
        FontSize       = 12
        FontStrikethru = 0 'False
        FontUnderline  = 0 'False
        Height         = 975
        Left           = 2820
        TabIndex      = 121
        Top            = 120
        Width          = 495
    End
    Begin SSPanel pnlNodeNo
        BevelInner     = 2 'Raised
        BevelOuter     = 0 'None
        Caption        = "No."
        Font3D         = 1 'Raised w/light shading
        ForeColor      = &H00000080&
        Height         = 315
        Left           = 60
        TabIndex      = 138
        Top            = 240
        Width          = 435
    End
    Begin SSPanel pnlNodeCo X
        BevelInner     = 2 'Raised
        BevelOuter     = 0 'None
        Caption        = "X"
        Font3D         = 1 'Raised w/light shading
        ForeColor      = &H00000080&
        Height         = 315
        Left           = 60
        TabIndex      = 137
        Top            = 540
        Width          = 435
    End
    Begin SSPanel pnlNodeCo Y
        BevelInner     = 2 'Raised
        BevelOuter     = 0 'None
        Caption        = "Y"
        Font3D         = 1 'Raised w/light shading
        ForeColor      = &H00004080&
        Height         = 315
        Left           = 60
        TabIndex      = 136
        Top            = 840
        Width          = 435
    End
    Begin SSPanel pnlLengthUnit
        BevelInner     = 2 'Raised
        BevelOuter     = 0 'None
        Caption        = "m."

```

FRMMAIN.FRM - 6

```

Font3D      = 1 'Raised w/light shading
ForeColor   = &H00000080&
Height     = 315
Index      = 6
Left       = 1020
TabIndex  = 131
Top       = 540
Width     = 315
End
Begin SSPanel pnlLengthUnit
BevelInner = 2 'Raised
BevelOuter = 0 'None
Caption    = "m."
Font3D    = 1 'Raised w/light shading
ForeColor  = &H00000080&
Height    = 315
Index     = 7
Left      = 1020
TabIndex  = 130
Top       = 840
Width     = 315
End
Begin TextBox txtNode
Alignment  = 2 'Center
Height    = 315
Index     = 5
Left      = 1320
TabIndex  = 128
Text      = "1"
Top       = 1140
Width     = 315
End
Begin SSPanel pnlComment
Caption    = "."
Font3D    = 1 'Raised w/light shading
ForeColor  = &H00FF0000&
Height    = 255
Index     = 0
Left      = 1140
ShadowColor = 1 'Black
TabIndex  = 126
Top       = 1200
Width     = 135
End
Begin SSPanel pnlComment
Caption    = "X:Y"
Font3D    = 1 'Raised w/light shading
ForeColor  = &H00000080&
Height    = 255
Index     = 1
Left      = 1680
ShadowColor = 1 'Black
TabIndex  = 125
Top       = 1140
Width     = 315
End
Begin TextBox txtNode

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 7

```

        Alignment      = 2 'Center
        Height         = 315
        Index          = 6
        Left           = 2100
        TabIndex       = 124
        Text           = "0"
        Top            = 1140
        Width          = 435
    End
    Begin SSPanel pnlComment
        Caption        = ":"
        Font3D         = 1 'Raised w/light shading
        ForeColor      = &H00FF0000&
        Height         = 255
        Index          = 2
        Left           = 2580
        ShadowColor    = 1 'Black
        TabIndex       = 123
        Top            = 1140
        Width          = 135
    End
    Begin TextBox txtNode
        Alignment      = 2 'Center
        Height         = 315
        Index          = 7
        Left           = 2760
        TabIndex       = 122
        Text           = "0"
        Top            = 1140
        Width          = 435
    End
    Begin TextBox txtNode
        Alignment      = 2 'Center
        Height         = 315
        Index          = 1
        Left           = 540
        TabIndex       = 120
        Top            = 240
        Width          = 435
    End
    Begin TextBox txtNode
        Alignment      = 2 'Center
        Height         = 315
        Index          = 2
        Left           = 540
        TabIndex       = 119
        Text           = "0"
        Top            = 540
        Width          = 435
    End
    Begin TextBox txtNode
        Alignment      = 2 'Center
        Height         = 315
        HideSelection  = 0 'False
        Index          = 3
        Left           = 540
        TabIndex       = 118

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 8

```

Text          = "0"
Top           = 840
Width        = 435
End
Begin SSPanel pnlNodeSupport
  BackColor   = &H00FFFF00&
  BevelInner  = 2 'Raised
  BevelOuter  = 0 'None
  Caption     = "Support Type"
  ForeColor   = &H00FF0000&
  Height      = 315
  Left        = 480
  TabIndex    = 132
  Top         = 600
  Visible     = 0 'False
  Width       = 1275
End
Begin SSOption optNodeSupport
  Caption     = "Pin"
  ForeColor   = &H00808000&
  Height      = 210
  Index       = 0
  Left        = 540
  TabIndex    = 135
  Top         = 300
  Value       = -1 'True
  Visible     = 0 'False
  Width       = 795
End
Begin SSOption optNodeSupport
  Caption     = "Fixed"
  ForeColor   = &H00808000&
  Height      = 210
  Index       = 1
  Left        = 600
  TabIndex    = 134
  TabStop     = 0 'False
  Top         = 300
  Visible     = 0 'False
  Width       = 735
End
Begin SSOption optNodeSupport
  Caption     = "Free"
  ForeColor   = &H00808000&
  Height      = 210
  Index       = 2
  Left        = 600
  TabIndex    = 133
  TabStop     = 0 'False
  Top         = 300
  Visible     = 0 'False
  Width       = 735
End
End
Begin Grid grdNode
  BackColor   = &H00FFFFFF&
  Cols        = 8

```

FRMMAIN.FRM - 9

```

    ForeColor      = &H000000FF&
    GridLineWidth  = 2
    Height         = 1575
    Left           = 4560
    Rows          = 100
    TabIndex       = 107
    Top            = 180
    Width         = 4335
End
Begin SSFrame Frame3D1
    Caption        = "Edge_X"
    Font3D         = 1 'Raised w/light shading
    ForeColor      = &H00FF0000&
    Height         = 795
    Left           = 3480
    TabIndex       = 103
    Top            = 120
    Width         = 1035
    Begin SSOption optEdgeX
        Caption    = "Interior"
        Font3D     = 3 'Inset w/light shading
        ForeColor  = &H00808000&
        Height     = 210
        Index      = 0
        Left       = 60
        TabIndex   = 106
        Top        = 240
        Value      = -1 'True
        Width      = 915
    End
    Begin SSOption optEdgeX
        Caption    = "Left"
        Font3D     = 3 'Inset w/light shading
        ForeColor  = &H00808000&
        Height     = 210
        Index      = 1
        Left       = 60
        TabIndex   = 105
        TabStop    = 0 'False
        Top        = 420
        Width      = 735
    End
    Begin SSOption optEdgeX
        Caption    = "Right"
        Font3D     = 3 'Inset w/light shading
        ForeColor  = &H00808000&
        Height     = 210
        Index      = 2
        Left       = 60
        TabIndex   = 104
        TabStop    = 0 'False
        Top        = 600
        Width      = 795
    End
End
Begin SSFrame fmeTextInputData

```

FRMMAIN.FRM - 10

```

Caption          = "Load Data"
Font3D          = 2 'Raised w/heavy shading
FontBold        = -1 'True
FontItalic      = 0 'False
FontName        = "MS Sans Serif"
FontSize        = 9.75
FontStrikethru  = 0 'False
FontUnderline   = 0 'False
ForeColor       = &H00FF0000&
Height          = 1815
Index           = 1
Left            = 9360
ShadowStyle     = 1 'Raised
TabIndex        = 70
Top             = 540
Visible         = 0 'False
Width           = 8835
Begin SSPanel  pnlLoadFactor
  BevelInner     = 2 'Raised
  BevelOuter     = 0 'None
  Caption        = "Dead Load Factor"
  Font3D         = 3 'Inset w/light shading
  ForeColor      = &H00FF0000&
  Height         = 315
  Index          = 1
  Left           = 5340
  TabIndex       = 157
  Top            = 1140
  Width          = 1935
End
Begin TextBox  txtLLFactor
  Height         = 315
  Left           = 7440
  TabIndex       = 152
  Text           = "1.7"
  Top            = 420
  Width          = 735
End
Begin SSPanel  pnlLoadFactor
  BevelInner     = 2 'Raised
  BevelOuter     = 0 'None
  Caption        = "Live Load Factor"
  Font3D         = 3 'Inset w/light shading
  ForeColor      = &H00FF0000&
  Height         = 315
  Index          = 0
  Left           = 5340
  TabIndex       = 154
  Top            = 420
  Width          = 1935
End
Begin SSPanel  pnlLoadUnit
  BevelInner     = 2 'Raised
  BevelOuter     = 0 'None
  Caption        = "Kg/m2"
  Font3D         = 1 'Raised w/light shading
  ForeColor      = &H00800000&

```

FRMMAIN.FRM - 11

```

Height      = 315
Index       = 5
Left        = 4200
TabIndex    = 150
Top         = 420
Width       = 675
End
Begin SSPanel pnlLL
  BevelInner = 2 'Raised
  BevelOuter = 0 'None
  Caption    = "Live Load"
  Font3D     = 3 'Inset w/light shading
  ForeColor  = &H00FF0000&
  Height     = 315
  Index      = 1
  Left       = 540
  TabIndex   = 149
  Top        = 420
  Width      = 1995
End
Begin TextBox txtLL
  Height     = 315
  Left       = 2820
  TabIndex   = 147
  Text       = "300"
  Top        = 420
  Width      = 1095
End
Begin SSPanel pnlLoadUnit
  BevelInner = 2 'Raised
  BevelOuter = 0 'None
  Caption    = "Kg/m2"
  Font3D     = 1 'Raised w/light shading
  ForeColor  = &H00800000&
  Height     = 315
  Index      = 4
  Left       = 4200
  TabIndex   = 71
  Top        = 1320
  Width      = 675
End
Begin TextBox txtSL
  Height     = 315
  Left       = 2820
  TabIndex   = 73
  Text       = "300"
  Top        = 1320
  Width      = 1095
End
Begin TextBox txtDLFactor
  Alignment  = 2 'Center
  Height     = 645
  Left       = 7440
  TabIndex   = 153
  Text       = "1.4"
  Top        = 960
  Width      = 735

```

FRMMAIN.FRM - 12

```

End
Begin SSPanel pnlLoadUnit
  BevelInner      = 2 'Raised
  BevelOuter      = 0 'None
  Caption         = "Kg/m2"
  Font3D          = 1 'Raised w/light shading
  ForeColor       = &H00800000&
  Height         = 315
  Index          = 6
  Left           = 4200
  TabIndex       = 151
  Top            = 960
  Width          = 675
End
Begin TextBox txtDL
  BackColor       = &H00000000&
  Enabled         = 0 'False
  ForeColor       = &H00FFFFFF&
  Height         = 315
  Left           = 2820
  TabIndex       = 146
  Text           = "0"
  Top            = 960
  Width          = 1095
End
Begin SSPanel Panel3D4
  BevelInner      = 2 'Raised
  BevelOuter      = 0 'None
  Caption         = "Super Impose Load"
  Font3D          = 3 'Inset w/light shading
  ForeColor       = &H00FF0000&
  Height         = 315
  Left           = 540
  TabIndex       = 72
  Top            = 1320
  Width          = 1995
End
Begin SSPanel pnlDL
  BevelInner      = 2 'Raised
  BevelOuter      = 0 'None
  Caption         = "Dead Load"
  Font3D          = 3 'Inset w/light shading
  ForeColor       = &H00FF0000&
  Height         = 315
  Index          = 1
  Left           = 540
  TabIndex       = 148
  Top            = 960
  Width          = 1995
End
Begin SSPanel Panel3D6 :
  BevelInner      = 2 'Raised
  BevelOuter      = 0 'None
  ForeColor       = &H00FF0000&
  Height         = 555
  Left           = 240
  TabIndex       = 156

```

FRMMAIN.FRM - 13

```

    Top          = 300
    Width        = 8355
End
Begin SSPanel Panel3D5
    BevelInner   = 2 'Raised
    BevelOuter   = 0 'None
    ForeColor    = &H00FF0000&
    Height       = 915
    Left         = 240
    TabIndex     = 155
    Top          = 840
    Width        = 8355
End
Begin SSFrame fmeLoadType
    Caption      = "Type"
    ForeColor    = &H00800080&
    Height       = 1095
    Left         = 7980
    TabIndex     = 97
    Top          = 1500
    Visible      = 0 'False
    Width        = 1275
    Begin SSOption optLoadType
        Caption   = "Point"
        ForeColor = &H00808000&
        Height    = 210
        Index     = 0
        Left      = 60
        TabIndex  = 101
        TabStop   = 0 'False
        Top       = 300
        Width     = 735
    End
    Begin SSOption optLoadType
        Caption   = "Uniform"
        ForeColor = &H00808000&
        Height    = 210
        Index     = 1
        Left      = 60
        TabIndex  = 100
        Top       = 480
        Value     = -1 'True
        Width     = 975
    End
    Begin SSOption optLoadType
        Caption   = "Trapizodal"
        ForeColor = &H00808000&
        Height    = 210
        Index     = 2
        Left      = 60
        TabIndex  = 99
        TabStop   = 0 'False
        Top       = 660
        Width     = 1155
    End
    Begin SSOption optLoadType
        Caption   = "Moment "

```

FRMMAIN.FRM - 14

```

    ForeColor      =   &H00808000&
    Height         =   210
    Index          =   3
    Left           =   60
    TabIndex       =   98
    TabStop        =   0   'False
    Top            =   840
    Width          =   1035
End
End
Begin SSFrame fmeLoadPosition
    Caption        =   "Position"
    ForeColor      =   &H00800080&
    Height         =   1695
    Left           =   8040
    TabIndex       =   84
    Top            =   1560
    Visible        =   0   'False
    Width          =   1635
    Begin SSPanel pnlLoadX1
        BackColor  =   &H00FFFF00&
        BevelInner =   2   'Raised
        BevelOuter =   0   'None
        Caption    =   "X1 :"
        ForeColor  =   &H00FF00FF&
        Height     =   315
        Left       =   60
        TabIndex   =   96
        Top        =   240
        Width      =   495
    End
    Begin SSPanel pnlLoadY2
        BackColor  =   &H00FFFF00&
        BevelInner =   2   'Raised
        BevelOuter =   0   'None
        Caption    =   "Y2 :"
        ForeColor  =   &H00FF00FF&
        Height     =   315
        Left       =   60
        TabIndex   =   95
        Top        =   1260
        Width      =   495
    End
    Begin SSPanel pnlLoadX2
        BackColor  =   &H00FFFF00&
        BevelInner =   2   'Raised
        BevelOuter =   0   'None
        Caption    =   "X2 :"
        ForeColor  =   &H00FF00FF&
        Height     =   315
        Left       =   60
        TabIndex   =   94
        Top        =   900
        Width      =   495
    End
    Begin SSPanel pnlLoadY1
        BackColor  =   &H00FFFF00&

```

FRMMAIN.FRM - 15

```

    BevelInner      = 2 'Raised
    BevelOuter     = 0 'None
    Caption        = "Y1 :"
    ForeColor      = &H00FF00FF&
    Height         = 315
    Left           = 60
    TabIndex       = 93
    Top            = 540
    Width          = 495
End
Begin TextBox txtLoadX1
    Height         = 315
    Left           = 540
    TabIndex       = 92
    Top            = 240
    Width          = 675
End
Begin TextBox txtLoadY1
    Height         = 315
    Left           = 540
    TabIndex       = 91
    Top            = 600
    Width          = 675
End
Begin TextBox txtLoadX2
    Height         = 315
    Left           = 540
    TabIndex       = 90
    Top            = 960
    Width          = 675
End
Begin TextBox txtLoadY2
    Height         = 315
    Left           = 540
    TabIndex       = 89
    Top            = 1320
    Width          = 675
End
Begin SSPanel pnlLengthUnit
    BackColor      = &H00FFFF00&
    BevelInner     = 2 'Raised
    BevelOuter     = 0 'None
    Caption        = "m."
    ForeColor      = &H000000FF&
    Height         = 315
    Index          = 1
    Left           = 1260
    TabIndex       = 88
    Top            = 240
    Width          = 315
End
Begin SSPanel pnlLengthUnit
    BackColor      = &H00FFFF00&
    BevelInner     = 2 'Raised
    BevelOuter     = 0 'None
    Caption        = "m."
    ForeColor      = &H000000FF&

```

FRMMAIN.FRM - 16

```

        Height      = 315
        Index       = 2
        Left        = 1260
        TabIndex    = 87
        Top         = 600
        Width       = 315
    End
    Begin SSPanel pnlLengthUnit
        BackColor   = &H00FFFF00&
        BevelInner  = 2 'Raised
        BevelOuter  = 0 'None
        Caption     = "m."
        ForeColor   = &H000000FF&
        Height      = 315
        Index       = 3
        Left        = 1260
        TabIndex    = 86
        Top         = 960
        Width       = 315
    End
    Begin SSPanel pnlSlabFc_U
        BackColor   = &H00FFFF00&
        BevelInner  = 2 'Raised
        BevelOuter  = 0 'None
        Caption     = "m."
        ForeColor   = &H000000FF&
        Height      = 315
        Index       = 4
        Left        = 1260
        TabIndex    = 85
        Top         = 1320
        Width       = 315
    End
End
Begin SSFrame fmeLoadMagnitude
    Caption       = "Magnitude"
    ForeColor     = &H00008000&
    Height        = 1455
    Left          = 7080
    TabIndex      = 77
    Top           = 1620
    Visible       = 0 'False
    Width         = 2055
    Begin TextBox txtLoadEnd
        Height     = 315
        Left       = 1980
        TabIndex   = 82
        Top        = -60
        Visible    = 0 'False
        Width      = 735
    End
    Begin SSPanel pnlLoadw1
        BackColor   = &H00FFFF00&
        BevelInner  = 2 'Raised
        BevelOuter  = 0 'None
        Caption     = "w1"
        ForeColor   = &H00FF00FF&

```

FRMMAIN.FRM - 17

```

        Height      = 315
        Left        = 2040
        TabIndex    = 81
        Top         = 0
        Visible     = 0 'False
        Width       = 375
    End
    Begin SSPanel pnlw2
        BackColor   = &H00FFFF00&
        BevelInner  = 2 'Raised
        BevelOuter  = 0 'None
        Caption     = "w2"
        ForeColor   = &H00FF00FF&
        Height      = 315
        Left        = -60
        TabIndex    = 80
        Top         = 1320
        Visible     = 0 'False
        Width       = 375
    End
    Begin SSPanel pnlLoadUnit
        BackColor   = &H00FFFF00&
        BevelInner  = 2 'Raised
        BevelOuter  = 0 'None
        Caption     = "Kg"
        ForeColor   = &H00FF00FF&
        Height      = 315
        Index       = 1
        Left        = 1980
        TabIndex    = 78
        Top         = -120
        Visible     = 0 'False
        Width       = 435
    End
    Begin TextBox txtLoadStart
        Height      = 315
        Left        = 1920
        TabIndex    = 83
        Top         = -120
        Visible     = 0 'False
        Width       = 735
    End
    Begin SSPanel pnlLoadUnit
        BackColor   = &H00FFFF00&
        BevelInner  = 2 'Raised
        BevelOuter  = 0 'None
        Caption     = "Kg"
        ForeColor   = &H00FF00FF&
        Height      = 315
        Index       = 0
        Left        = -300
        TabIndex    = 79
        Top         = 1260
        Visible     = 0 'False
        Width       = 435
    End
End
End

```

FRMMAIN.FRM - 18

```

Begin Grid grdLoad
  Cols          = 8
  ForeColor     = &H00FF0000&
  Height        = 1575
  Left          = 8760
  Rows          = 100
  TabIndex      = 76
  Top           = 1740
  Visible       = 0 'False
  Width         = 3975
End
Begin SSPanel pnlLoadNo
  BackColor     = &H00FFFF00&
  BevelInner    = 2 'Raised
  Caption       = "No."
  ForeColor     = &H00FF00FF&
  Height        = 375
  Left          = 8520
  TabIndex      = 75
  Top           = 1500
  Visible       = 0 'False
  Width         = 555
End
Begin TextBox txtLoadNo
  Height        = 315
  Left          = 8340
  TabIndex      = 74
  Top           = 1380
  Visible       = 0 'False
  Width         = 675
End
End
Begin SSFrame fmeTextInputData
  Caption       = "Element Data"
  Font3D        = 2 'Raised w/heavy shading
  FontBold      = -1 'True
  FontItalic    = 0 'False
  FontName      = "MS Sans Serif"
  FontSize      = 9.75
  FontStrikethru = 0 'False
  FontUnderline = 0 'False
  ForeColor     = &H00808000&
  Height        = 1815
  Index         = 2
  Left          = 9240
  ShadowStyle   = 1 'Raised
  TabIndex      = 68
  Top           = 540
  Visible       = 0 'False
  Width         = 8835
End
Begin Grid grdElement
  Cols          = 9
  ForeColor     = &H000000FF&
  GridLineWidth = 2
  Height        = 1455
  Left          = 60
  Rows          = 5000

```

FRMMAIN.FRM - 19

```

        TabIndex      = 69
        Top           = 300
        Width        = 8715
    End
End
Begin SSFrame fmeTextInputData
    Caption          = "Information Data"
    Font3D           = 2 'Raised w/heavy shading
    FontBold         = -1 'True
    FontItalic       = 0 'False
    FontName         = "MS Sans Serif"
    FontSize         = 9.75
    FontStrikethru   = 0 'False
    FontUnderline    = 0 'False
    ForeColor        = &H00FF0000&
    Height           = 1815
    Index            = 4
    Left             = 9000
    ShadowStyle      = 1 'Raised
    TabIndex         = 53
    Top              = 480
    Visible          = 0 'False
    Width            = 8955
    Begin SSPanel pnlFileName
        Alignment     = 1 'Left Justify - MIDDLE
        BevelInner    = 2 'Raised
        BevelOuter    = 0 'None
        Caption       = "FileName"
        Font3D        = 1 'Raised w/light shading
        ForeColor     = &H00808000&
        Height        = 375
        Left          = 60
        TabIndex      = 67
        Top           = 300
        Width         = 1275
    End
    Begin TextBox txtFileName
        BackColor     = &H00E0FFFF&
        ForeColor     = &H00008000&
        Height        = 315
        HideSelection = 0 'False
        Left          = 1440
        TabIndex      = 66
        Top           = 360
        Width         = 1875
    End
    Begin SSPanel pnlProjectName
        Alignment     = 1 'Left Justify - MIDDLE
        BevelInner    = 2 'Raised
        BevelOuter    = 0: 'None
        Caption       = "Project Name"
        Font3D        = 1 'Raised w/light shading
        ForeColor     = &H00808000&
        Height        = 375
        Left          = 60
        TabIndex      = 65
        Top           = 660
    End

```

FRMMAIN.FRM - 20

```

Width          = 1275
End
Begin TextBox txtProjectName
  BackColor    = &H00E0FFFF&
  ForeColor    = &H00008000&
  Height       = 315
  HideSelection = 0 'False
  Left         = 1440
  TabIndex     = 64
  Top         = 720
  Width       = 1875
End
Begin SSPanel pnlTime
  Alignment    = 1 'Left Justify - MIDDLE
  BevelInner   = 2 'Raised
  BevelOuter   = 0 'None
  Caption      = "Time"
  Font3D       = 1 'Raised w/light shading
  ForeColor    = &H00808000&
  Height       = 375
  Left         = 60
  TabIndex     = 63
  Top         = 1020
  Width       = 1275
End
Begin TextBox txtTime
  BackColor    = &H00E0FFFF&
  ForeColor    = &H00008000&
  Height       = 315
  HideSelection = 0 'False
  Left         = 1440
  TabIndex     = 62
  Top         = 1080
  Width       = 1875
End
Begin SSPanel pnlDate
  Alignment    = 1 'Left Justify - MIDDLE
  BevelInner   = 2 'Raised
  BevelOuter   = 0 'None
  Caption      = "Date"
  Font3D       = 1 'Raised w/light shading
  ForeColor    = &H00808000&
  Height       = 375
  Left         = 60
  TabIndex     = 61
  Top         = 1440
  Width       = 1275
End
Begin TextBox txtDate
  BackColor    = &H00E0FFFF&
  ForeColor    = &H00008000&
  Height       = 315
  HideSelection = 0 'False
  Left         = 1440
  TabIndex     = 60
  Top         = 1440
  Width       = 1875

```

FRMMAIN.FRM - 21

```

End
Begin SSFrame fmeInfomation
  Caption      = "&Information"
  Font3D       = 1 'Raised w/light shading
  ForeColor    = &H00808000&
  Height       = 1575
  Left         = 7080
  TabIndex     = 58
  Top          = 180
  Width        = 1695
  Begin TextBox txtInformation
    BackColor   = &H00E0FFFF&
    ForeColor    = &H00008000&
    Height       = 1275
    HideSelection = 0 'False
    Left         = 60
    MultiLine    = -1 'True
    TabIndex     = 59
    Top          = 240
    Width        = 1515
  End
End
Begin Timer tmrTime
  Interval     = 1000
  Left         = 7920
  Top          = 600
End
Begin SSPanel pnlAddress
  Alignment    = 1 'Left Justify - MIDDLE
  BevelInner   = 2 'Raised
  BevelOuter   = 0 'None
  Caption      = "Address"
  Font3D       = 1 'Raised w/light shading
  ForeColor    = &H00808000&
  Height       = 855
  Left         = 3600
  TabIndex     = 57
  Top          = 300
  Width        = 975
End
Begin TextBox txtAddress
  BackColor    = &H00E0FFFF&
  ForeColor    = &H00008000&
  Height       = 855
  HideSelection = 0 'False
  Left         = 4740
  MultiLine    = -1 'True
  TabIndex     = 56
  Top          = 360
  Width        = 1875
End
Begin SSPanel pnlEngineer
  Alignment    = 1 'Left Justify - MIDDLE
  BevelInner   = 2 'Raised
  BevelOuter   = 0 'None
  Caption      = "Engineer"
  Font3D       = 1 'Raised w/light shading

```

FRMMAIN.FRM - 22

```

        ForeColor      =   &H00808000&
        Height         =   375
        Left           =   3600
        TabIndex       =   55
        Top            =   1200
        Width          =   975
    End
    Begin TextBox txtEngineer
        BackColor      =   &H00E0FFFF&
        ForeColor      =   &H00008000&
        Height         =   315
        HideSelection  =   0   'False
        Left           =   4740
        TabIndex       =   54
        Top            =   1260
        Width          =   1875
    End
End
Begin SSPanel pnlSelectInput
    BevelWidth        =   0
    BorderWidth       =   1
    ForeColor         =   &H000000FF&
    Height            =   1815
    Left              =   120
    RoundedCorners    =   0   'False
    TabIndex          =   52
    Top               =   120
    Width             =   435
    Begin SSRibbon gphSelectInput
        AutoSize       =   0   'None
        Height         =   375
        Index          =   0
        Left           =   60
        PictureDisabled = (none)
        PictureDn      = (Bitmap)
        PictureUp      = (Bitmap)
        Top            =   0
        Width          =   375
    End
    Begin SSRibbon gphSelectInput
        AutoSize       =   0   'None
        Height         =   375
        Index          =   1
        Left           =   60
        PictureDisabled = (none)
        PictureDn      = (Bitmap)
        PictureUp      = (Bitmap)
        Top            =   720
        Width          =   375
    End
    Begin SSRibbon gphSelectInput
        AutoSize       =   0   'None
        Height         =   375
        Index          =   2
        Left           =   60
        PictureDisabled = (none)
        PictureDn      = (Bitmap)
    End

```

FRMMAIN.FRM - 23

```

    PictureUp      = (Bitmap)
    Top            = 360
    Width         = 375
End
Begin SSRibbon gphSelectInput
    AutoSize      = 0 'None
    Height       = 375
    Index        = 3
    Left         = 60
    PictureDisabled = (none)
    PictureDn    = (Bitmap)
    PictureUp    = (Bitmap)
    Top         = 1080
    Width       = 375
End
Begin SSRibbon gphSelectInput
    AutoSize      = 0 'None
    Height       = 375
    Index        = 4
    Left         = 60
    PictureDisabled = (none)
    PictureDn    = (Bitmap)
    PictureUp    = (Bitmap)
    Top         = 1440
    Width       = 375
End
End
Begin SSFrame fmeTextInputData
    Caption      = "General Data"
    Font3D      = 2 'Raised w/heavy shading
    FontBold    = -1 'True
    FontItalic  = 0 'False
    FontName    = "MS Sans Serif"
    FontSize    = 9.75
    FontStrikethru = 0 'False
    FontUnderline = 0 'False
    ForeColor   = &H00FF00FF&
    Height     = 1815
    Index     = 3
    Left      = 9120
    ShadowStyle = 1 'Raised
    TabIndex  = 13
    Top      = 180
    Visible  = 0 'False
    Width   = 8955
Begin SSFrame fmeStrand
    Caption      = "Strand Property"
    Font3D      = 1 'Raised w/light shading
    ForeColor   = &H00004000&
    Height     = 735
    Left      = 1440
    TabIndex  = 14
    Top      = 300
    Width   = 2415
Begin ComboBox cmbSFs_
    BackColor   = &H00FF0000&
    ForeColor   = &H00FFFFFF&

```

FRMMAIN.FRM - 24

```

    Height      = 315
    Left        = 540
    TabIndex    = 17
    Text        = "19000"
    Top         = 300
    Width       = 1095
End
Begin SPanel pnlFyStrand
    Alignment   = 1 'Left Justify - MIDDLE
    BevelInner  = 2 'Raised
    BevelOuter  = 0 'None
    Caption     = "p"
    Font3D      = 1 'Raised w/light shading
    ForeColor   = &H00808000&
    Height      = 375
    Left        = 60
    TabIndex    = 16
    Top         = 240
    Width       = 435
End
Begin SPanel pnlFs U
    Alignment   = 1 'Left Justify - MIDDLE
    BevelInner  = 2 'Raised
    BevelOuter  = 0 'None
    Caption     = "Kg/No."
    Font3D      = 1 'Raised w/light shading
    ForeColor   = &H00FF0000&
    Height      = 375
    Index      = 1
    Left        = 1680
    TabIndex    = 15
    Top         = 240
    Width       = 675
End
End
Begin SFrame fmeSlabSystem
    Caption     = "Slab System"
    Font3D      = 1 'Raised w/light shading
    ForeColor   = &H00004000&
    Height      = 735
    Left        = 60
    TabIndex    = 49
    Top         = 300
    Width       = 1335
Begin SOption optSystem
    Caption     = "Flat Plate"
    Font3D      = 3 'Inset w/light shading
    ForeColor   = &H00008000&
    Height      = 210
    Index       = 0
    Left        = 60
    TabIndex    = 51
    Top         = 480
    Value       = -1 'True
    Width       = 1095
End
Begin SOption optSystem

```

FRMMAIN.FRM - 25

```

Caption          = "Flat Slab"
Font3D           = 3 'Inset w/light shading
ForeColor        = &H00008000&
Height          = 210
Index           = 1
Left            = 60
TabIndex        = 50
TabStop         = 0 'False
Top             = 240
Width           = 1095
End
End
Begin SSFrame fmeDropOrCapital
Caption          = "Column Head"
Font3D           = 1 'Raised w/light shading
ForeColor        = &H00004000&
Height          = 735
Left            = 60
TabIndex        = 46
Top             = 1020
Width           = 1335
Begin SSCheck chkCHHead
Caption          = "Capital"
Enabled          = 0 'False
Font3D           = 3 'Inset w/light shading
ForeColor        = &H00008000&
Height          = 195
Index           = 1
Left            = 60
TabIndex        = 48
Top             = 480
Width           = 1095
End
Begin SSCheck chkCHHead
Caption          = "Drop panel"
Enabled          = 0 'False
Font3D           = 3 'Inset w/light shading
ForeColor        = &H00008000&
Height          = 240
Index           = 0
Left            = 60
TabIndex        = 47
Top             = 240
Width           = 1215
End
End
Begin SSPanel pnlThick_U
BevelInner      = 2 'Raised
BevelOuter      = 0 'None
Caption         = "cm."
Font3D           = 1 'Raised w/light shading
ForeColor        = &H00FF0000&
Height          = 375
Index           = 1
Left            = 8460
TabIndex        = 45
Top             = 960

```

FRMMAIN.FRM - 26

```

Width           = 435
End
Begin TextBox txtThick
  BackColor     = &H00FFFFFF&
  Enabled       = 0 'False
  ForeColor     = &H00FF0000&
  Height        = 315
  Left          = 7980
  TabIndex      = 44
  Top           = 960
  Width         = 435
End
Begin SSPanel pnlThick
  Alignment     = 1 'Left Justify - MIDDLE
  BevelInner    = 2 'Raised
  BevelOuter    = 0 'None
  Caption       = "Slab Thk."
  Font3D        = 1 'Raised w/light shading
  ForeColor     = &H00808000&
  Height        = 375
  Left          = 7020
  TabIndex      = 43
  Top           = 960
  Width         = 975
End
Begin SSFrame fmeReinforce
  Caption       = "Steel Property"
  Font3D        = 1 'Raised w/light shading
  ForeColor     = &H00400000&
  Height        = 735
  Left          = 1440
  TabIndex      = 39
  Top           = 1020
  Width         = 2415
Begin SSPanel pnlFs_U
  Alignment     = 1 'Left Justify - MIDDLE
  BevelInner    = 2 'Raised
  BevelOuter    = 0 'None
  Caption       = "Ksc"
  Font3D        = 1 'Raised w/light shading
  ForeColor     = &H00FF0000&
  Height        = 375
  Index         = 0
  Left          = 1740
  TabIndex      = 42
  Top           = 240
  Width         = 495
End
Begin SSPanel pnlFs_
  Alignment     = 1 'Left Justify - MIDDLE
  BevelInner    = 2 'Raised
  BevelOuter    = 0 'None
  Caption       = "Fy"
  Font3D        = 1 'Raised w/light shading
  ForeColor     = &H00808000&
  Height        = 375
  Left          = 60

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 27

```

    TabIndex      = 41
    Top           = 240
    Width        = 435
End
Begin ComboBox cmbFs_
    BackColor     = &H00FF0000&
    ForeColor     = &H00FFFFFF&
    Height        = 315
    Left         = 540
    TabIndex      = 40
    Text          = "2400"
    Top          = 300
    Width        = 1095

```

End

End

Begin SSFrame fmeSlab

```

Caption         = "Concrete Properties"
Font3D          = 1 'Raised w/light shading
ForeColor       = &H00004000&
Height         = 1215
Left           = 3900
TabIndex       = 29
Top            = 180
Width          = 3075

```

Begin SSPanel pnlMP

```

Alignment      = 1 'Left Justify - MIDDLE
BevelOuter     = 0 'None
Caption        = "Slab Column"
Font3D         = 3 'Inset w/light shading
FontBold       = -1 'True
FontItalic     = -1 'True
FontName       = "MS Sans Serif"
FontSize       = 8.25
FontStrikethru = 0 'False
FontUnderline  = 0 'False
ForeColor      = &H00008000&
Height        = 255
Left          = 360
TabIndex      = 38
Top           = 225
Width        = 1875

```

End

Begin SSPanel pnlSlabEcU

```

Alignment      = 1 'Left Justify - MIDDLE
BevelInner     = 2 'Raised
BevelOuter     = 0 'None
Caption        = "Ksc"
Font3D         = 1 'Raised w/light shading
ForeColor      = &H00FF0000&
Height        = 390
Left          = 2580
TabIndex      = 37
Top           = 780
Width        = 435

```

End

Begin SSPanel pnlSlabFc_U

```

Alignment      = 1 'Left Justify - MIDDLE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 28

```

BevelInner      = 2 'Raised
BevelOuter     = 0 'None
Caption        = "Ksc"
Font3D         = 1 'Raised w/light shading
ForeColor      = &H00FF0000&
Height         = 375
Index          = 0
Left           = 2580
TabIndex       = 36
Top            = 420
Width          = 435
End
Begin SSPanel  pnlSlabModulus
  Alignment     = 1 'Left Justify - MIDDLE
  BevelInner    = 2 'Raised
  BevelOuter    = 0 'None
  Caption       = "Ec"
  Font3D        = 1 'Raised w/light shading
  ForeColor     = &H00808000&
  Height        = 375
  Left          = 60
  TabIndex      = 35
  Top           = 780
  Width         = 375
End
Begin SSPanel  pnlSlabFc_
  Alignment     = 1 'Left Justify - MIDDLE
  BevelInner    = 2 'Raised
  BevelOuter    = 0 'None
  Caption       = "f'c"
  Font3D        = 1 'Raised w/light shading
  ForeColor     = &H00808000&
  Height        = 375
  Left          = 60
  TabIndex      = 34
  Top           = 420
  Width         = 375
End
Begin ComboBox cmbSlabFc_
  BackColor     = &H00FF0000&
  ForeColor     = &H00FFFFFF&
  Height        = 315
  Left          = 420
  TabIndex      = 33
  Text          = "375"
  Top           = 480
  Width         = 1035
End
Begin ComboBox cmbColumnFc_
  BackColor     = &H00FF0000&
  ForeColor     = &H00FFFFFF&
  Height        = 315
  Left          = 1560
  TabIndex      = 32
  Text          = "375"
  Top           = 480
  Width         = 1035

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 29

```

End
Begin ComboBox cmbSlabEc
  BackColor      =  &H00FF0000&
  ForeColor      =  &H00FFFFFF&
  Height         =  315
  Left           =  420
  TabIndex       =  31
  Text           =  "375"
  Top            =  840
  Width          =  1035
End
Begin ComboBox cmbColumnEc
  BackColor      =  &H00FF0000&
  ForeColor      =  &H00FFFFFF&
  Height         =  315
  Left           =  1560
  TabIndex       =  30
  Text           =  "375"
  Top            =  840
  Width          =  1035
End
End
Begin SSFrame fmeLengthUnit
  Caption        =  "Length Unit"
  Enabled        =  0 'False
  ForeColor      =  &H00800080&
  Height         =  735
  Left           =  1440
  TabIndex       =  26
  Top            =  240
  Visible        =  0 'False
  Width          =  1275
  Begin SSOption optLengthUnit
    Caption      =  "Meter"
    ForeColor    =  &H00008000&
    Height       =  210
    Index        =  4
    Left         =  60
    TabIndex     =  28
    Top          =  240
    Value        =  -1 'True
    Width        =  1155
  End
  Begin SSOption optLengthUnit
    Caption      =  "Centimeter"
    ForeColor    =  &H00008000&
    Height       =  210
    Index        =  5
    Left         =  60
    TabIndex     =  27
    TabStop      =  0 'False
    Top          =  480
    Width        =  1155
  End
End
End
Begin SSFrame fmeLoadUnit
  Caption        =  "Load Unit"

```

FRMMAIN.FRM - 30

```

Enabled          = 0 'False
ForeColor        = &H00800080&
Height          = 735
Left            = 2760
TabIndex        = 23
Top            = 240
Visible         = 0 'False
Width          = 1095
Begin SSOption optLoadUnit
  Caption        = "Kilogram"
  ForeColor     = &H00008000&
  Height        = 210
  Index         = 4
  Left          = 60
  TabIndex      = 25
  Top           = 240
  Value         = -1 'True
  Width         = 975
End
Begin SSOption optLoadUnit
  Caption        = "Newton"
  ForeColor     = &H00008000&
  Height        = 210
  Index         = 5
  Left          = 60
  TabIndex      = 24
  TabStop       = 0 'False
  Top           = 480
  Width         = 975
End
End
Begin SSFrame fmeStrandSystem
  Caption        = "Strand System"
  Font3D        = 1 'Raised w/light shading
  ForeColor     = &H00400000&
  Height        = 675
  Left          = 7020
  TabIndex      = 20
  Top           = 240
  Width         = 1815
Begin SSOption optBond
  Caption        = "Unbond"
  Font3D        = 3 'Inset w/light shading
  ForeColor     = &H00008000&
  Height        = 210
  Index         = 2
  Left          = 60
  TabIndex      = 22
  TabStop       = 0 'False
  Top           = 420
  Width         = 1155
End
Begin SSOption optBond
  Caption        = "Bond"
  Font3D        = 3 'Inset w/light shading
  ForeColor     = &H00008000&
  Height        = 210

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 31

```

        Index          = 3
        Left           = 60
        TabIndex       = 21
        Top            = 240
        Value          = -1 'True
        Width          = 1155
    End
End
Begin SSPanel Panel3D1
    Alignment         = 1 'Left Justify - MIDDLE
    BevelInner        = 2 'Raised
    BevelOuter        = 0 'None
    Caption           = "Strand Type"
    Font3D            = 1 'Raised w/light shading
    ForeColor         = &H00808000&
    Height            = 375
    Left              = 3900
    TabIndex          = 19
    Top               = 1380
    Width             = 1155
End
Begin ComboBox cmbStrand
    BackColor         = &H00FF0000&
    ForeColor         = &H00FFFFFF&
    Height            = 315
    Left              = 5100
    TabIndex          = 18
    Top               = 1440
    Width             = 3735
End
End
Begin SSPanel pnlWork
    Align             = 2 'Align Bottom
    Alignment         = 0 'Left Justify - TOP
    ForeColor         = &H000000FF&
    Height            = 435
    Left              = 0
    Outline           = -1 'True
    TabIndex          = 9
    Top               = 1740
    Visible           = 0 'False
    Width             = 9600
Begin TextBox txtDummy
    Height            = 315
    Left              = 3300
    TabIndex          = 11
    Text              = "txtDummy"
    Top               = 420
    Visible           = 0 'False
    Width             = 210
End
Begin CommonDialog cmdlgSystem
    Color             = &H00C0C0C0&
    DefaultExt        = ".PSD"
    DialogTitle       = "Open"
    Filename           = "Untitled"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 32

```

    Filter      = "Prestress data files (*.psd)|*.psd All files
    Left        = 7200
    Top         = 0
End
Begin Image imgWorkTool1
    Height      = 330
    Left        = 60
    Picture     = (Bitmap)
    Top         = 45
    Width       = 360
End
Begin Image imgWorkTool2
    Height      = 330
    Left        = 420
    Picture     = (Bitmap)
    Top         = 60
    Width       = 360
End
Begin Image imgWorkTool3
    Height      = 330
    Left        = 780
    Picture     = (Bitmap)
    Top         = 45
    Width       = 360
End
Begin Image imgWorkTool4
    Height      = 330
    Left        = 1140
    Picture     = (Bitmap)
    Top         = 45
    Width       = 360
End
Begin Image imgWorkTool5
    Height      = 330
    Left        = 1860
    Picture     = (Bitmap)
    Top         = 45
    Width       = 360
End
Begin Image imgWorkTool6
    Height      = 330
    Left        = 1500
    Picture     = (Bitmap)
    Top         = 45
    Width       = 360
End
End
Begin SSPanel pnlStatus
    Align       = 2 'Align Bottom
    Alignment   = 1 'Left Justify - MIDDLE
    Caption     = " Ready...."
    Font3D      = 3 'Inset w/light shading
    FontBold    = -1 'True
    FontItalic  = 0 'False
    FontName    = "MS Sans Serif"
    FontSize    = 9.75
    FontStrikethru = 0 'False

```

FRMMAIN.FRM - 33

```

FontUnderline = 0 'False
ForeColor = &H00FF0000&
Height = 480
Left = 0
RoundedCorners = 0 'False
TabIndex = 1
Top = 2175
Width = 9600
Begin SSCheck chkTextInputData
  Caption = "DataTool"
  ForeColor = &H00008000&
  Height = 210
  Left = 8400
  TabIndex = 8
  Top = 60
  Width = 1155
End
Begin SSCheck chkInput
  Caption = "WorkTool"
  Enabled = 0 'False
  ForeColor = &H00008000&
  Height = 270
  Left = 6960
  TabIndex = 7
  Top = 240
  Visible = 0 'False
  Width = 1215
End
Begin SSCheck chkSystem
  Caption = "SystemTool"
  ForeColor = &H00008000&
  Height = 240
  Left = 6960
  TabIndex = 6
  Top = 60
  Width = 1335
End
Begin MhState kstInsert
  Autosize = -1 'True
  Height = 420
  Left = 8760
  Style = 2 'Insert State
  TabIndex = 5
  TimerInterval = 100
  Top = 30
  Value = 0 'False
  Visible = 0 'False
  Width = 420
End
Begin MhState kstNumLock
  Autosize = -1 'True
  Height = 420
  Left = 8340
  Style = 1 'Num Lock
  TabIndex = 4
  TimerInterval = 100
  Top = 30

```

FRMMAIN.FRM - 34

```

Value           = 0 'False
Visible         = 0 'False
Width          = 420
End
Begin MhState kstCapLock
Autosize       = -1 'True
Height         = 420
Left           = 7920
TabIndex       = 3
TimerInterval  = 100
Top            = 30
Value         = 0 'False
Visible       = 0 'False
Width        = 420
End
Begin MhState kstScrollLock
Autosize       = -1 'True
Height         = 420
Left           = 9180
Style          = 3 'Scroll Lock
TabIndex       = 2
TimerInterval  = 100
Top            = 60
Value         = 0 'False
Visible       = 0 'False
Width        = 420
End
Begin SSPanel pnlMenu
Align          = 1 'Align Top
Alignment      = 4 'Right Justify - MIDDLE
ForeColor      = &H00FF0000&
Height         = 435
Left           = 0
Outline        = -1 'True
RoundedCorners = 0 'False
TabIndex       = 0
Top            = 0
Width          = 9600
Begin SSCommand cmdPrevious
BevelWidth     = 1
Caption        = "<< Previous"
Font3D         = 2 'Raised w/heavy shading
ForeColor      = &H00FF0000&
Height         = 315
Left           = 3480
Picture        = (none)
TabIndex       = 145
Top            = 60
Visible        = 0 'False
Width          = 1095
End
Begin SSCommand cmdNext
BevelWidth     = 1
Caption        = "Next >>"
Font3D         = 2 'Raised w/heavy shading
ForeColor      = &H00FF0000&

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 35

```

    Height      = 315
    Left        = 4620
    Picture     = (none)
    TabIndex    = 144
    Top         = 60
    Visible     = 0 'False
    Width       = 1095
End
Begin SSPanel pnlT
    BevelOuter  = 0 'None
    Font3D      = 2 'Raised w/heavy shading
    FontBold    = -1 'True
    FontItalic  = 0 'False
    FontName    = "Arial"
    FontSize    = 9.75
    FontStrikethru = 0 'False
    FontUnderline = 0 'False
    ForeColor   = &H00FF0000&
    Height      = 315
    Left        = 5880
    TabIndex    = 10
    Top         = 60
    Width       = 3615
End
Begin Image imgTool1
    Height      = 330
    Index       = 6
    Left        = 2340
    Picture     = (Bitmap)
    Top         = 45
    Width       = 360
End
Begin Image imgTool1
    Height      = 330
    Index       = 5
    Left        = 1260
    Picture     = (Bitmap)
    Top         = 45
    Width       = 360
End
Begin Image imgTool1
    Height      = 330
    Index       = 7
    Left        = 2700
    Picture     = (Bitmap)
    Top         = 45
    Width       = 360
End
Begin Image imgTool1
    Height      = 330
    Index       = 4
    Left        = 1980
    Picture     = (Bitmap)
    Top         = 45
    Width       = 360
End
Begin Image imgTool1

```

FRMMAIN.FRM - 36

```

    Height      = 330
    Index       = 1
    Left        = 420
    Picture     = (Bitmap)
    Top         = 45
    Width       = 360
End
Begin Image imgTool1
    Height      = 330
    Index       = 3
    Left        = 1620
    Picture     = (Bitmap)
    Top         = 45
    Width       = 360
End
Begin Image imgTool1
    Height      = 330
    Index       = 2
    Left        = 780
    Picture     = (Bitmap)
    Top         = 45
    Width       = 360
End
Begin Image imgTool1
    Height      = 330
    Index       = 0
    Left        = 60
    Picture     = (Bitmap)
    Top         = 45
    Width       = 360
End
End
Begin Menu mnuFile
    Caption     = "&File"
    Begin Menu mnuFile_New
        Caption = "&New"
        Shortcut = {F1}
    End
    Begin Menu mnuFile_Open
        Caption = "&Open..."
        Shortcut = {F3}
    End
    Begin Menu mnuFile_Save
        Caption = "&Save"
        Enabled = 0 'False
        Shortcut = {F2}
    End
    Begin Menu mnuFile_SaveAs
        Caption = "Save &As..."
    End
    Begin Menu mnuFile__1
        Caption = "- "
    End
    Begin Menu mnuFile_Print
        Caption = "&Print..."
        Begin Menu mnuFile_Print_Prestress
            Caption = "Prestress"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 37

```

Begin Menu mnuFile_Print_Prestress_Profile
  Caption      = "&Profile Tendon"
  Enabled      = 0 'False
  Shortcut     = ^P
End
Begin Menu mnuFile_Print_
  Caption      = "&Layout Tendon"
  Enabled      = 0 'False
  Shortcut     = ^L
  Visible     = 0 'False
End
Begin Menu mnuFile_Print_Prestress_LProfile
  Caption      = "List Pro&file"
  Enabled      = 0 'False
  Shortcut     = ^F
End
Begin Menu mnuFile_Print_Prestress_LCompute
  Caption      = "List &Computation "
  Enabled      = 0 'False
  Shortcut     = ^C
End
End
Begin Menu mnuFile_Print_Flat
  Caption      = "&Flat Slab && Flat Plate"
  Begin Menu mnuFile_Print_Flat_LCompute
    Caption    = "List &Computation"
    Enabled    = 0 'False
    Shortcut   = ^R
  End
  Begin Menu mnuFile_Print_Flat_Reinforce
    Caption    = "List &Reinforcement"
    Enabled    = 0 'False
    Visible    = 0 'False
  End
End
End
Begin Menu mnuFile_Print_
  Caption      = "-"
End
Begin Menu mnuFile_Print_Geometric
  Caption      = "&Geometric"
  Enabled      = 0 'False
  Shortcut     = ^G
End
Begin Menu mnuFile_Print_Shear
  Caption      = "&Shear Diagram"
  Enabled      = 0 'False
  Shortcut     = ^S
End
Begin Menu mnuFile_Print_Moment
  Caption      = "&Moment Diagram"
  Enabled      = 0 'False
  Shortcut     = ^M
End
End
Begin Menu mnuFile_2
  Caption      = "-"
End

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 38

```

Begin Menu mnuFile_Exit
  Caption = "E&xit"
End
End
Begin Menu mnuInput
  Caption = "&Input"
  Begin Menu mnuInput_Node
    Caption = "&Node"
    Enabled = 0 'False'
    Shortcut = {F4}
  End
  Begin Menu mnuInput_Element
    Caption = "&Design Strip Element"
    Enabled = 0 'False'
    Shortcut = {F5}
  End
  Begin Menu mnuInput_Load
    Caption = "&Load "
    Enabled = 0 'False'
    Shortcut = {F6}
  End
  Begin Menu mnuInput_Material
    Caption = "&Material Properties"
    Enabled = 0 'False'
    Shortcut = {F7}
  End
  Begin Menu mnuInput_Information
    Caption = "&Information"
    Enabled = 0 'False'
    Shortcut = {F8}
  End
End
Begin Menu mnuDesign
  Caption = "&Design"
  Begin Menu mnuDesign_Prestress
    Caption = "&Prestress"
    Enabled = 0 'False'
    Shortcut = {F11}
  End
  Begin Menu mnuDesign_Flat
    Caption = "&Flat Slab && Flat Plate"
    Enabled = 0 'False'
    Shortcut = {F12}
  End
End
Begin Menu mnuResult
  Caption = "&Result"
  Begin Menu mnuResult_Prestress
    Caption = "&Prestress"
    Begin Menu mnuResult_Prestress_Profile
      Caption = "&Profile Tendon"
      Enabled = .0 'False'
      Shortcut = ^K
    End
    Begin Menu mnuResult_Prestress_Layout
      Caption = "&Layout Tendon"
      Enabled = 0 'False'
    End
  End
End

```

FRMMAIN.FRM - 39

```

Shortcut      = ^O
Visible       = 0 'False
End
Begin Menu mnuResult_Prestress_LProfile
Caption       = "List Profile"
Enabled      = 0 'False
Shortcut     = ^U
End
Begin Menu mnuResult_Prestress_LCompute
Caption       = "List &Computation"
Enabled      = 0 'False
Shortcut     = ^V
End
End
Begin Menu mnuResult_Flat
Caption       = "&Flat Slab && Flat Plate"
Begin Menu mnuResult_Flat_LCompute
Caption       = "List &Computation"
Enabled      = 0 'False
Shortcut     = ^Q
End
Begin Menu mnuResult_Flat_LReinforce
Caption       = "List &Reinforcement"
Enabled      = 0 'False
Visible     = 0 'False
End
End
Begin Menu mnuResult_
Caption       = "-"
End
Begin Menu mnuResult_Geometric
Caption       = "&Geometric"
Enabled      = 0 'False
Shortcut     = ^X
End
Begin Menu mnuResult_Shear
Caption       = "&Shear Diagram"
Enabled      = 0 'False
Shortcut     = ^Y
End
Begin Menu mnuResult_Moment
Caption       = "&Moment Diagram"
Enabled      = 0 'False
Shortcut     = ^Z
End
End
Begin Menu mnuOption
Caption       = "&Option"
Begin Menu mnuOption_SaveSetting
Caption       = "&Save Setting "
Checked      = -1 'True
End
End
Begin Menu mnuWindow
Caption       = "&Window"
Begin Menu mnuWindow_ArrangeIcon
Caption       = "&Arrange Icon"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 40

```

        Shortcut      =   ^A
    End
    Begin Menu mnuWindow_Close
        Caption       =   "&Close All Windows"
        Enabled       =   0   'False
        Shortcut      =   ^W
    End
End
Begin Menu mnuHelp
    Caption         =   "&Help"
    Begin Menu mnuHelp_System
        Caption     =   "&System"
    End
    Begin Menu mnuHelp_About
        Caption     =   "&About"
    End
End
End
End

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 1

```

Sub Init_Screen_Resize ()
Dim LoadWidth As Integer
Dim NodeWidth As Integer
Dim kstWidth As Integer
Const kstTop = 30
Const kstRight = 30
LoadWidth = (grdLoad.Width - 300) \ grdLoad.Cols
NodeWidth = (grdNode.Width - 300) \ grdNode.Cols
kstWidth = kstCapLock.Width
For I = 0 To 4
    fmeTextInputData(I).Width = frmMain.Width - 860
Next I
frmMain.kstCapLock.Top = kstTop
frmMain.kstNumLock.Top = kstTop
frmMain.kstInsert.Top = kstTop
frmMain.kstScrollLock.Top = kstTop
frmMain.kstScrollLock.Left = frmMain.ScaleWidth - kstRight - kstWidth
frmMain.kstInsert.Left = frmMain.ScaleWidth - kstRight - (2 * kstWidth
)
frmMain.kstNumLock.Left = frmMain.ScaleWidth - kstRight - (3 * kstWidt
h)
frmMain.kstCapLock.Left = frmMain.ScaleWidth - kstRight - (4 * kstWidt
h)
'grdNode
For I = 0 To grdNode.Cols - 1
    grdNode.ColWidth(I) = NodeWidth
Next I
'grdLoad
For I = 0 To grdLoad.Cols - 1
    grdLoad.ColWidth(I) = LoadWidth
Next I
End Sub

Sub Init_ValueMain ()
'Toolbar1
Icon1(0, 0) = "new-up.bmp"
Icon1(0, 1) = "new-mds.bmp"
Icon1(1, 0) = "open-up.bmp"
Icon1(1, 1) = "open-mds.bmp"
Icon1(2, 0) = "save-up.bmp"
Icon1(2, 1) = "save-mds.bmp"
Icon1(3, 0) = "light-up.bmp"
Icon1(3, 1) = "light-dn.bmp"
Icon1(4, 0) = "lightfup.bmp"
Icon1(4, 1) = "lightfdn.bmp"
Icon1(5, 0) = "grd01-up.bmp"
Icon1(5, 1) = "grd01-dw.bmp"
Icon1(6, 0) = "Cals&mu.bmp"
Icon1(6, 1) = "Cals&md.bmp"
Icon1(7, 0) = "CalM&mu.bmp"
Icon1(7, 1) = "CalM&md.bmp"

'Toolbar2
Icon2(0, 0) = "new-up.bmp"
Icon2(0, 1) = "new-mds.bmp"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 2

```

Icon2(1, 0) = "grd01-up.bmp"
Icon2(1, 1) = "grd01-dw.bmp"
Icon2(2, 0) = "save-up.bmp"
Icon2(2, 1) = "save-mds.bmp"
Icon2(3, 0) = "save-up.bmp"
Icon2(3, 1) = "save-mds.bmp"
Icon2(4, 0) = "save-up.bmp"
Icon2(4, 1) = "save-mds.bmp"
Icon2(6, 0) = "Cals&mu.bmp"
Icon2(6, 1) = "Cals&md.bmp"
'ToolSelect
'init menu
Initial_Menu 1

```

End Sub

```

Sub mnuFile_Print_Prestress_Layout_Click ()
Print_Layout

```

End Sub

```

Sub chkCHead_Click (Index As Integer, Value As Integer)

```

```

If chkCHead(0).Value Then
GeneralData(CProj).SlabIndex = 1
End If
If chkCHead(1).Value Then
GeneralData(CProj).SlabIndex = 2
End If
If chkCHead(0).Value And chkCHead(1).Value Then
GeneralData(CProj).SlabIndex = 3
End If
If Not (chkCHead(0).Value) And Not (chkCHead(1).Value) Then
GeneralData(CProj).SlabIndex = 0
End If

```

End Sub

```

Sub chkInput_Click (Value As Integer)

```

```

'pnlWork.Top = 435
'pnlWork.Visible = Value
frmMain.Arrange 3

```

End Sub

```

Sub chkSystem_Click (Value As Integer)

```

```

pnlMenu.Top = 0
pnlMenu.Visible = Value
frmMain.Arrange 3

```

End Sub

```

Sub chkTextInputData_Click (Value As Integer)

```

```

pnlTextInputData.Visible = Value
frmMain.Arrange 3

```

End Sub

```

Sub cmbColumnEc_Change ()

```

```

GeneralData(CProj).CEc = CSng(cmbColumnEc.Text)

```

End Sub

```

Sub cmbColumnFc_Change ()

```

```

If cmbSlabFc_.Text <> "" Then

```

FRMMAIN.FRM - 3

```

    GeneralData(CProj).CFc = CSng(cmbColumnFc_.Text)
    cmbColumnEc.Text = Fc_ToEc(CSng(cmbColumnFc_.Text))
End If
End Sub

```

```

Sub cmbColumnFc_Click ()
    If cmbColumnFc_.Text <> "" Then
        GeneralData(CProj).CFc = CSng(cmbColumnFc_.Text)
        cmbColumnEc.Text = Fc_ToEc(CSng(cmbColumnFc_.Text))
    End If
End Sub

```

```

Sub cmbFs_Change ()
    If cmbFs_.Text <> "" Then
        GeneralData(CProj).RFs_ = CSng(cmbFs_.Text)
    End If
End Sub

```

```

Sub cmbFs_Click ()
    If cmbFs_.Text <> "" Then
        GeneralData(CProj).RFs_ = CSng(cmbFs_.Text)
    End If
End Sub

```

```

Sub cmbSFs_Change ()
    If cmbFs_.Text <> "" Then
        GeneralData(CProj).SFs_ = CSng(cmbSFs_.Text)
    End If
End Sub

```

```

Sub cmbSFs_Click ()
    If cmbFs_.Text <> "" Then
        GeneralData(CProj).SFs_ = CSng(cmbSFs_.Text)
    End If
End Sub

```

```

Sub cmbSlabEc_Change ()
    GeneralData(CProj).SEc = CSng(cmbSlabEc.Text)
End Sub

```

```

Sub cmbSlabEc_Click ()
    GeneralData(CProj).SEc = CSng(cmbSlabEc.Text)
End Sub

```

```

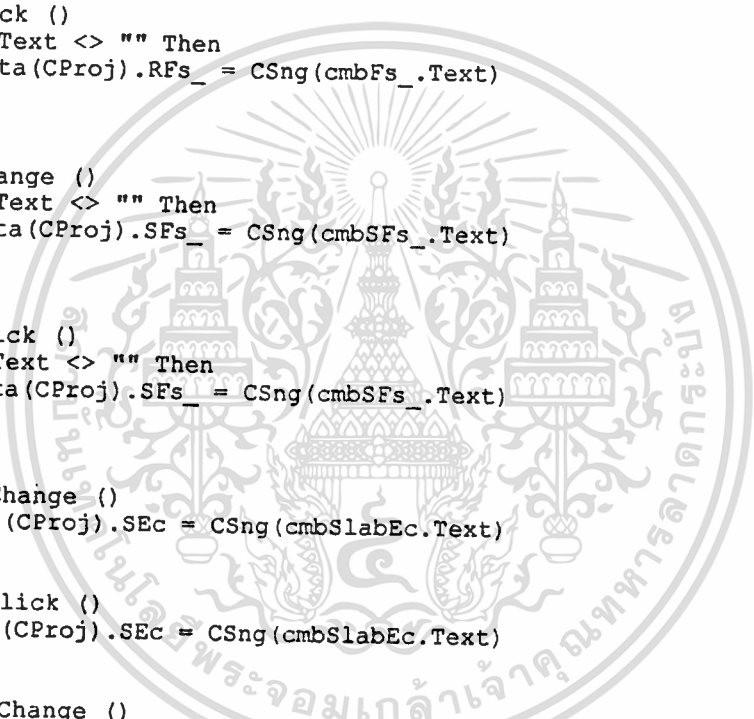
Sub cmbSlabFc_Change ()
    If cmbSlabFc_.Text <> "" Then
        GeneralData(CProj).SFc_ = CSng(cmbSlabFc_.Text)
        cmbSlabEc.Text = Fc_ToEc(CSng(cmbSlabFc_.Text))
    End If
End Sub

```

```

Sub cmbSlabFc_Click ()
    If cmbSlabFc_.Text <> "" Then
        GeneralData(CProj).SFc_ = CSng(cmbSlabFc_.Text)
        cmbSlabEc.Text = Fc_ToEc(CSng(cmbSlabFc_.Text))
    End If
End Sub

```



5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 4

```

Sub cmbSlabFc__DragOver (Source As Control, X As Single, Y As Single, State As Integer)
    pnlStatus.Caption = " concrete ultimate strength of slab ~210-400 ks
c"
End Sub

Sub cmbStrand_Click ()
    StrandIndex(CProj) = cmbStrand.ListIndex
    pnlStatus.Caption = " Select Tendon Type"
End Sub

Sub cmdNext_Click ()
    If CurrentPage < TextModule Then
        frmList(CProj).txtListCompute(CurrentPage).Visible = False
        CurrentPage = CurrentPage + 1
        frmList(CProj).txtListCompute(CurrentPage).Visible = True
        If CurrentPage = TextModule Then
            frmMain.cmdNext.Enabled = False
        Else
            frmMain.cmdNext.Enabled = True
        End If
        If CurrentPage = 1 Then
            frmMain.cmdPrevious.Enabled = False
        Else
            frmMain.cmdPrevious.Enabled = True
        End If
    End If
End Sub

Sub cmdNodeOK_Click ()
    Input_NodeData
End Sub

Sub cmdNodeOK_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    pnlStatus.Caption = " Click when data ready"
End Sub

Sub cmdPrevious_Click ()
    If CurrentPage > 1 Then
        frmList(CProj).txtListCompute(CurrentPage).Visible = False
        CurrentPage = CurrentPage - 1
        frmList(CProj).txtListCompute(CurrentPage).Visible = True
        If CurrentPage = TextModule Then
            frmMain.cmdNext.Enabled = False
        Else
            frmMain.cmdNext.Enabled = True
        End If
        If CurrentPage = 1 Then
            frmMain.cmdPrevious.Enabled = False
        Else
            frmMain.cmdPrevious.Enabled = True
        End If
    End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 5

```

Sub fmeDropOrCapital_MouseMove (Button As Integer, Shift As Integer, X As
Single, Y As Single)
    pnlStatus.Caption = " Select type of column,Drop panel or capital or
both"
End Sub

Sub fmeEdgeYIndex_MouseMove (Button As Integer, Shift As Integer, X As Sin
gle, Y As Single)
    pnlStatus.Caption = " Edge note,in Y axis edge in top or bottom or no
edge(Interior)"
End Sub

Sub fmeNode_MouseMove (Button As Integer, Shift As Integer, X As Single, Y
As Single)
    pnlStatus.Caption = " Input Column Data"
End Sub

Sub fmeReinforce_MouseMove (Button As Integer, Shift As Integer, X As Sing
le, Y As Single)
    pnlStatus.Caption = " Yeild strength of reinforce steel"
End Sub

Sub fmeSlab_MouseMove (Button As Integer, Shift As Integer, X As Single, Y
As Single)
    pnlStatus.Caption = " Properties of concrete in slab and column"
End Sub

Sub fmeSlabSystem_MouseMove (Button As Integer, Shift As Integer, X As Sin
gle, Y As Single)
    pnlStatus.Caption = " Slab system,Select flat plate or flat slab"
End Sub

Sub fmeStrand_MouseMove (Button As Integer, Shift As Integer, X As Single,
Y As Single)
    pnlStatus.Caption = " Breaking strength of 1 unit strand"
End Sub

Sub fmeStrandSystem_DragDrop (Source As Control, X As Single, Y As Single)
    pnlStatus.Caption = " Method to Construct ,Bond or Unbond"
End Sub

Sub fmeTextInputData_MouseMove (Index As Integer, Button As Integer, Shift
As Integer, X As Single, Y As Single)
    pnlStatus.Caption = " Ready..."
End Sub

Sub Frame3D1_MouseMove (Button As Integer, Shift As Integer, X As Single,
Y As Single)
    pnlStatus.Caption = " Edge note,in X axis edge in left or right or no
edge(Interior)"
End Sub

Sub gphSelectInput_Click (Index As Integer, Value As Integer)
    fmeTextInputData(InputTextIndex).Visible = False
    InputTextIndex = Index
    fmeTextInputData(InputTextIndex).Visible = gphSelectInput(InputTextInd
ex).Value

```

FRMMAIN.FRM - 6

```

'''' Set to False
mnuInput_Node.Checked = False
mnuInput_Load.Checked = False
mnuInput_Material.Checked = False
mnuInput_Information.Checked = False
mnuInput_Element.Checked = False

Select Case Index
Case 0: mnuInput_Node.Checked = gphSelectInput(InputTextIndex).Value
Case 1: mnuInput_Load.Checked = gphSelectInput(InputTextIndex).Value
Case 2: mnuInput_Element.Checked = gphSelectInput(InputTextIndex).Value
Case 3: mnuInput_Material.Checked = gphSelectInput(InputTextIndex).Value
Case 4: mnuInput_Information.Checked = gphSelectInput(InputTextIndex).Value
End Select
'If InputTextIndex = 0 Then
' grdNode.SetFocus
'End If
End Sub

Sub grdElement_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    pnlStatus.Caption = " Element data for design, Separate in line and line element"
End Sub

Sub grdLoad_Click ()
    'Dim a, b, c As Integer
    'If ((frmMain.grdLoad.Row > 1) And (frmMain.grdLoad.Row < SLoadNum(CPro
j))) Then
    ' For I = 1 To 2
    '   frmMain.grdLoad.Row = frmMain.grdLoad.Row: frmMain.grdLoad.Col = I
    '   frmMain.txtLoad(I) = frmMain.grdLoad.Text
    ' Next I
    '   frmMain.grdLoad.Col = 3
    '   a = CInt(frmMain.grdLoad.Text)
    '   frmMain.grdLoad.Col = 4
    '   b = CInt(frmMain.grdLoad.Text)
    '   frmMain.grdLoad.Col = 5
    '   c = CInt(frmMain.grdLoad.Text)
    '   frmMain.optLoadSupport(a).Value = True
    '   frmMain.optNodeSupport(b).Value = True
    '   frmMain.optNodeSupport(c).Value = True
    'End If
End Sub

Sub grdNode_Click ()
    Dim a, b, c As Integer
    If ((frmMain.grdNode.Row >= 1) And (frmMain.grdNode.Row <= SNodeNum(CPro
roj))) Then
        frmMain.grdNode.Row = frmMain.grdNode.Row
        frmMain.grdNode.Col = 0
        frmMain.txtNode(1) = frmMain.grdNode.Text
        frmMain.grdNode.Col = 1
    
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 7

```

frmMain.txtNode(2) = frmMain.grdNode.Text
frmMain.grdNode.Col = 2
frmMain.txtNode(3) = frmMain.grdNode.Text
frmMain.grdNode.Col = 3
b = CInt(frmMain.grdNode.Text)
frmMain.grdNode.Col = 4
c = CInt(frmMain.grdNode.Text)
frmMain.grdNode.Col = 5
a = CInt(frmMain.grdNode.Text)
frmMain.optNodeSupport(a).Value = True
frmMain.optEdgeX(b).Value = True
frmMain.optEdgeY(c).Value = True
frmMain.txtNode(4).Text = "1"
frmMain.txtNode(5).Text = "1"
End If
End Sub

Sub grdNode_DblClick ()
'If (grdNode.Row > 0) And (grdNode.Col > 0) Then
'  grdNode_KeyPress 13
'End If
End Sub

Sub grdNode_KeyPress (KeyAscii As Integer)
'Dim Char As String
'Select Case KeyAscii
'Case 27: 'Escape
'Case 9: 'Tab
'  If Shifting Then
'    If grdNode.Col > 1 Then
'      grdNode.Col = grdNode.Col - 1
'    End If
'  Else
'    If grdNode.Col < (grdNode.Cols - 1) Then
'      grdNode.Col = grdNode.Col + 1
'    End If
'  End If
'  UnSelect
'Case Else:
'  If KeyAscii = 13 Then
'    txtGrid.Text = grdNode.Text
'    txtGrid.SelStart = Len(txtGrid.Text)
'  Else
'    Char = Chr$(KeyAscii)
'    txtGrid.Text = Char
'    txtGrid.SelStart = 1
'  End If
'  ShowTextBox
'  KeyAscii = 0
'End Select
If KeyAscii = 8 Then
  If ((frmMain.grdNode.Row >= 1) And (frmMain.grdNode.Row <= SNodeNum(C
Proj))) Then
    SNodeNum(CProj) = SNodeNum(CProj) - 1
    For I = frmMain.grdNode.Row To SNodeNum(CProj)
      Column(CProj, I) = Column(CProj, I + 1)
    Next I
  End If
End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 8

```

    End If
    End If
    Show_Data
End Sub

Sub grdNode_MouseDown (Button As Integer, Shift As Integer, X As Single, Y
As Single)
    txtGrid.Visible = False
End Sub

Sub grdNode_MouseMove (Button As Integer, Shift As Integer, X As Single, Y
As Single)
    pnlStatus.Caption = " Display column data such as coordinate,length,c
olumn size,etc."
End Sub

Sub imgTooll_MouseDown (Index As Integer, Button As Integer, Shift As Inte
ger, X As Single, Y As Single)
    imgTooll(Index).Picture = LoadPicture(DirApp + "picico\" + Icon1(Index
, 1))
    IconClick1 = True
End Sub

Sub imgTooll_MouseMove (Index As Integer, Button As Integer, Shift As Inte
ger, X As Single, Y As Single)
    If Not (X >= 0 And X <= 360 And Y >= 0 And Y <= 330) Then
        IconClick1 = False
        imgTooll(Index).Picture = LoadPicture(DirApp + "picico\" + Icon1(Inde
x, 1))
    End If
End Sub

Sub imgTooll_MouseUp (Index As Integer, Button As Integer, Shift As Intege
r, X As Single, Y As Single)
    If X >= 0 And X <= 360 And Y >= 0 And Y <= 330 And IconClick1 Then
        Select Case Index
        Case 0:
            New_File
        Case 1:
            Open_File
        Case 2:
            Save_File
        Case 3:
            pnlStatus.Caption = " Design processing...."
            Screen.MousePointer = 11
            If SNodeNum(CProj) <> 0 Then
                Design_PSlab
            ElseIf SNodeNum(CProj) = 0 Then
                MsgBox "Data Error. Please input column data.", 0 + 48, "Data Ero
r."
            End If
            Screen.MousePointer = 0
            pnlStatus.Caption = " Ready...."
        Case 4:
            pnlStatus.Caption = " Design processing...."
            Screen.MousePointer = 11
            If SNodeNum(CProj) <> 0 Then

```

FRMMAIN.FRM - 9

```

    Design_FSlab
    ElseIf SNodeNum(CProj) = 0 Then
        MsgBox "Data Error. Please input column data.", 0 + 48, "Data Error."
    End If
    Screen.MousePointer = 0
    pnlStatus.Caption = " Ready...."
Case 5:
    Screen.MousePointer = 11
    frmMain.pnlStatus.Caption = " Generating slab form...."
    If SNodeNum(CProj) <> 0 Then
        Screen_Geometric
    ElseIf SNodeNum(CProj) = 0 Then
        MsgBox "Data Error. Please input column data.", 0 + 48, "Data Error."
    End If
    frmMain.pnlStatus.Caption = " Ready...."
    Screen.MousePointer = 0
Case 6:
    Screen.MousePointer = 11
    frmMain.pnlStatus.Caption = " Generating slab form...."
    If SNodeNum(CProj) <> 0 Then
        Screen_Shear
    ElseIf SNodeNum(CProj) = 0 Then
        MsgBox "Data Error. Please input column data.", 0 + 48, "Data Error."
    End If
    frmMain.pnlStatus.Caption = " Ready...."
    Screen.MousePointer = 0
Case 7:
    Screen.MousePointer = 11
    frmMain.pnlStatus.Caption = " Generating slab form...."
    If SNodeNum(CProj) <> 0 Then
        Screen_Moment
    ElseIf SNodeNum(CProj) = 0 Then
        MsgBox "Data Error. Please input column data.", 0 + 48, "Data Error."
    End If
    frmMain.pnlStatus.Caption = " Ready...."
    Screen.MousePointer = 0
End Select
End If
imgTool1(Index).Picture = LoadPicture(DirApp + "picico\" + Icon1(Index
, 0))
End Sub

Sub imgWorkTool1_DragDrop (Source As Control, X As Single, Y As Single)
    imgWorkTool1.Picture = LoadPicture(DirApp + "picico\" + Icon2(Index,
0))
    Unload frmWorkTool1
End Sub

Sub imgWorkTool1_MouseDown (Button As Integer, Shift As Integer, X As Single, Y As Single)
    imgWorkTool1.Picture = LoadPicture(DirApp + "picico\" + Icon2(0, 1))
    imgWorkTool1.Drag 1
    Dim ToolNum As Integer

```

FRMMAIN.FRM - 10

```

ToolNum = 0
If frmMain.chkSystem Then ToolNum = ToolNum + 1
If frmMain.chkInput Then ToolNum = ToolNum + 1
'frmMain.Panel3D1.Visible = True
frmWorkTool1.Top = frmMain.Top + 620 + frmMain.pnlMenu.Height * ToolNum
m'frmMain.Height - (frmMain.SScaleHeight * Screen.TwipsPerPixelY) + (frmMain.pnlMenu.Height * ToolNum)
frmWorkTool1.Left = frmMain.Left + frmMain.imgWorkTool1.Left + 80' (frmMain.Width / 2)
Load frmWorkTool1
frmWorkTool1.Show
frmWorkTool1.SetFocus
End Sub

Sub imgWorkTool2_DragDrop (Source As Control, X As Single, Y As Single)
imgWorkTool2.Picture = LoadPicture(DirApp + "picico\" + Icon2(1, 0))
Unload frmWorkTool2
End Sub

Sub imgWorkTool2_MouseDown (Button As Integer, Shift As Integer, X As Single, Y As Single)
imgWorkTool2.Picture = LoadPicture(DirApp + "picico\" + Icon2(1, 1))
imgWorkTool2.Drag 1
Dim ToolNum As Integer
ToolNum = 0
If frmMain.chkSystem Then ToolNum = ToolNum + 1
If frmMain.chkInput Then ToolNum = ToolNum + 1
'frmMain.Panel3D1.Visible = True
frmWorkTool2.Top = frmMain.Top + 620 + frmMain.pnlMenu.Height * ToolNum
m'frmMain.Height - (frmMain.SScaleHeight * Screen.TwipsPerPixelY) + (frmMain.pnlMenu.Height * ToolNum)
frmWorkTool2.Left = frmMain.Left + frmMain.imgWorkTool2.Left + 60' (frmMain.Width / 2)
'Work Tool2 ,Column Dimension
Load frmWorkTool2
frmWorkTool2.Show
frmWorkTool2.SetFocus
End Sub

Sub imgWorkTool3_MouseDown (Button As Integer, Shift As Integer, X As Single, Y As Single)
imgTooll1(Index).Picture = LoadPicture(DirApp + "picico\" + Icon1(Index, 1))
IconClick1 = True
End Sub

Sub imgWorkTool3_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
If Not (X >= 0 And X <= 360 And Y >= 0 And Y <= 330) Then
IconClick1 = False
imgTooll1(Index).Picture = LoadPicture(DirApp + "picico\" + Icon1(Index, 1))
End If
End Sub

Sub imgWorkTool5_MouseDown (Button As Integer, Shift As Integer, X As Single, Y As Single)

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 11

```

    imgWorkTool5.Picture = LoadPicture(DirApp + "picico\" + Icon1(4, 1))
    IconClick1 = True
End Sub

Sub imgWorkTool5_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    If Not (X >= 0 And X <= 360 And Y >= 0 And Y <= 330) Then
        IconClick1 = False
        imgWorkTool5.Picture = LoadPicture(DirApp + "picico\" + Icon1(4, 1))
    End If
End Sub

Sub imgWorkTool5_MouseUp (Button As Integer, Shift As Integer, X As Single, Y As Single)
    If X >= 0 And X <= 360 And Y >= 0 And Y <= 330 And IconClick1 Then
        Screen.MousePointer = 11
        pnlStatus.Caption = "Wait...."
        If FileIndex <> -1 Then
            BCal
        Else
            Screen.MousePointer = 0
            Beep
            MsgBox "Load data first.", 0 + 48, "Data error."
        End If
        pnlStatus.Caption = "Ready...."
        Screen.MousePointer = 0
    End If
    imgWorkTool5.Picture = LoadPicture(DirApp + "picico\" + Icon1(4, 0))
End Sub

Sub imgWorkTool6_MouseDown (Button As Integer, Shift As Integer, X As Single, Y As Single)
    imgWorkTool6.Picture = LoadPicture(DirApp + "picico\" + Icon2(6, 1))
    IconClick1 = True
End Sub

Sub imgWorkTool6_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    If Not (X >= 0 And X <= 360 And Y >= 0 And Y <= 330) Then
        IconClick1 = False
        imgWorkTool6.Picture = LoadPicture(DirApp + "picico\" + Icon2(6, 1))
    End If
End Sub

Sub imgWorkTool6_MouseUp (Button As Integer, Shift As Integer, X As Single, Y As Single)
    If X >= 0 And X <= 360 And Y >= 0 And Y <= 330 And IconClick1 Then
        Screen.MousePointer = 11
        pnlStatus.Caption = "Wait...."
        If FileIndex <> -1 Then
            Load frmInputDesign
            frmInputDesign.Top = (Screen.Height - frmInputDesign.Height) / 2
            frmInputDesign.Left = (Screen.Width - frmInputDesign.Width) / 2
            frmInputDesign.Show MODEL
        Else
            Screen.MousePointer = 0
            Beep
        End If
    End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 12

```

    MsgBox "Load data first.", 0 + 48, "Data error."
End If
pnlStatus.Caption = "Ready...."
Screen.MousePointer = 0
End If
imgWorkTool6.Picture = LoadPicture(DirApp + "picico\" + Icon2(6, 0))
End Sub

Sub MDIForm_DragDrop (Source As Control, X As Single, Y As Single)
'Unload frmWorkTool2 'try
End Sub

Sub MDIForm_Load ()
Init_DataValue
Init_ValueMain
End Sub

Sub MDIForm_Resize ()
If WindowState <> 1 Then
Init_Screen_Resize
End If
End Sub

Sub mnuDesign_Flat_Click ()
mnuDesign_Prestress.Checked = False
mnuDesign_Flat.Checked = True
pnlStatus.Caption = " Design processing...."
Screen.MousePointer = 11
If SNodeNum(CProj) <> 0 Then
Design FSlab
Initial_Menu 4
ElseIf SNodeNum(CProj) = 0 Then
MsgBox "Data Error. Please input column data.", 0 + 48, "Data Error."
End If
Screen.MousePointer = 0
pnlStatus.Caption = " Ready...."
End Sub

Sub mnuDesign_Prestress_Click ()
mnuDesign_Flat.Checked = False
mnuDesign_Prestress.Checked = True
pnlStatus.Caption = " Design processing...."
Screen.MousePointer = 11
If SNodeNum(CProj) <> 0 Then
Design PSlab
Initial_Menu 3
ElseIf SNodeNum(CProj) = 0 Then
MsgBox "Data Error. Please input column data.", 0 + 48, "Data Error."
End If
Screen.MousePointer = 0
pnlStatus.Caption = " Ready...."
End Sub

Sub mnuFile_Exit_Click ()
Ready_Exit
End
End Sub

```

FRMMAIN.FRM - 13

```

Sub mnuFile_New_Click ()
  New_File
End Sub

Sub mnuFile_Open_Click ()
  Open_File
End Sub

Sub mnuFile_Print_Flat_LCompute_Click ()
  Screen.MousePointer = 11
  frmMain.pnlStatus.Caption = " Listing Computation For Flate slab & Fl
at Plate system...."
  Print_FlatListCompute
  frmMain.pnlStatus.Caption = " Ready...."
  Screen.MousePointer = 0
End Sub

Sub mnuFile_Print_Geometric_Click ()
  Screen.MousePointer = 11
  frmMain.pnlStatus.Caption = " Printing Slab...."
  Print_Geometric
  frmMain.pnlStatus.Caption = " Ready...."
  Screen.MousePointer = 0
End Sub

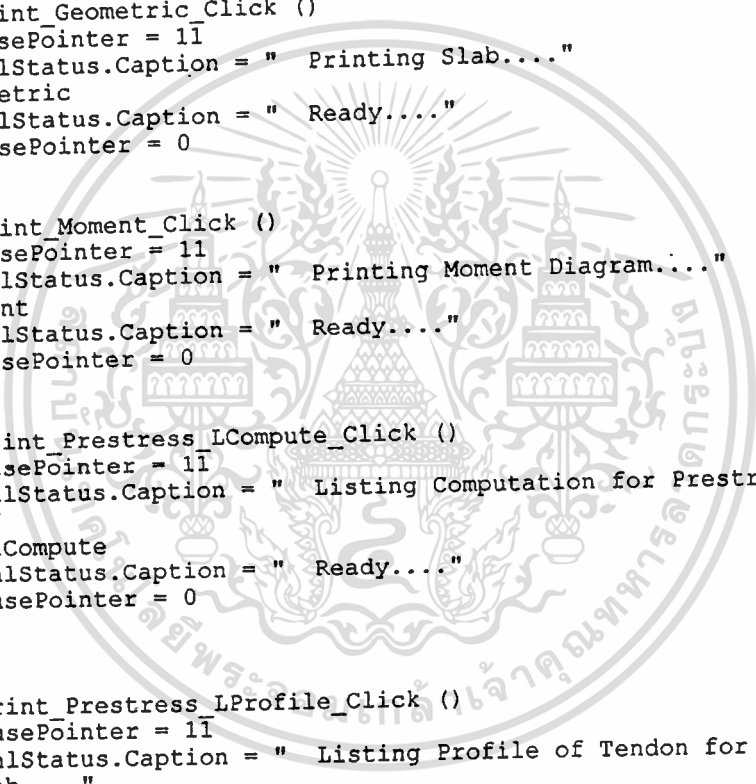
Sub mnuFile_Print_Moment_Click ()
  Screen.MousePointer = 11
  frmMain.pnlStatus.Caption = " Printing Moment Diagram...."
  Print_Moment
  frmMain.pnlStatus.Caption = " Ready...."
  Screen.MousePointer = 0
End Sub

Sub mnuFile_Print_Prestress_LCompute_Click ()
  Screen.MousePointer = 11
  frmMain.pnlStatus.Caption = " Listing Computation for Prestress-Tensi
oned slab...."
  Print_ListCompute
  frmMain.pnlStatus.Caption = " Ready...."
  Screen.MousePointer = 0
End Sub

Sub mnuFile_Print_Prestress_LProfile_Click ()
  Screen.MousePointer = 11
  frmMain.pnlStatus.Caption = " Listing Profile of Tendon for Prestress
-Tensioned slab...."
  Print_ListProfile
  frmMain.pnlStatus.Caption = " Ready...."
  Screen.MousePointer = 0
End Sub

Sub mnuFile_Print_Prestress_Profile_Click ()
  Screen.MousePointer = 11
  frmMain.pnlStatus.Caption = " Printing Profile of Tendon...."

```



FRMMAIN.FRM - 14

```
Print_Profile
frmMain.pnlStatus.Caption = " Ready...."
Screen.MousePointer = 0
```

End Sub

```
Sub mnuFile_Print_Shear_Click ()
Screen.MousePointer = 11
frmMain.pnlStatus.Caption = " Printing Shear Diagram...."
Print_Shear
frmMain.pnlStatus.Caption = " Ready...."
Screen.MousePointer = 0
```

End Sub

```
Sub mnuFile_Save_Click ()
Save_File
```

End Sub

```
Sub mnuFile_SaveAs_Click ()
SaveAs_File
```

End Sub

```
Sub mnuHelp_About_Click ()
frmMain.pnlStatus.Caption = " About Program...."
frmAbout.Top = (Screen.Height - frmAbout.Height) \ 2
frmAbout.Left = (Screen.Width - frmAbout.Width) \ 2
Load frmAbout
frmAbout.Show MODAL
frmMain.pnlStatus.Caption = " Ready...."
'Print_Try
```

End Sub

```
Sub mnuHelp_System_Click ()
frmSysInfo.Top = (Screen.Height - frmSysInfo.Height) / 2
frmSysInfo.Left = (Screen.Width - frmSysInfo.Width) / 2
frmSysInfo.Show MODAL
'Screen_Try
```

End Sub

```
Sub mnuInput_Element_Click ()
frmMain.mnuInput_Node.Checked = False
frmMain.mnuInput_Element.Checked = False
frmMain.mnuInput_Information.Checked = False
frmMain.mnuInput_Material.Checked = False
frmMain.mnuInput_Load.Checked = False
frmMain.gphSelectInput(2).Value = True
```

End Sub

```
Sub mnuInput_Information_Click ()
frmMain.mnuInput_Node.Checked = False
frmMain.mnuInput_Element.Checked = False
frmMain.mnuInput_Information.Checked = False
frmMain.mnuInput_Material.Checked = False
frmMain.mnuInput_Load.Checked = False
frmMain.gphSelectInput(4).Value = True
```

End Sub

FRMMAIN.FRM - 15

```

Sub mnuInput_Load_Click ()
    frmMain.mnuInput_Node.Checked = False
    frmMain.mnuInput_Element.Checked = False
    frmMain.mnuInput_Information.Checked = False
    frmMain.mnuInput_Material.Checked = False
    frmMain.mnuInput_Load.Checked = False
    frmMain.gphSelectInput(1).Value = True
End Sub

Sub mnuInput_Material_Click ()
    frmMain.mnuInput_Node.Checked = False
    frmMain.mnuInput_Element.Checked = False
    frmMain.mnuInput_Information.Checked = False
    frmMain.mnuInput_Material.Checked = False
    frmMain.mnuInput_Load.Checked = False
    frmMain.gphSelectInput(3).Value = True
End Sub

Sub mnuInput_Node_Click ()
    frmMain.mnuInput_Node.Checked = False
    frmMain.mnuInput_Element.Checked = False
    frmMain.mnuInput_Information.Checked = False
    frmMain.mnuInput_Material.Checked = False
    frmMain.mnuInput_Load.Checked = False
    frmMain.gphSelectInput(0).Value = True
End Sub

Sub mnuOption_SaveSetting_Click ()
    If mnuOption_saveSetting.Checked Then
        mnuOption_saveSetting.Checked = False
    Else
        mnuOption_saveSetting.Checked = True
    End If
End Sub

Sub mnuResult_Flat_ICompute_Click ()
    Screen.MousePointer = 11
    frmMain.pnlStatus.Caption = " Listing computing...."
    Screen.FlatListCompute
    frmMain.pnlStatus.Caption = " Ready...."
    Screen.MousePointer = 0
End Sub

Sub mnuResult_Geometric_Click ()
    Screen.MousePointer = 11
    frmMain.pnlStatus.Caption = " Generating slab form...."
    Screen.Geometric
    frmMain.pnlStatus.Caption = " Ready...."
    Screen.MousePointer = 0
End Sub

Sub mnuResult_Moment_Click ()
    Screen.MousePointer = 11
    frmMain.pnlStatus.Caption = " Generating moment diagram...."
    Screen.Moment
    frmMain.pnlStatus.Caption = " Ready...."
    Screen.MousePointer = 0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 16

End Sub

```
Sub mnuResult_Prestress_LCompute_Click ()
  Screen.MousePointer = 11
  frmMain.pnlStatus.Caption = " Listing computing...."
  Screen_ListCompute
  frmMain.pnlStatus.Caption = " Ready...."
  Screen.MousePointer = 0
```

End Sub

```
Sub mnuResult_Prestress_LProfile_Click ()
  Screen.MousePointer = 11
  frmMain.pnlStatus.Caption = " Listing profile...."
  Screen_ListProfile
  frmMain.pnlStatus.Caption = " Ready...."
  Screen.MousePointer = 0
```

End Sub

```
Sub mnuResult_Prestress_Profile_Click ()
  Screen.MousePointer = 11
  frmMain.pnlStatus.Caption = " Generating strand profile...."
  Screen_Profile
  frmMain.pnlStatus.Caption = " Ready...."
  Screen.MousePointer = 0
```

End Sub

```
Sub mnuResult_Shear_Click ()
  Screen.MousePointer = 11
  frmMain.pnlStatus.Caption = " Generating shear diagram...."
  Screen_Shear
  frmMain.pnlStatus.Caption = " Ready...."
  Screen.MousePointer = 0
```

End Sub

```
Sub mnuWindow_ArrangeIcon_Click ()
  frmMain.Arrange 3
```

End Sub

```
Sub mnuWindow_Close_Click ()
  Unload frmList(CProj)
  Unload frmData(CProj)
  Unload frmOutGraph(CProj)
  Initial_Menu 1
```

End Sub

```
Sub optBond_Click (Index As Integer, Value As Integer)
  If Value Then
    StrandIndex(CProj) = Index
```

End Sub

```
Sub optSystem_Click (Index As Integer, Value As Integer)
  Select Case Index
  Case 0:
    If Value Then
      GeneralData(CProj).SlabIndex = Index
      chkCHead(0).Value = False
```

FRMMAIN.FRM - 17

```

chkCHead(0).Enabled = False
chkCHead(1).Value = False
chkCHead(1).Enabled = False
End If
Case 1:
If Value Then
If chkCHead(0).Value Then
GeneralData(CProj).SlabIndex = 1
End If
If chkCHead(1).Value Then
GeneralData(CProj).SlabIndex = 2
End If
If chkCHead(0).Value And chkCHead(1).Value Then
GeneralData(CProj).SlabIndex = 3
End If
If Not (chkCHead(0).Value) And Not (chkCHead(1).Value)
GeneralData(CProj).SlabIndex = 0
End If
chkCHead(0).Enabled = True
chkCHead(1).Enabled = True
End If
End Select
End Sub

Sub optSystem_MouseMove (Index As Integer, Button As Integer, X As Single, Y As Single)
If Index = 1 Then pnlStatus.Caption = " Flat slab,slab
1 or capital or both"
If Index = 0 Then pnlStatus.Caption = " Flat plate,slab
panel and capital"
End Sub

Sub Panel3D1_MouseMove (Button As Integer, Shift As Integer, Y As Single)
pnlStatus.Caption = " Select Tendon Type"
End Sub

Sub pnlStatus_MouseMove (Button As Integer, Shift As Integer, Y As Single)
pnlStatus.Caption = " Status bar,Information data"
End Sub

Sub pnlTextInputData_MouseMove (Button As Integer, Shift As Integer, Y As Single)
pnlStatus.Caption = " Ready...."
End Sub

Sub pnlThick_MouseMove (Button As Integer, Shift As Integer, Y As Single)
pnlStatus.Caption = " Slab Thickness [Output]"
End Sub

Sub pnlThick_U_MouseMove (Index As Integer, Button As Integer, X As Single, Y As Single)
pnlStatus.Caption = " Slab Thickness [Output]"
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 18

Sub tmrTime_Timer ()

```
txtTime.Text = Format$(Now, "hh:mm:ss AM/PM")
pnlT.Caption = Format$(Now, "dddd mm/dd/yyyy hh:mm:ss AM/PM ")
```

End Sub

Sub txtAddress_KeyPress (KeyAscii As Integer)

```
If KeyAscii = 13 Then
    Input_Data 125, txtAddress
End If
```

End Sub

Sub txtAddress_LostFocus ()

```
' Input_Data 125, txtAddress
```

End Sub

Sub txtAddress_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)

```
pnlStatus.Caption = " Information data "
```

End Sub

Sub txtDate_KeyPress (KeyAscii As Integer)

```
If KeyAscii = 13 Then
    Input_Data 124, txtDate
End If
```

End Sub

Sub txtDate_LostFocus ()

```
' Input_Data 124, txtDate
```

End Sub

Sub txtDate_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)

```
pnlStatus.Caption = " Information data "
```

End Sub

Sub txtDL_KeyPress (KeyAscii As Integer)

```
If KeyAscii = 13 Then
    DeadLoad(CProj) = CSng(txtDL.Text)
    txtLL.SetFocus
End If
```

End Sub

Sub txtDL_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)

```
pnlStatus.Caption = " Actual Dead load at complete design"
```

End Sub

Sub txtDLFactor_KeyPress (KeyAscii As Integer)

```
If KeyAscii = 13 Then
```

```
DeadCo = CSng(txtDLFactor.Text)
txtLL.SetFocus
End If
End Sub

Sub txtEngineer_KeyPress (KeyAscii As Integer)
If KeyAscii = 13 Then
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 19

```

    Input_Data 126, txtEngineer
End If
End Sub

Sub txtEngineer_LostFocus ()
' Input_Data 126, txtEngineer
End Sub

Sub txtEngineer_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    pnlStatus.Caption = " Information data "
End Sub

Sub txtFileName_KeyPress (KeyAscii As Integer)
    If KeyAscii = 13 Then
        Input_Data 121, txtFileName
    End If
End Sub

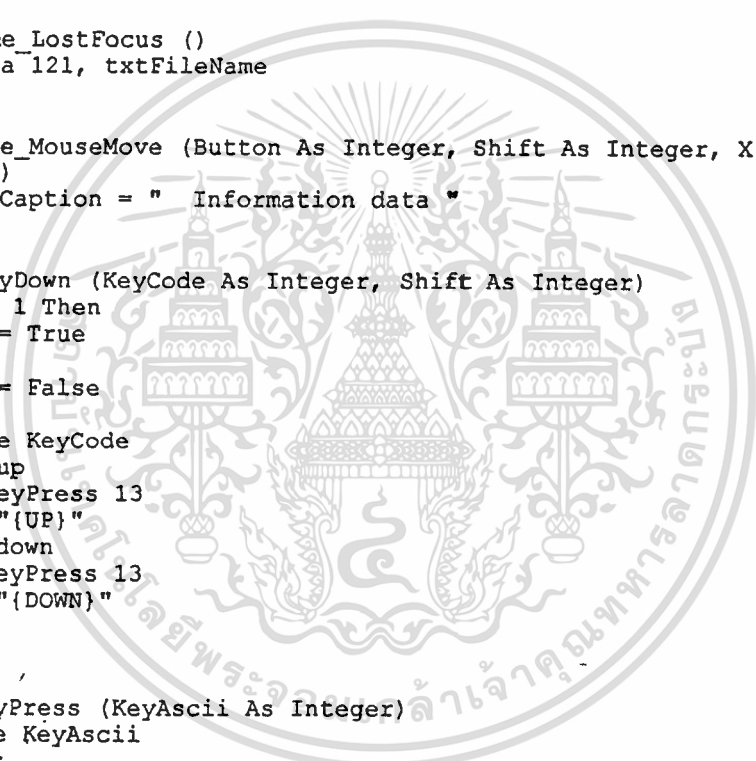
Sub txtFileName_LostFocus ()
'Input_Data 121, txtFileName
End Sub

Sub txtFileName_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    pnlStatus.Caption = " Information data "
End Sub

Sub txtGrid_KeyDown (KeyCode As Integer, Shift As Integer)
    If Shift = 1 Then
        Shifting = True
    Else
        Shifting = False
    End If
    Select Case KeyCode
    Case 38: 'up
        txtGrid_KeyPress 13
        SendKeys "{UP}"
    Case 40: 'down
        txtGrid_KeyPress 13
        SendKeys "{DOWN}"
    End Select
End Sub

Sub txtGrid_KeyPress (KeyAscii As Integer)
Select Case KeyAscii
Case 13, 9:
    grdNode.Text = txtGrid.Text
    txtGrid.Visible = False
    grdNode.SetFocus
    If (KeyAscii = 9) And (grdNode.Col < (grdNode.Cols - 1)) Then
        If grdNode.Col > 1 Then
            grdNode.Col = grdNode.Col - 1
        End If
    Else
        If grdNode.Col < (grdNode.Cols - 1) Then
            grdNode.Col = grdNode.Col + 1
        End If
    End Select

```



FRMMAIN.FRM - 20

```

    End If
  End If
  UnSelect
  Case 27:
    KeyAscii = 0
    txtGrid.Visible = False
    grdNode.SetFocus
  End Select
End Sub

Sub txtInformation_KeyPress (KeyAscii As Integer)
  If KeyAscii = 13 Then
    Input_Data 127, txtInformation
  End If
End Sub

Sub txtInformation_LostFocus ()
  Input_Data 127, txtInformation
End Sub

Sub txtInformation_MouseMove (Button As Integer,
  X As Integer, Y As Single)
  pnlStatus.Caption = " Information data "
End Sub

Sub txtLL_KeyPress (KeyAscii As Integer)
  If KeyAscii = 13 Then
    LiveLoad(CProj) = CSng(txtLL.Text)
    txtSL.SetFocus
  End If
End Sub

Sub txtLL_MouseMove (Button As Integer, Shift As Boolean,
  X As Integer, Y As Single)
  pnlStatus.Caption = " Live Load"
End Sub

Sub txtLLFactor_KeyPress (KeyAscii As Integer)
  If KeyAscii = 13 Then
    LiveCo = CSng(txtLLFactor.Text)
    txtLL.SetFocus
  End If
End Sub

Sub txtLoadEnd_KeyPress (KeyAscii As Integer)
  If KeyAscii = 13 Then
    Input_Data 66, txtLoadEnd
  End If
End Sub

Sub txtLoadEnd_LostFocus ()
  Input_Data 66, txtLoadEnd
End Sub

Sub txtLoadStart_KeyPress (KeyAscii As Integer)
  If KeyAscii = 13 Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 21

```

    Input_Data 65, txtLoadStart
End If
End Sub

Sub txtLoadStart_LostFocus ()
    Input_Data 65, txtLoadStart
End Sub

Sub txtLoadX1_KeyPress (KeyAscii As Integer)
    If KeyAscii = 13 Then
        Input_Data 61, txtLoadX1
    End If
End Sub

Sub txtLoadX1_LostFocus ()
    Input_Data 61, txtLoadX1
End Sub

Sub txtLoadX2_KeyPress (KeyAscii As Integer)
    If KeyAscii = 13 Then
        Input_Data 63, txtLoadX2
    End If
End Sub

Sub txtLoadX2_LostFocus ()
    Input_Data 63, txtLoadX2
End Sub

Sub txtLoadY1_KeyPress (KeyAscii As Integer)
    If KeyAscii = 13 Then
        Input_Data 62, txtLoadY1
    End If
End Sub

Sub txtLoadY1_LostFocus ()
    Input_Data 62, txtLoadY1
End Sub

Sub txtLoadY2_KeyPress (KeyAscii As Integer)
    If KeyAscii = 13 Then
        Input_Data 64, txtLoadY2
    End If
End Sub

Sub txtLoadY2_LostFocus ()
    Input_Data 64, txtLoadY2
End Sub

Sub txtNode_Change (Index As Integer)
    'Dim N As Integer
    'Dim R As Integer
    'If txtNode(1).Text <> "" Then
    ' Select Case Index
    ' Case 1:
    ' On Error GoTo ErrorData2
    ' N = CInt(txtNode(Index).Text)
    ' R = Check_Error(1, (txtNode(Index).Text))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRMMAIN.FRM - 22

```

' If R <> 0 Then
'   Error_Msg R
' End If
' cmdNodeOK.Enabled = True
' If (N > 0 And N < SNodeNum(CProj)) Then
'   txtNode(2).Text = Column(CProj, N).Co_X
'   txtNode(3).Text = Column(CProj, N).Co_Y
'   txtNode(4).Text = 1
'   txtNode(5).Text = 1
'   txtNode(6).Text = 0
'   txtNode(7).Text = 0
'   'CurrentNode(CProj) = N
' Else
'   txtNode(2).Text = 0
'   txtNode(3).Text = 0
'   txtNode(4).Text = 1
'   txtNode(5).Text = 1
'   txtNode(6).Text = 0
'   txtNode(7).Text = 0
' End If
' End Select
'End If
Exit Sub
ErrorData2: MsgBox "Error : " & Format$(Error), 0 + 48, "Data Error"
          frmMain.txtNode(Index).SetFocus
          Exit Sub
End Sub

Sub txtNode_KeyPress (Index As Integer, KeyAscii As Integer)
If (KeyAscii = 13) And (Index < 7) Then
txtNode(Index + 1).SetFocus
End If
Exit Sub
End Sub

Sub txtNode_MouseMove (Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single)
Select Case Index
Case 1:
pnlStatus.Caption = " Column number must be in order, [Must be integer]"
Case 2:
pnlStatus.Caption = " Coordinate in X axis from left to right"
Case 3:
pnlStatus.Caption = " Coordinate in Y axis from top to bottom"
Case 4:
pnlStatus.Caption = " Array of column in X axis ,1 is not array"
Case 5:
pnlStatus.Caption = " Array of column in Y axis ,1 is not array"
Case 6:
pnlStatus.Caption = " Coordinate of oposite corner column in X axis
"
Case 7:
pnlStatus.Caption = " Coordinate of oposite corner column in Y axis
"
End Select
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRINT.BAS - 1

```

Sub txtProjectName_KeyPress (KeyAscii As Integer)
  If KeyAscii = 13 Then
    Input_Data 122, txtProjectName
  End If
End Sub

Sub txtProjectName_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
  pnlStatus.Caption = " Information data "
End Sub

Sub txtSL_KeyPress (KeyAscii As Integer)
  If KeyAscii = 13 Then
    SuperLoad(CProj) = CSng(txtSL.Text)
    'txtDummy.SetFocus
  End If
End Sub

Sub txtSL_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
  pnlStatus.Caption = " Super imposed dead load use in prestress design "
End Sub

Sub txtThick_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
  pnlStatus.Caption = " Slab Thickness [Output]"
End Sub

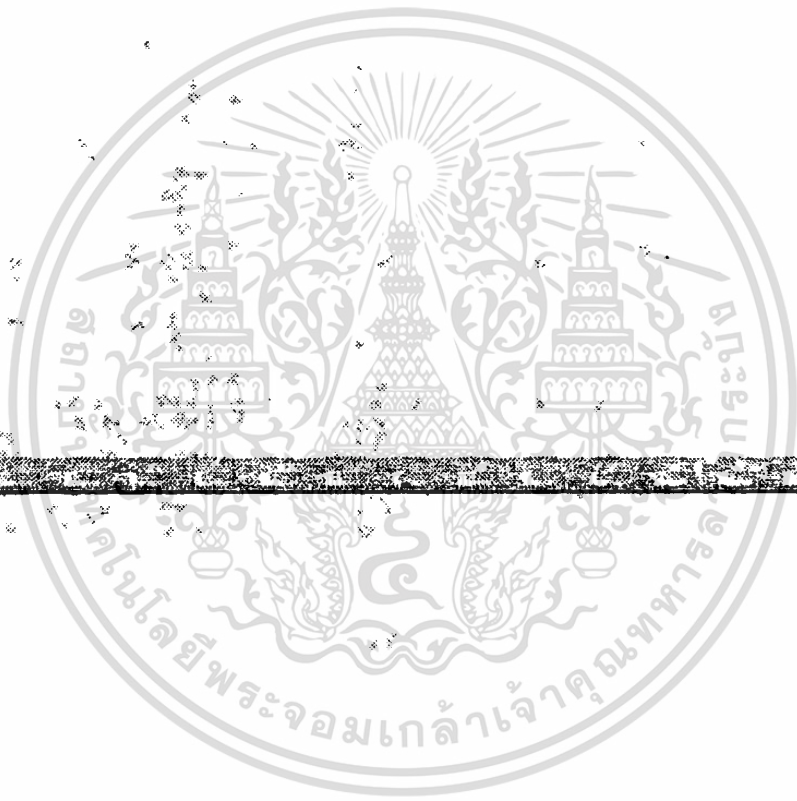
Sub txtTime_KeyPress (KeyAscii As Integer)
  If KeyAscii = 13 Then
    Input_Data 123, txtTime
  End If
End Sub

Sub txtTime_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
  pnlStatus.Caption = " Information data "
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LISTCOM.FRM - 1

VERSION 2.00

Begin Form frmListInfo

```

AutoRedraw      = -1 'True
Caption         = "Listing "
ClientHeight    = 3690
ClientLeft      = 1845
ClientTop       = 1800
ClientWidth     = 7770
FontBold        = 0 'False
FontItalic      = 0 'False
FontName        = "MS Sans Serif"
FontSize        = 8.25
FontStrikethru  = 0 'False
FontUnderline   = 0 'False
Height          = 4095
Icon            = (Icon)
Left            = 1785
LinkTopic       = "Form1"
MDIChild        = -1 'True
ScaleHeight     = 3690
ScaleWidth      = 7770
Top             = 1455
Width           = 7890
WindowState     = 2 'Maximized

```

Begin TextBox txtListCompute

```

BackColor       = &H00FFFFFF&
BorderStyle     = 0 'None
FontBold        = 0 'False
FontItalic      = 0 'False
FontName        = "Courier"
FontSize        = 9.75
FontStrikethru  = 0 'False
FontUnderline   = 0 'False
ForeColor       = &H00000000&
Height          = 6915
Index           = 18
Left            = 0
Multiline       = -1 'True
ScrollBars      = 3 'Both
TabIndex        = 17
TabStop         = 0 'False
Top             = 0
Width           = 9615

```

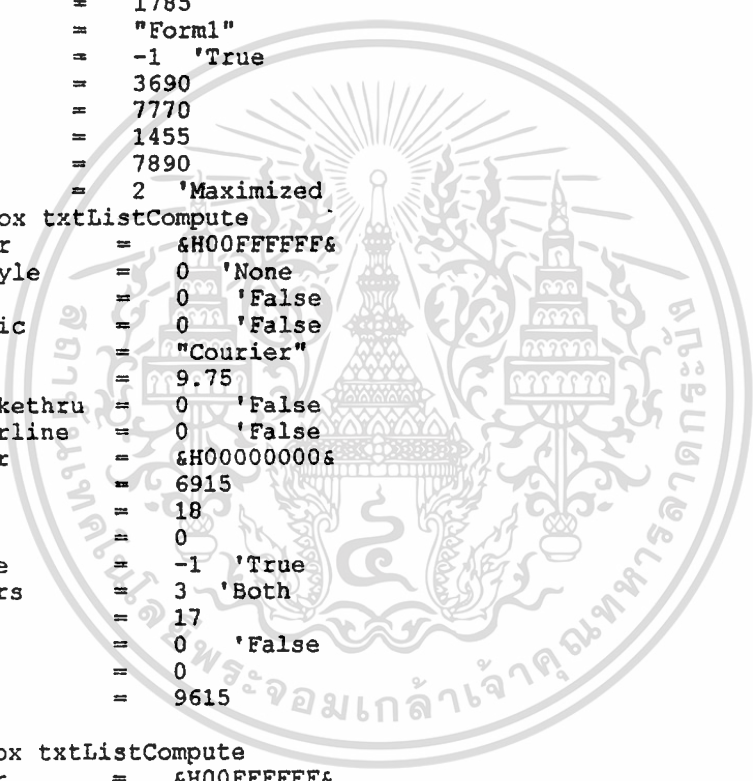
End

Begin TextBox txtListCompute

```

BackColor       = &H00FFFFFF&
BorderStyle     = 0 'None
FontBold        = 0 'False
FontItalic      = 0 'False
FontName        = "Arial"
FontSize        = 8.25
FontStrikethru  = 0 'False
FontUnderline   = 0 'False
ForeColor       = &H00000000&
Height          = 6915
Index           = 10
Left            = -60

```

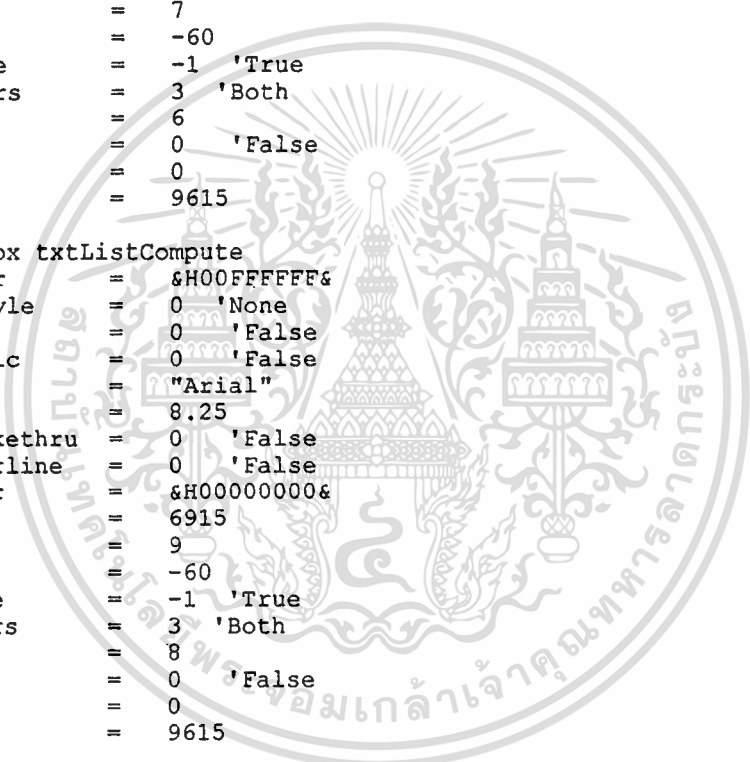


LISTCOM.FRM - 2

```

MultiLine      = -1 'True
ScrollBars     = 3  'Both
TabIndex       = 9
TabStop        = 0  'False
Top            = 0
Width          = 9615
End
Begin TextBox txtListCompute
BackColor      = &H00FFFFFF&
BorderStyle   = 0  'None
FontBold       = 0  'False
FontItalic     = 0  'False
FontName       = "Arial"
FontSize       = 8.25
FontStrikethru = 0  'False
FontUnderline  = 0  'False
ForeColor      = &H00000000&
Height        = 6915
Index         = 7
Left          = -60
MultiLine     = -1 'True
ScrollBars    = 3  'Both
TabIndex      = 6
TabStop       = 0  'False
Top           = 0
Width        = 9615
End
Begin TextBox txtListCompute
BackColor      = &H00FFFFFF&
BorderStyle   = 0  'None
FontBold       = 0  'False
FontItalic     = 0  'False
FontName       = "Arial"
FontSize       = 8.25
FontStrikethru = 0  'False
FontUnderline  = 0  'False
ForeColor      = &H00000000&
Height        = 6915
Index         = 9
Left          = -60
MultiLine     = -1 'True
ScrollBars    = 3  'Both
TabIndex      = 8
TabStop       = 0  'False
Top           = 0
Width        = 9615
End
Begin TextBox txtListCompute
BackColor      = &H00FFFFFF&
BorderStyle   = 0  'None
FontBold       = 0  'False
FontItalic     = 0  'False
FontName       = "Arial"
FontSize       = 8.25
FontStrikethru = 0  'False
FontUnderline  = 0  'False
ForeColor      = &H00000000&

```

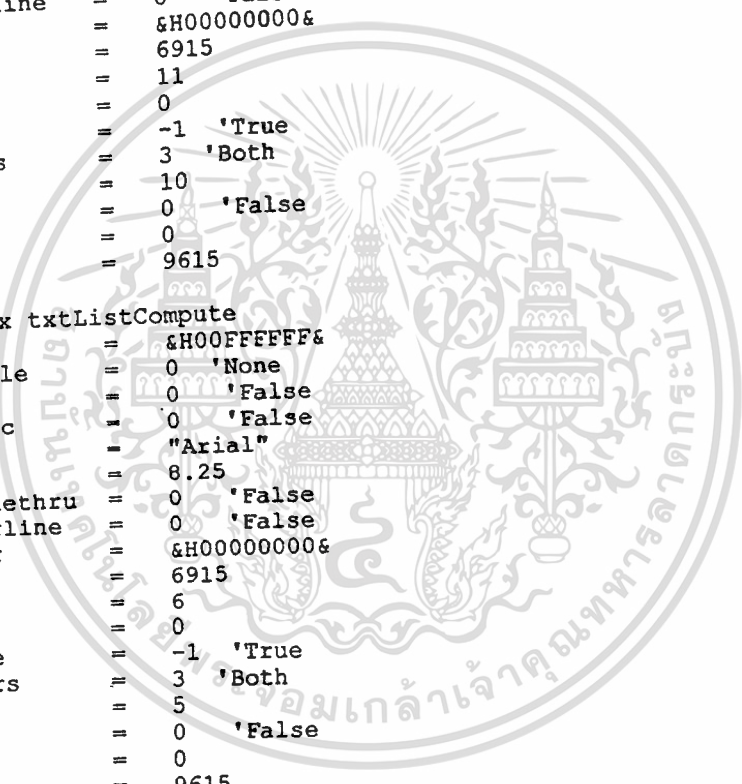


LISTCOM.FRM - 3

```

Height          = 6915
Index           = 12
Left            = 0
MultiLine       = -1 'True
ScrollBars      = 3  'Both
TabIndex        = 11
TabStop         = 0  'False
Top             = 0
Width           = 9615
End
Begin TextBox txtListCompute
BackColor       = &H00FFFFFF&
BorderStyle    = 0  'None
FontBold        = 0  'False
FontItalic      = 0  'False
FontName        = "Arial"
FontSize        = 8.25
FontStrikethru  = 0  'False
FontUnderline   = 0  'False
ForeColor       = &H00000000&
Height          = 6915
Index           = 11
Left            = 0
MultiLine       = -1 'True
ScrollBars      = 3  'Both
TabIndex        = 10
TabStop         = 0  'False
Top             = 0
Width           = 9615
End
Begin TextBox txtListCompute
BackColor       = &H00FFFFFF&
BorderStyle    = 0  'None
FontBold        = 0  'False
FontItalic      = 0  'False
FontName        = "Arial"
FontSize        = 8.25
FontStrikethru  = 0  'False
FontUnderline   = 0  'False
ForeColor       = &H00000000&
Height          = 6915
Index           = 6
Left            = 0
MultiLine       = -1 'True
ScrollBars      = 3  'Both
TabIndex        = 5
TabStop         = 0  'False
Top             = 0
Width           = 9615
End
Begin TextBox txtListCompute
BackColor       = &H00FFFFFF&
BorderStyle    = 0  'None
FontBold        = 0  'False
FontItalic      = 0  'False
FontName        = "Arial"
FontSize        = 8.25

```



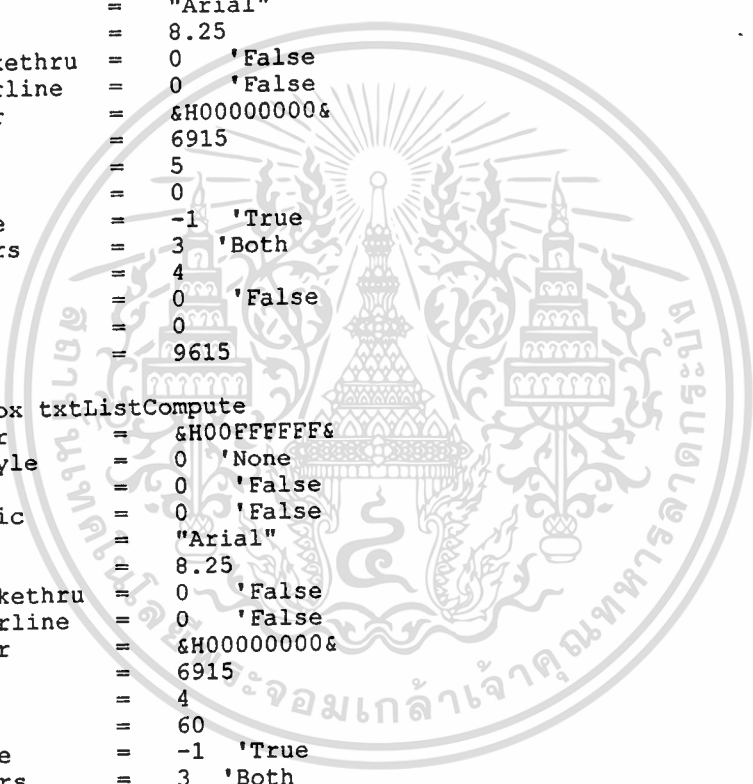
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LISTCOM.FRM - 4

```

FontStrikethru = 0 'False
FontUnderline  = 0 'False
ForeColor      = &H00000000&
Height        = 6915
Index         = 15
Left          = 0
MultiLine     = -1 'True
ScrollBars    = 3 'Both
TabIndex      = 14
TabStop       = 0 'False
Top           = 0
Width        = 9615
End
Begin TextBox.txtListCompute
BackColor     = &H00FFFFFF&
BorderStyle  = 0 'None
FontBold     = 0 'False
FontItalic   = 0 'False
FontName     = "Arial"
FontSize     = 8.25
FontStrikethru = 0 'False
FontUnderline = 0 'False
ForeColor    = &H00000000&
Height      = 6915
Index       = 5
Left        = 0
MultiLine   = -1 'True
ScrollBars  = 3 'Both
TabIndex    = 4
TabStop     = 0 'False
Top         = 0
Width      = 9615
End
Begin TextBox.txtListCompute
BackColor     = &H00FFFFFF&
BorderStyle  = 0 'None
FontBold     = 0 'False
FontItalic   = 0 'False
FontName     = "Arial"
FontSize     = 8.25
FontStrikethru = 0 'False
FontUnderline = 0 'False
ForeColor    = &H00000000&
Height      = 6915
Index       = 4
Left        = 60
MultiLine   = -1 'True
ScrollBars  = 3 'Both
TabIndex    = 3
TabStop     = 0 'False
Top         = 0
Width      = 9615
End
Begin TextBox.txtListCompute
BackColor     = &H00FFFFFF&
BorderStyle  = 0 'None
FontBold     = 0 'False

```



LISTCOM.FRM - 5

```

· FontItalic      = 0 'False
FontName         = "Arial"
FontSize        = 8.25
FontStrikethru  = 0 'False
FontUnderline   = 0 'False
ForeColor       = &H00000000&
Height          = 6915
Index           = 3
Left            = 0
MultiLine       = -1 'True
ScrollBars      = 3 'Both
TabIndex        = 2
TabStop         = 0 'False
Top             = -60
Width           = 9615

```

End

Begin TextBox txtListCompute

```

BackColor       = &H00FFFFFF&
BorderStyle     = 0 'None
FontBold        = 0 'False
FontItalic      = 0 'False
FontName        = "Arial"
FontSize        = 8.25
FontStrikethru  = 0 'False
FontUnderline   = 0 'False
ForeColor       = &H00000000&
Height          = 6915
Index           = 19
Left            = 0
MultiLine       = -1 'True
ScrollBars      = 3 'Both
TabIndex        = 18
TabStop         = 0 'False
Top             = -60
Width           = 9615

```

End

Begin TextBox txtListCompute

```

BackColor       = &H00FFFFFF&
BorderStyle     = 0 'None
FontBold        = 0 'False
FontItalic      = 0 'False
FontName        = "Arial"
FontSize        = 8.25
FontStrikethru  = 0 'False
FontUnderline   = 0 'False
ForeColor       = &H00000000&
Height          = 6915
Index           = 20
Left            = -60
MultiLine       = -1 'True
ScrollBars      = 3 'Both
TabIndex        = 19
TabStop         = 0 'False
Top             = 0
Width           = 9615

```

End

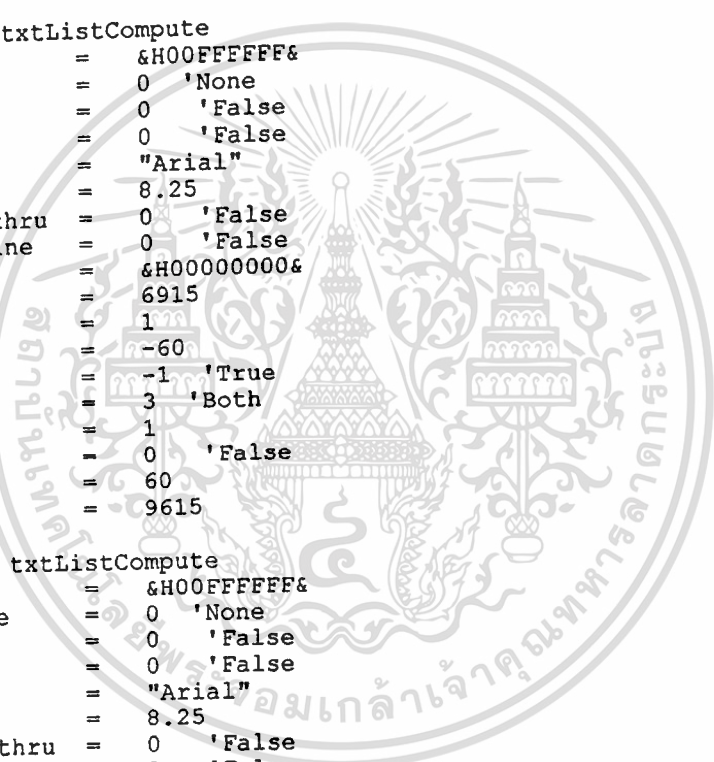
Begin TextBox txtListCompute

LISTCOM.FRM - 6

```

BackColor      = &H00FFFFFF&
BorderStyle    = 0 'None
FontBold       = 0 'False
FontItalic     = 0 'False
FontName       = "Arial"
FontSize       = 8.25
FontStrikethru = 0 'False
FontUnderline  = 0 'False
ForeColor      = &H00000000&
Height         = 6915
Index          = 17
Left           = 420
MultiLine     = -1 'True
ScrollBars     = 3 'Both
TabIndex       = 16
TabStop        = 0 'False
Top            = 0
Width          = 9615
End
Begin TextBox txtListCompute
BackColor      = &H00FFFFFF&
BorderStyle    = 0 'None
FontBold       = 0 'False
FontItalic     = 0 'False
FontName       = "Arial"
FontSize       = 8.25
FontStrikethru = 0 'False
FontUnderline  = 0 'False
ForeColor      = &H00000000&
Height         = 6915
Index          = 1
Left           = -60
MultiLine     = -1 'True
ScrollBars     = 3 'Both
TabIndex       = 1
TabStop        = 0 'False
Top            = 60
Width          = 9615
End
Begin TextBox txtListCompute
BackColor      = &H00FFFFFF&
BorderStyle    = 0 'None
FontBold       = 0 'False
FontItalic     = 0 'False
FontName       = "Arial"
FontSize       = 8.25
FontStrikethru = 0 'False
FontUnderline  = 0 'False
ForeColor      = &H00000000&
Height         = 6915
Index          = 2
Left           = -60
MultiLine     = -1 'True
ScrollBars     = 3 'Both
TabIndex       = 0
TabStop        = 0 'False
Top            = 0

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LISTCOM.FRM - 7

```

Width = 9615
End
Begin TextBox txtListCompute
BackColor = &H00FFFFFF&
BorderStyle = 0 'None
FontBold = 0 'False
FontItalic = 0 'False
FontName = "Arial"
FontSize = 8.25
FontStrikethru = 0 'False
FontUnderline = 0 'False
ForeColor = &H00000000&
Height = 6915
Index = 14
Left = 0
MultiLine = -1 'True
ScrollBars = 3 'Both
TabIndex = 13
TabStop = 0 'False
Top = 0
Width = 9615
End
Begin TextBox txtListCompute
BackColor = &H00FFFFFF&
BorderStyle = 0 'None
FontBold = 0 'False
FontItalic = 0 'False
FontName = "Arial"
FontSize = 8.25
FontStrikethru = 0 'False
FontUnderline = 0 'False
ForeColor = &H00000000&
Height = 6915
Index = 13
Left = 0
MultiLine = -1 'True
ScrollBars = 3 'Both
TabIndex = 12
TabStop = 0 'False
Top = 0
Width = 9615
End
Begin TextBox txtListCompute
BackColor = &H00FFFFFF&
BorderStyle = 0 'None
FontBold = 0 'False
FontItalic = 0 'False
FontName = "Arial"
FontSize = 8.25
FontStrikethru = 0 'False
FontUnderline = 0 'False
ForeColor = &H00000000&
Height = 6915
Index = 8
Left = -60
MultiLine = -1 'True
ScrollBars = 3 'Both

```

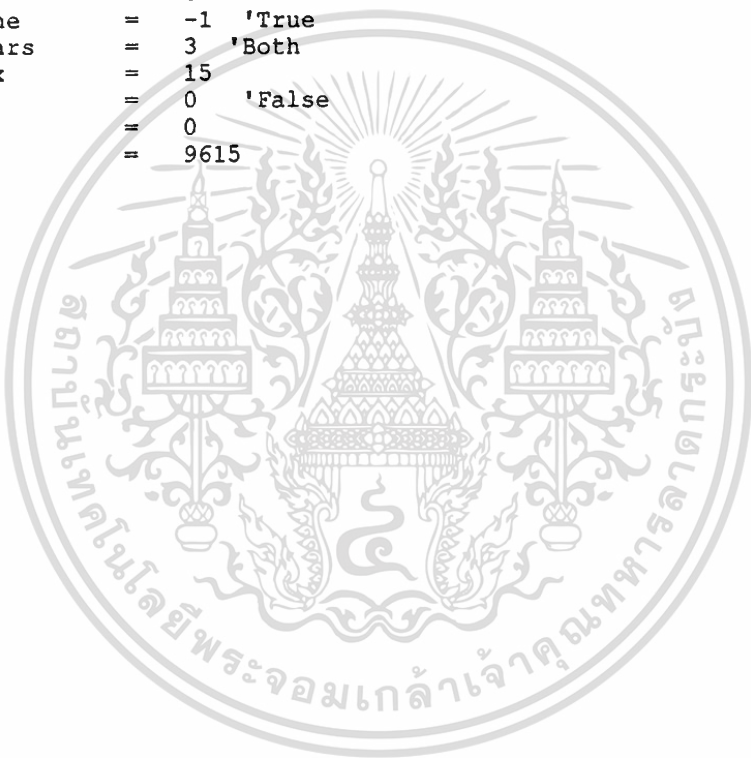
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LISTCOM.FRM - 8

```

    TabIndex      = 7
    TabStop       = 0 'False
    Top           = 0
    Width         = 9615
End
Begin TextBox txtListCompute
    BackColor     = &H00FFFFFF&
    BorderStyle   = 0 'None
    FontBold      = 0 'False
    FontItalic    = 0 'False
    FontName      = "Arial"
    FontSize      = 8.25
    FontStrikethru = 0 'False
    FontUnderline = 0 'False
    ForeColor     = &H00000000&
    Height        = 6915
    Index         = 16
    Left          = 0
    MultiLine     = -1 'True
    ScrollBars    = 3 'Both
    TabIndex      = 15
    TabStop       = 0 'False
    Top           = 0
    Width         = 9615
End
End

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LISTCOM.FRM - 1

```

Sub Form_Activate ()
frmMain.cmdPrevious.Visible = True
frmMain.cmdNext.Visible = True
If TextModule = 1 Then
    frmMain.cmdPrevious.Enabled = False
    frmMain.cmdNext.Enabled = False
End If
End Sub

Sub Form_Deactivate ()
frmMain.cmdPrevious.Visible = False
frmMain.cmdNext.Visible = False
End Sub

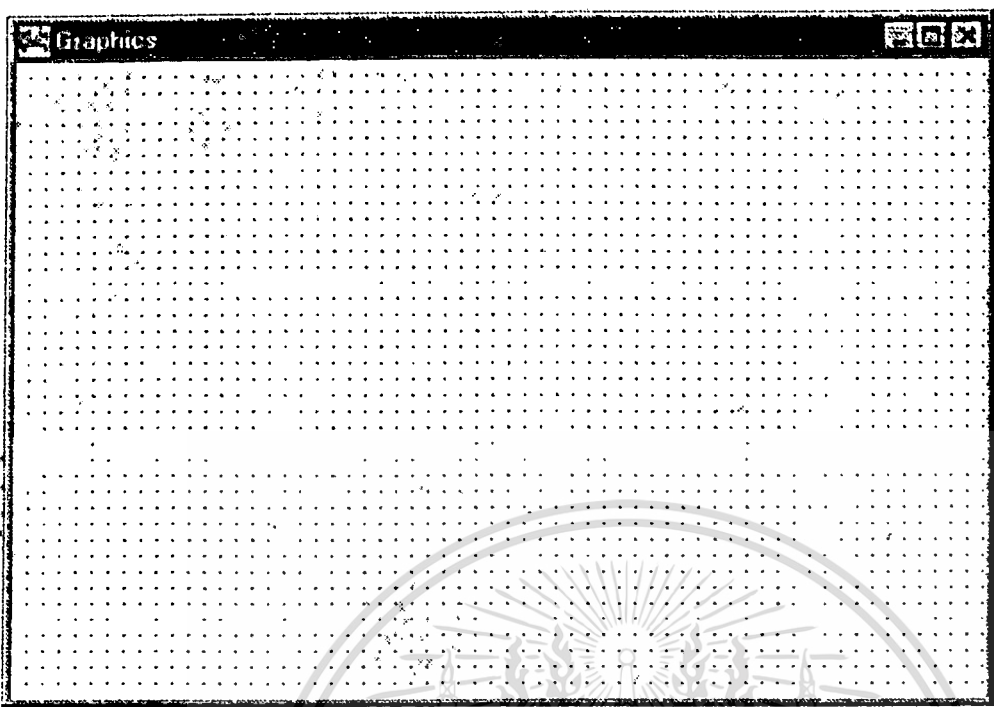
Sub Form_Load ()
For I = 1 To 20
    frmList(1).txtListCompute(I).FontName = "Courier"
    frmList(1).txtListCompute(I).FontSize = 6
Next I
End Sub

Sub Form_Resize ()
For I = 1 To 20
    frmList(CProj).txtListCompute(I).Width = frmList(CProj).ScaleWidth
    frmList(CProj).txtListCompute(I).Height = frmList(CProj).ScaleHeight
Next I
End Sub

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTGRAPH.FRM - 1

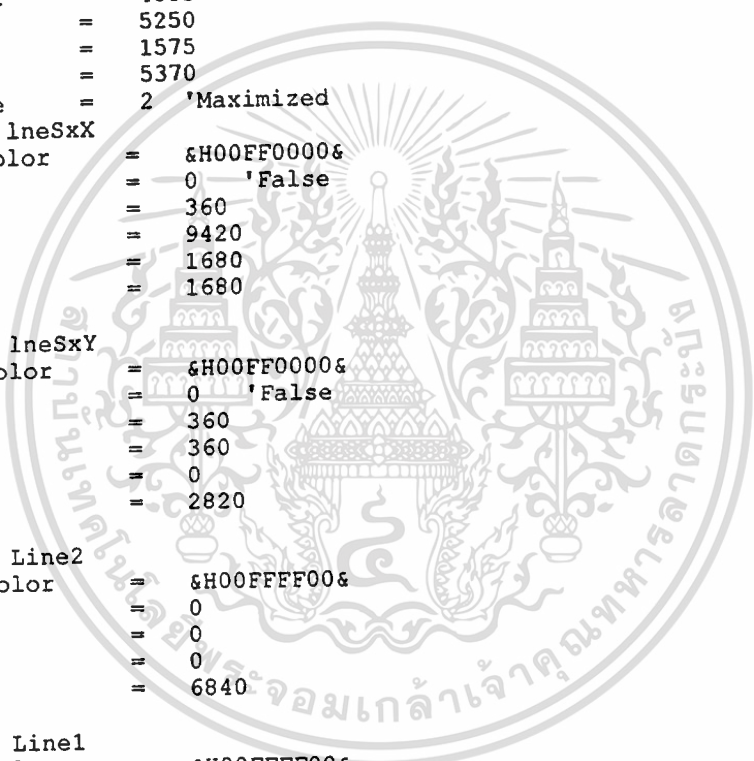
VERSION 2.00

Begin Form frmOutGraphInfo

```

AutoRedraw      = -1 'True
BackColor       = &H00E0FFFF&
Caption        = "Graphics"
ClientHeight    = 4005
ClientLeft     = 2475
ClientTop      = 1920
ClientWidth    = 5250
ForeColor      = &H00FF00FF&
Height         = 4410
Icon           = (Icon)
Left           = 2415
LinkTopic      = "Form1"
MDIChild       = -1 'True
ScaleHeight    = 4005
ScaleWidth     = 5250
Top            = 1575
Width          = 5370
WindowState    = 2 'Maximized
Begin Line lneSxX
  BorderColor   = &H00FF0000&
  Visible       = 0 'False
  X1            = 360
  X2            = 9420
  Y1            = 1680
  Y2            = 1680
End
Begin Line lneSxY
  BorderColor   = &H00FF0000&
  Visible       = 0 'False
  X1            = 360
  X2            = 360
  Y1            = 0
  Y2            = 2820
End
Begin Line Line2
  BorderColor   = &H00FFFF00&
  X1            = 0
  X2            = 0
  Y1            = 0
  Y2            = 6840
End
Begin Line Line1
  BorderColor   = &H00FFFF00&
  X1            = 0
  X2            = 9480
  Y1            = 0
  Y2            = 0
End
Begin Line lneMxX
  BorderColor   = &H00FF0000&
  Visible       = 0 'False
  X1            = 360
  X2            = 9420
  Y1            = 3900
  Y2            = 3900

```



OUTGRAPH.FRM - 2

```

End
Begin Line lneMxY
  BorderColor == &H00FF0000&
  Visible == 0 'False
  X1 == 360
  X2 == 360
  Y1 == 2760
  Y2 == 6420
End
End

```

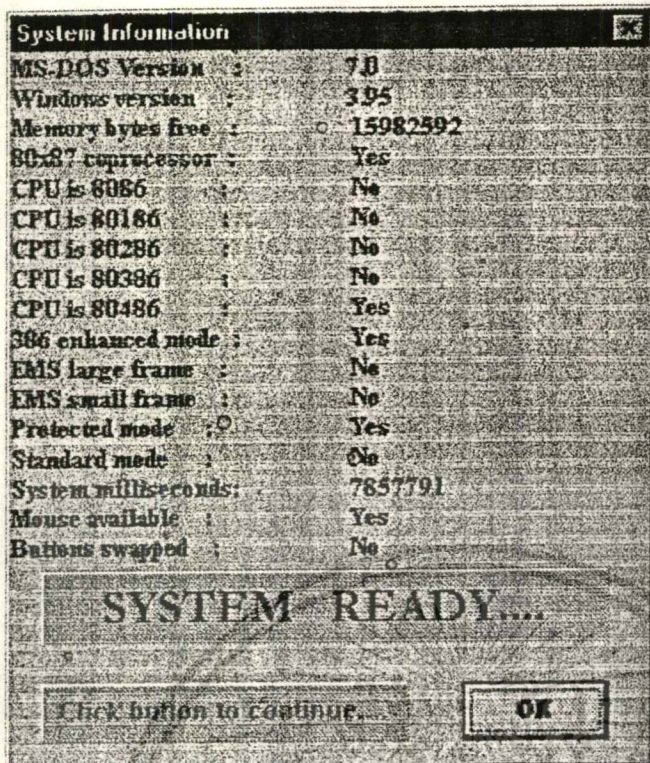


OUTGRAPH.FRM - 1

```
Sub Form_Activate ()  
  'For I = 0 To FileIndex  
  ' If frmMain.ActiveForm.Caption = frmData(I).Caption  
  '   Ready_Activate (I)  
  ' End If  
  'Next I  
End Sub
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SYSINFO.FRM - 1

VERSION 2.00

Begin Form frmSysInfo

```

AutoRedraw      = -1 'True
BackColor       = &H00FF80FF&
BorderStyle    = 3 'Fixed Double
Caption        = "System Information"
ClientHeight   = 5340
ClientLeft     = 1575
ClientTop      = 1485
ClientWidth    = 4800
FillColor     = &H0000FF00&
FillStyle      = 0 'Solid
FontBold       = -1 'True
FontItalic     = 0 'False
FontName       = "Times New Roman"
FontSize       = 9.75
FontStrikethru = 0 'False
FontUnderline  = 0 'False
ForeColor      = &H00FF0000&
Height         = 5745
Icon           = (Icon)
Left           = 1515
LinkTopic     = "Form1"
MaxButton     = 0 'False
MinButton     = 0 'False
Picture        = (Bitmap)
ScaleHeight   = 5340
ScaleWidth    = 4800
Top           = 1140
Width         = 4920

```

Begin SSPanel pnlMessage

```

Alignment      = 1 'Left Justify - MIDDLE
BackColor      = &H00FFFF00&
Caption        = "Click button to continue...."
FontBold       = -1 'True
FontItalic     = 0 'False
FontName       = "System"
FontSize       = 9.75
FontStrikethru = 0 'False
FontUnderline  = 0 'False
ForeColor      = &H000000FF&
Height         = 435
Left           = 240
TabIndex      = 0
Top            = 4740
Width         = 2775

```

End

Begin SSPanel pnlSysStatus

```

BackColor      = &H00FFFF00&
Caption        = "SYSTEM READY...."
FontBold       = -1 'True
FontItalic     = 0 'False
FontName       = "Garamond"
FontSize       = 18
FontStrikethru = 0 'False
FontUnderline  = 0 'False
ForeColor      = &H00FF0000&

```

SYSINFO.FRM - 2

```

Height      = 555
Left        = 240
TabIndex    = 1
Top         = 3900
Width       = 4275

```

End

Begin CommandButton cmdOK

```

Caption      = "OK"
Default      = -1 'True
FontBold     = -1 'True
FontItalic   = 0  'False
FontName     = "Terminal"
FontSize     = 9
FontStrikethru = 0  'False
FontUnderline = 0  'False
Height       = 435
Left         = 3420
TabIndex     = 2
Top          = 4740
Width        = 1095

```

End

End



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BYSINFO.FRM - 1

```

Sub cmdOK_Click ()
frmSysInfo.Visible = False
Unload frmSysInfo
'If SystemPass Then
' Load frmMain
' frmMain.Visible = True
'Else
' End
'End If
End Sub

Sub Form_Load ()
Check_System
'If SystemPass Then
' pnlSysStatus.Caption = "SYSTEM READY...."
'Else
' pnlSysStatus.Caption = "SYSTEM NOT READY...."
'End If
End Sub

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRESTRES.GBL - 1

```

''''Type''''
Type SpecialData
  FileName As String
  ProjectData As String * 100
  TimeData As String * 20
  DateData As String * 20
  AddressData As String * 100
  Engineer As String * 30
  InformData As String
End Type
Type GeneralType
  SlabOrBeam As Integer
  SlabIndex As Integer 'flat plat or flat slab
  CHeadIndex As Integer 'capital or panel or both or none
  LengthUnitIndex As Integer'length unit
  LoadUnitIndex As Integer 'load unit

  SFc_ As Single 'fc' of slab concrete
  SFs_ As Single 'fs' of strand steel
  RFs_ As Single 'fs' of reinforce steel
  CFc_ As Single 'fc' of column concrete
  SEc_ As Single 'E of slab concrete
  SEs_ As Single 'E of strand steel
  REs_ As Single 'E of reinforce steel
  CEc_ As Single 'E of column concrete
End Type
Type NodeType
  Co_X As Single
  Co_Y As Single
  End_XFlag As Integer
  End_YFlag As Integer
  SupportType As Integer
End Type
Type ColumnType
  Type As Integer
  Co_X As Single
  Co_Y As Single
  X_Flag As Integer
  Y_Flag As Integer
  c1 As Single
  c2 As Single
  DropX As Single
  DropY As Single
  DropThick As Single
  CapitalX As Single
  CapitalY As Single
  Length As Single
  SupportType As Integer
End Type
Type ElementType
  Start As Integer
  End As Integer
  CStart As Integer
  CEnd As Integer
  L1 As Single
  L2 As Single
  Length As Single

```



PRESTRES.GBL - 2

```

End Type
Type SectionType
  Type As Integer
  b As Single
  d As Single
End Type
Type LoadType
  Type As Integer
  ElementRef As Integer
  Start As Integer
  End As Integer
  StartLoad As Single
  EndLoad As Single
End Type
Type SLoadType
  Type As Integer
  ElementRef As Integer
  Start As Integer
  End As Integer
  'X1 As Single
  'Y1 As Single
  'X2 As Single
  'Y2 As Single
  StartLoad As Single
  EndLoad As Single
End Type
Type ResultType
  Line_X As Integer
  Line_Y As Integer
  StartMoment As Single
  EndMoment As Single
  StartShear As Single
  EndShear As Single
  EndReaction As Single
End Type
'''
''''DECLARE FUNCTION Call''''
Declare Function GetVersion Lib "Kernel" () As Long
Declare Function GetFreeSpace Lib "Kernel" (ByVal wFlags As Integer) As Long
Declare Function GetFreeSystemResources Lib "User" (ByVal fuSysResource As Integer) As Integer
Declare Function GetWinFlags Lib "Kernel" () As Long
Declare Function GetCurrentTime Lib "User" () As Long
Declare Function GetSingleClickTime Lib "User" () As Integer
Declare Function GetSystemMetrics Lib "User" (ByVal nIndex As Integer) As Integer

''''DEFINED VARIABLES''''

'General Graphic variable
Global Icon1(10, 3) As String * 12
Global Icon2(10, 3) As String * 12

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRESTRES.GBL - 3

```

Global ColumnColor As Double
'Global LineX As New frmOutGraphInfoLine1

'General Tool variable

Global SelectTool(5, 10) As Integer
Global InputTextIndex As Integer
Global InputTextIndexOpen As Integer

'Constant
Global Const MaxFileNum = 1
Global frmData(MaxFileNum) As New frmDataInfo
Global frmOutGraph(MaxFileNum) As New frmOutGraphInfo
Global frmOutText(MaxFileNum) As New frmOutTextInfo
Global frmList(MaxFileNum) As New frmListInfo

'System Variable
Global FFName As String
Global SaveTag As Integer
Global DirApp As String
Global FileIndex As Integer
Global FileNames(MaxFileNum) As String
Global Colors(47) As Long
Global SystemPass As Integer
Global Pass As Integer
Global DragEnter As Integer
Global IconClick1 As Integer
Global IconClick As Integer
Global TagChange(MaxFileNum) As Integer
Global CProj As Integer
Global Num As Integer
Global Const KEY_CR = &HD
Global Const KEY_LF = &HA
Global TextModule As Integer
Global CurrentPage As Integer
Global PrintCount As Long
Global TextToPrint As String
Global TextToScreen As String
Global TextNum As Integer
'General Data For Cal
Global SlabType As Integer
Global CalIndex As Integer
Global Const s1 = " "
Global Const s2 = " "
Global Const s3 = " "
Global Const s4 = " "
Global Const s5 = " "
Global Const s6 = " "
Global Const s7 = " "
Global Const s8 = " "
Global Const s9 = " "
Global Const s10 = " "
Global Const s15 = " "

''''Data In File *.PSD
Global SData(MaxFileNum) As SpecialData 'Data of project
Global Node(MaxFileNum, 1000) As NodeType'Data of specific Beam

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRESTRES.GBL - 4

```

Global Element(MaxFileNum, 1000) As ElementType'Data of specific Beam
Global LoadData(MaxFileNum, 1000) As LoadType'Data of specific Beam
Global Section(MaxFileNum, 100) As SectionType'Data of specific Beam
Global Result(MaxFileNum, 10000) As ResultType
Global ML(MaxFileNum, 100) As Single
Global MD(MaxFileNum, 100) As Single
Global MR(MaxFileNum, 100) As Single
Global RL(MaxFileNum, 100) As Single
Global RR(MaxFileNum, 100) As Single
Global COFL(MaxFileNum, 100) As Single
Global COFR(MaxFileNum, 100) As Single
Global FEMCL(MaxFileNum, 100) As Single
Global FEMCR(MaxFileNum, 100) As Single
Global GeneralData(MaxFileNum) As GeneralType
Global KfrictionLength(MaxFileNum) As Single
Global UfrictionAngle(MaxFileNum) As Single
Global NodeNum(MaxFileNum) As Integer 'Specific Node Number
Global LoadNum(MaxFileNum) As Integer 'Specific Load Number
Global ElementNum(MaxFileNum) As Integer 'Specific Element Number
Global SNodeNum(MaxFileNum) As Integer
Global SLoadNum(MaxFileNum) As Integer
Global SElementNum(MaxFileNum) As Integer
Global StrandIndex(MaxFileNum) As Integer
Global EElement(MaxFileNum, 10000) As ElementType
Global SElement(MaxFileNum, 1, 10000) As ElementType'General data of slab
Global EleColRow(MaxFileNum, 1, 100, 100) As ElementType
Global SNode(MaxFileNum, 10000) As NodeType'General data of slab
Global Column(MaxFileNum, 10000) As ColumnType
'Global Load(MaxFileNum,1000) As SLoadType
Global SLoad(MaxFileNum, 1, 100, 100) As LoadType
Global DeadLoad(MaxFileNum) As Single
Global LiveLoad(MaxFileNum) As Single
Global SuperLoad(MaxFileNum) As Single
Global SRow(MaxFileNum) As Integer
Global SCol(MaxFileNum) As Integer
Global RowEle(100) As Integer
Global ColEle(100) As Integer
Global ElementXNum(MaxFileNum) As Integer
Global ElementYNum(MaxFileNum) As Integer
Global Thick(MaxFileNum) As Single
Global d(MaxFileNum) As Single
Global Emax(MaxFileNum) As Single
Global DeadCo As Single
Global LiveCo As Single

```

```

'Try
Global ColNum(MaxFileNum) As Integer
Global RowNum(MaxFileNum) As Integer
Global CurrentNode(MaxFileNum) As Integer
Global CurrentLoad(MaxFileNum) As Integer
'Global SNodeNum(MaxFileNum) As Integer

```

'''DEFINED CONSTANTS WITH PROGRAM'''

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRESTRES.GBL - 5

```
Global Const SHADOW_COLOR = 8421504
Global Const BRIGHT_COLOR = 16777215
Global Const BEAM_INDEX = 0
Global Const SLAB_INDEX = 1
```

'/' DEFINED CONSTANTS WITH SYSTEM CODE''''

```
' Visual Basic global constant file. This file can be loaded
' into a code module.
'
' Some constants are commented out because they have
' duplicates (e.g., NONE appears several places).
'
' If you are updating a Visual Basic application written with
' an older version, you should replace your global constants
' with the constants in this file.
'
' .....
```

' General

' Clipboard formats

```
Global Const CF_LINK = &HBF00
Global Const CF_TEXT = 1
Global Const CF_BITMAP = 2
Global Const CF_METAFILE = 3
Global Const CF_DIB = 8
Global Const CF_PALETTE = 9
```

' DragOver

```
Global Const ENTER = 0
Global Const LEAVE = 1
Global Const OVER = 2
```

' Drag (controls)

```
Global Const CANCEL = 0
Global Const BEGIN_DRAG = 1
Global Const END_DRAG = 2
```

' Show parameters

```
Global Const MODAL = 1
Global Const MODELESS = 0
```

' Arrange Method

' for MDI Forms

```
Global Const CASCADE = 0
Global Const TILE_HORIZONTAL = 1
Global Const TILE_VERTICAL = 2
Global Const ARRANGE_ICONS = 3
```

' ZOrder Method

```
Global Const BRINGTOFRONT = 0
Global Const SENDTOBACK = 1
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRESTRES.GBL - 6

' Key Codes

```

Global Const KEY_LBUTTON = &H1
Global Const KEY_RBUTTON = &H2
Global Const KEY_CANCEL = &H3
Global Const KEY_MBUTTON = &H4      ' NOT contiguous with L & RBUTTON
Global Const KEY_BACK = &H8
Global Const KEY_TAB = &H9
Global Const KEY_CLEAR = &HC
Global Const KEY_RETURN = &HD
Global Const KEY_SHIFT = &H10
Global Const KEY_CONTROL = &H11
Global Const KEY_MENU = &H12
Global Const KEY_PAUSE = &H13
Global Const KEY_CAPITAL = &H14
Global Const KEY_ESCAPE = &H1B
Global Const KEY_SPACE = &H20
Global Const KEY_PRIOR = &H21
Global Const KEY_NEXT = &H22
Global Const KEY_END = &H23
Global Const KEY_HOME = &H24
Global Const KEY_LEFT = &H25
Global Const KEY_UP = &H26
Global Const KEY_RIGHT = &H27
Global Const KEY_DOWN = &H28
Global Const KEY_SELECT = &H29
Global Const KEY_PRINT = &H2A
Global Const KEY_EXECUTE = &H2B
Global Const KEY_SNAPSHOT = &H2C
Global Const KEY_INSERT = &H2D
Global Const KEY_DELETE = &H2E
Global Const KEY_HELP = &H2F

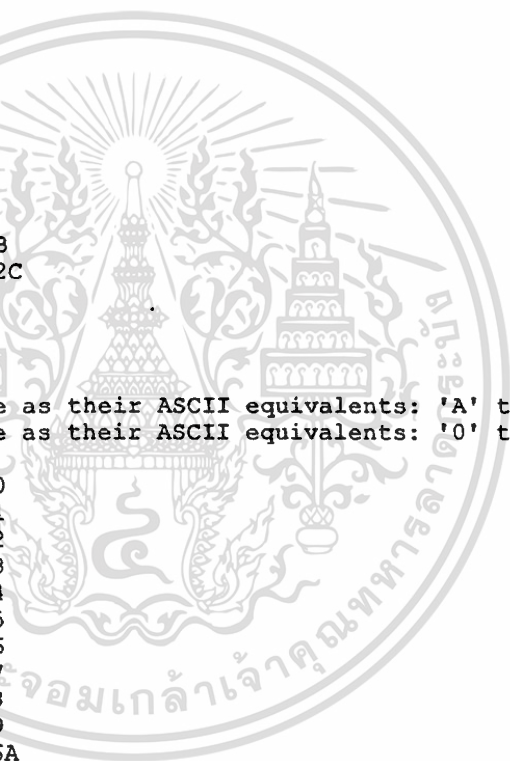
```

' KEY_A thru KEY_Z are the same as their ASCII equivalents: 'A' thru 'Z'
 ' KEY_0 thru KEY_9 are the same as their ASCII equivalents: '0' thru '9'

```

Global Const KEY_NUMPAD0 = &H60
Global Const KEY_NUMPAD1 = &H61
Global Const KEY_NUMPAD2 = &H62
Global Const KEY_NUMPAD3 = &H63
Global Const KEY_NUMPAD4 = &H64
Global Const KEY_NUMPAD5 = &H65
Global Const KEY_NUMPAD6 = &H66
Global Const KEY_NUMPAD7 = &H67
Global Const KEY_NUMPAD8 = &H68
Global Const KEY_NUMPAD9 = &H69
Global Const KEY_MULTIPLY = &H6A
Global Const KEY_ADD = &H6B
Global Const KEY_SEPARATOR = &H6C
Global Const KEY_SUBTRACT = &H6D
Global Const KEY_DECIMAL = &H6E
Global Const KEY_DIVIDE = &H6F
Global Const KEY_F1 = &H70
Global Const KEY_F2 = &H71
Global Const KEY_F3 = &H72
Global Const KEY_F4 = &H73
Global Const KEY_F5 = &H74

```



PRESTRES.GBL - 7

Global Const KEY_F6 = &H75
 Global Const KEY_F7 = &H76
 Global Const KEY_F8 = &H77
 Global Const KEY_F9 = &H78
 Global Const KEY_F10 = &H79
 Global Const KEY_F11 = &H7A
 Global Const KEY_F12 = &H7B
 Global Const KEY_F13 = &H7C
 Global Const KEY_F14 = &H7D
 Global Const KEY_F15 = &H7E
 Global Const KEY_F16 = &H7F

Global Const KEY_NUMLOCK = &H90

' Variant VarType tags

Global Const V_EMPTY = 0
 Global Const V_NULL = 1
 Global Const V_INTEGER = 2
 Global Const V_LONG = 3
 Global Const V_Single = 4
 Global Const V_DOUBLE = 5
 Global Const V_CURRENCY = 6
 Global Const V_DATE = 7
 Global Const V_STRING = 8

' Event Parameters

' ErrNum (LinkError)

Global Const WRONG_FORMAT = 1
 Global Const DDE_SOURCE_CLOSED = 6
 Global Const TOO_MANY_LINKS = 7
 Global Const DATA_TRANSFER_FAILED = 8

' QueryUnload

Global Const FORM_CONTROLMENU = 0
 Global Const FORM_CODE = 1
 Global Const APP_WINDOWS = 2
 Global Const APP_TASKMANAGER = 3
 Global Const FORM_MDIFORM = 4

' Properties

' Colors

Global Const BLACK = &H0&
 Global Const RED = &HFF&
 Global Const GREEN = &HFF00&
 Global Const YELLOW = &HFFFF&
 Global Const BLUE = &HFF0000
 Global Const MAGENTA = &HFF00FF
 Global Const CYAN = &HFFFF00
 Global Const WHITE = &HFFFFFF

' System Colors

Global Const SCROLL_BARS = &H80000000 ' Scroll-bars gray area.
 Global Const DESKTOP = &H80000001 ' Desktop.



PRESTRES.GBL - 8

```

Global Const ACTIVE_TITLE_BAR = &H80000002 ' Active window caption.
Global Const INACTIVE_TITLE_BAR = &H80000003 ' Inactive window caption.
Global Const MENU_BAR = &H80000004 ' Menu background.
Global Const WINDOW_BACKGROUND = &H80000005 ' Window background.
Global Const WINDOW_FRAME = &H80000006 ' Window frame.
Global Const MENU_TEXT = &H80000007 ' Text in menus.
Global Const WINDOW_TEXT = &H80000008 ' Text in windows.
Global Const TITLE_BAR_TEXT = &H80000009 ' Text in caption, size bo
x, scroll-bar arrow box..
Global Const ACTIVE_BORDER = &H8000000A ' Active window border.
Global Const INACTIVE_BORDER = &H8000000B ' Inactive window border.
Global Const APPLICATION_WORKSPACE = &H8000000C ' Background color of mult
iple document interface (MDI) applications.
Global Const HIGHLIGHT = &H8000000D ' Items selected item in a
control.
Global Const HIGHLIGHT_TEXT = &H8000000E ' Text of item selected in
a control.
Global Const BUTTON_FACE = &H8000000F ' Face shading on command
buttons.
Global Const BUTTON_SHADOW = &H80000010 ' Edge shading on command
buttons.
Global Const GRAY_TEXT = &H80000011 ' Grayed (disabled) text.
This color is set to 0 if the current display driver does not support a s
olid gray color.
Global Const BUTTON_TEXT = &H80000012 ' Text on push buttons.

' Enumerated Types

' Align (picture box)
Global Const NONE = 0
Global Const ALIGN_TOP = 1
Global Const ALIGN_BOTTOM = 2

' Alignment
Global Const LEFT_JUSTIFY = 0 ' 0 - Left Justify
Global Const RIGHT_JUSTIFY = 1 ' 1 - Right Justify
Global Const CENTER = 2 ' 2 - Center

' BorderStyle (form)
Global Const NONE = 0 ' 0 - None
Global Const FIXED_SINGLE = 1 ' 1 - Fixed Single
Global Const SIZABLE = 2 ' 2 - Sizable (Forms only)
Global Const FIXED_DOUBLE = 3 ' 3 - Fixed Double (Forms only)

' BorderStyle (Shape and Line)
Global Const TRANSPARENT = 0 ' 0 - Transparent
Global Const SOLID = 1 ' 1 - Solid
Global Const DASH = 2 ' 2 - Dash
Global Const DOT = 3 ' 3 - Dot
Global Const DASH_DOT = 4 ' 4 - Dash-Dot
Global Const DASH_DOT_DOT = 5 ' 5 - Dash-Dot-Dot
Global Const INSIDE_SOLID = 6 ' 6 - Inside Solid

' MousePointer
Global Const DEFAULT = 0 ' 0 - Default
Global Const ARROW = 1 ' 1 - Arrow
Global Const CROSSHAIR = 2 ' 2 - Cross

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRESTRES.GBL - 9

```
Global Const IBEAM = 3          ' 3 - I-Beam
Global Const ICON_POINTER = 4   ' 4 - Icon
Global Const SIZE_POINTER = 5   ' 5 - Size
Global Const SIZE_NE_SW = 6     ' 6 - Size NE SW
Global Const SIZE_N_S = 7       ' 7 - Size N S
Global Const SIZE_NW_SE = 8     ' 8 - Size NW SE
Global Const SIZE_W_E = 9       ' 9 - Size W E
Global Const UP_ARROW = 10      ' 10 - Up Arrow
Global Const HOURGLASS = 11     ' 11 - Hourglass
Global Const NO_DROP = 12       ' 12 - No drop
```

' DragMode

```
Global Const MANUAL = 0        ' 0 - Manual
Global Const AUTOMATIC = 1     ' 1 - Automatic
```

' DrawMode

```
Global Const BLACKNESS = 1     ' 1 - Blackness
Global Const NOT_MERGE_PEN = 2 ' 2 - Not Merge Pen
Global Const MASK_NOT_PEN = 3  ' 3 - Mask Not Pen
Global Const NOT_COPY_PEN = 4  ' 4 - Not Copy Pen
Global Const MASK_PEN_NOT = 5  ' 5 - Mask Pen Not
Global Const INVERT = 6        ' 6 - Invert
Global Const XOR_PEN = 7        ' 7 - Xor Pen
Global Const NOT_MASK_PEN = 8  ' 8 - Not Mask Pen
Global Const MASK_PEN = 9      ' 9 - Mask Pen
Global Const NOT_XOR_PEN = 10  ' 10 - Not Xor Pen
Global Const NOP = 11          ' 11 - Nop
Global Const MERGE_NOT_PEN = 12 ' 12 - Merge Not Pen
Global Const COPY_PEN = 13     ' 13 - Copy Pen
Global Const MERGE_PEN_NOT = 14 ' 14 - Merge Pen Not
Global Const MERGE_PEN = 15    ' 15 - Merge Pen
Global Const WHITENESS = 16    ' 16 - Whiteness
```

' DrawStyle

```
Global Const SOLID = 0         ' 0 - Solid
Global Const DASH = 1         ' 1 - Dash
Global Const DOT = 2          ' 2 - Dot
Global Const DASH_DOT = 3     ' 3 - Dash-Dot
Global Const DASH_DOT_DOT = 4 ' 4 - Dash-Dot-Dot
Global Const INVISIBLE = 5    ' 5 - Invisible
Global Const INSIDE_SOLID = 6 ' 6 - Inside Solid
```

' FillStyle

```
Global Const SOLID = 0         ' 0 - Solid
Global Const TRANSPARENT = 1   ' 1 - Transparent
Global Const HORIZONTAL_LINE = 2 ' 2 - Horizontal Line
Global Const VERTICAL_LINE = 3 ' 3 - Vertical Line
Global Const UPWARD_DIAGONAL = 4 ' 4 - Upward Diagonal
Global Const DOWNWARD_DIAGONAL = 5 ' 5 - Downward Diagonal
Global Const CROSS = 6         ' 6 - Cross
Global Const DIAGONAL_CROSS = 7 ' 7 - Diagonal Cross
```

' LinkMode (forms and controls)

```
Global Const NONE = 0         ' 0 - None
Global Const LINK_SOURCE = 1  ' 1 - Source (forms only)
Global Const LINK_AUTOMATIC = 1 ' 1 - Automatic (controls only)
Global Const LINK_MANUAL = 2  ' 2 - Manual (controls only)
```

PRESTRES.GBL - 10

Global Const LINK_NOTIFY = 3 ' 3 - Notify (controls only)

' LinkMode (kept for VB1.0 compatibility, use new constants instead)

Global Const HOT = 1 ' 1 - Hot (controls only)

Global Const SERVER = 1 ' 1 - Server (forms only)

Global Const COLD = 2 ' 2 - Cold (controls only)

' ScaleMode

Global Const USER = 0 ' 0 - User

Global Const TWIPS = 1 ' 1 - Twip

Global Const POINTS = 2 ' 2 - Point

Global Const PIXELS = 3 ' 3 - Pixel

Global Const CHARACTERS = 4 ' 4 - Character

Global Const INCHES = 5 ' 5 - Inch

Global Const MILLIMETERS = 6 ' 6 - Millimeter

Global Const CENTIMETERS = 7 ' 7 - Centimeter

' ScrollBar

' Global Const NONE = 0 ' 0 - None

Global Const HORIZONTAL = 1 ' 1 - Horizontal

Global Const VERTICAL = 2 ' 2 - Vertical

Global Const BOTH = 3 ' 3 - Both

' Shape

Global Const SHAPE_RECTANGLE = 0

Global Const SHAPE_SQUARE = 1

Global Const SHAPE_OVAL = 2

Global Const SHAPE_CIRCLE = 3

Global Const SHAPE_ROUNDED_RECTANGLE = 4

Global Const SHAPE_ROUNDED_SQUARE = 5

' WindowState

Global Const NORMAL = 0 ' 0 - Normal

Global Const MINIMIZED = 1 ' 1 - Minimized

Global Const MAXIMIZED = 2 ' 2 - Maximized

' Check Value

Global Const UNCHECKED = 0 ' 0 - Unchecked

Global Const CHECKED = 1 ' 1 - Checked

Global Const GRAYED = 2 ' 2 - Grayed

' Shift parameter masks

Global Const SHIFT_MASK = 1

Global Const CTRL_MASK = 2

Global Const ALT_MASK = 4

' Button parameter masks

Global Const LEFT_BUTTON = 1

Global Const RIGHT_BUTTON = 2

Global Const MIDDLE_BUTTON = 4

' Function Parameters

' MsgBox parameters

Global Const MB_OK = 0

' OK button only

Global Const MB_OKCANCEL = 1

' OK and Cancel buttons

Global Const MB_ABORTRETRYIGNORE = 2

' Abort, Retry, and Ignore buttons

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRESTRES.GBL - 11

```

Global Const MB_YESNOCANCEL = 3      ' Yes, No, and Cancel buttons
Global Const MB_YESNO = 4           ' Yes and No buttons
Global Const MB_RETRYCANCEL = 5     ' Retry and Cancel buttons

Global Const MB_ICONSTOP = 16      ' Critical message
Global Const MB_ICONQUESTION = 32  ' Warning query
Global Const MB_ICONEXCLAMATION = 48 ' Warning message
Global Const MB_ICONINFORMATION = 64 ' Information message

Global Const MB_APPLMODAL = 0      ' Application Modal Message Box
Global Const MB_DEFBUTTON1 = 0     ' First button is default
Global Const MB_DEFBUTTON2 = 256   ' Second button is default
Global Const MB_DEFBUTTON3 = 512   ' Third button is default
Global Const MB_SYSTEMMODAL = 4096 ' System Modal

' MsgBox return values
Global Const IDOK = 1              ' OK button pressed
Global Const IDCANCEL = 2         ' Cancel button pressed
Global Const IDABORT = 3          ' Abort button pressed
Global Const IDRETRY = 4          ' Retry button pressed
Global Const IDIGNORE = 5         ' Ignore button pressed
Global Const IDYES = 6            ' Yes button pressed
Global Const IDNO = 7             ' No button pressed

' SetAttr, Dir, GetAttr functions
Global Const ATTR_NORMAL = 0
Global Const ATTR_READONLY = 1
Global Const ATTR_HIDDEN = 2
Global Const ATTR_SYSTEM = 4
Global Const ATTR_VOLUME = 8
Global Const ATTR_DIRECTORY = 16
Global Const ATTR_ARCHIVE = 32

'Grid
'ColAlignment,FixedAlignment Properties
Global Const GRID_ALIGNLEFT = 0
Global Const GRID_ALIGNRIGHT = 1
Global Const GRID_ALIGNCENTER = 2

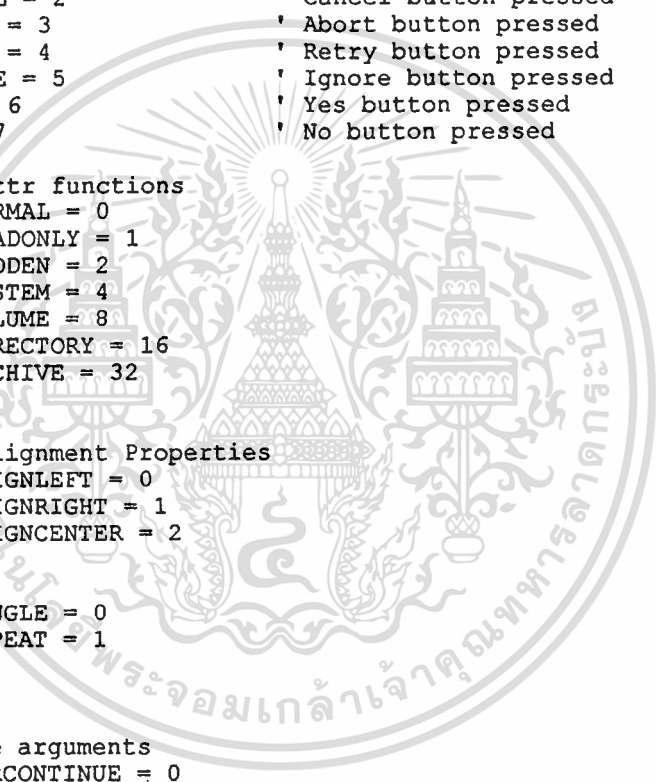
'Fillstyle Property
Global Const GRID_SINGLE = 0
Global Const GRID_REPEAT = 1

'Data control
'Error event Response arguments
Global Const DATA_ERRCONTINUE = 0
Global Const DATA_ERRDISPLAY = 1

'Editmode property values
Global Const DATA_EDITNONE = 0
Global Const DATA_EDITMODE = 1
Global Const DATA_EDITADD = 2

' Options property values
Global Const DATA_DENYWRITE = &H1
Global Const DATA_DENYREAD = &H2

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRESTRES.GBL - 12

Global Const DATA_READONLY = &H4
 Global Const DATA_APPENDONLY = &H8
 Global Const DATA_INCONSISTENT = &H10
 Global Const DATA_CONSISTENT = &H20
 Global Const DATA_SQLPASSTHROUGH = &H40

'Validate event Action arguments

Global Const DATA_ACTIONCANCEL = 0
 Global Const DATA_ACTIONMOVEFIRST = 1
 Global Const DATA_ACTIONMOVEPREVIOUS = 2
 Global Const DATA_ACTIONMOVENEXT = 3
 Global Const DATA_ACTIONMOVELAST = 4
 Global Const DATA_ACTIONADDNEW = 5
 Global Const DATA_ACTIONUPDATE = 6
 Global Const DATA_ACTIONDELETE = 7
 Global Const DATA_ACTIONFIND = 8
 Global Const DATA_ACTIONBOOKMARK = 9
 Global Const DATA_ACTIONCLOSE = 10
 Global Const DATA_ACTIONUNLOAD = 11

'OLE Client Control

'Actions

Global Const OLE_CREATE_EMBED = 0
 Global Const OLE_CREATE_NEW = 0 'from ole1 control
 Global Const OLE_CREATE_LINK = 1
 Global Const OLE_CREATE_FROM_FILE = 1 'from ole1 control
 Global Const OLE_COPY = 4
 Global Const OLE_PASTE = 5
 Global Const OLE_UPDATE = 6
 Global Const OLE_ACTIVATE = 7
 Global Const OLE_CLOSE = 9
 Global Const OLE_DELETE = 10
 Global Const OLE_SAVE_TO_FILE = 11
 Global Const OLE_READ_FROM_FILE = 12
 Global Const OLE_INSERT_OBJ_DLG = 14
 Global Const OLE_PASTE_SPECIAL_DLG = 15
 Global Const OLE_FETCH_VERBS = 17
 Global Const OLE_SAVE_TO_OLE1FILE = 18

'OLEType

Global Const OLE_LINKED = 0
 Global Const OLE_EMBEDDED = 1
 Global Const OLE_NONE = 3

'OLETypeAllowed

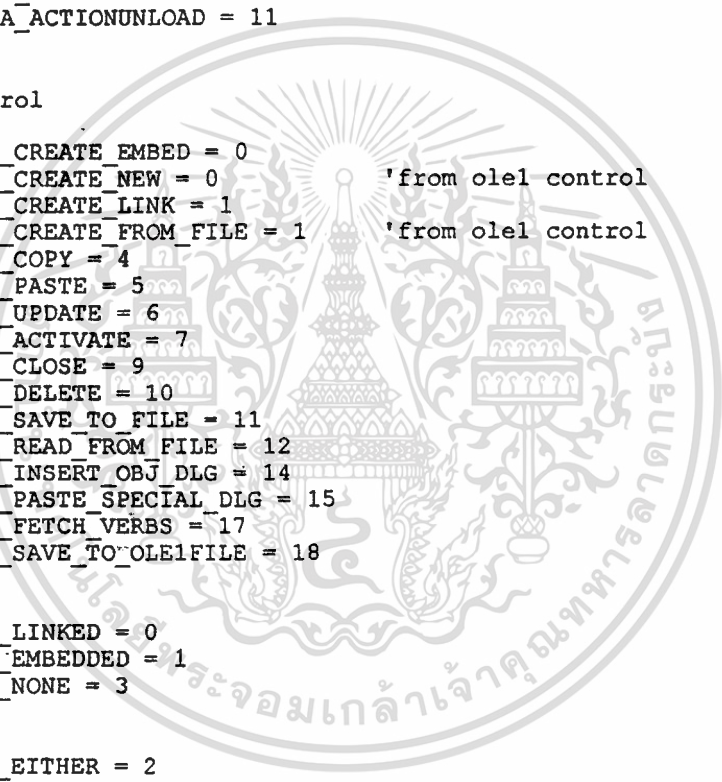
Global Const OLE_EITHER = 2

'UpdateOptions

Global Const OLE_AUTOMATIC = 0
 Global Const OLE_FROZEN = 1
 Global Const OLE_MANUAL = 2

'AutoActivate modes

'Note that OLE_ACTIVATE_GETFOCUS only applies to objects that
 'support "inside-out" activation. See related Verb notes below.
 Global Const OLE_ACTIVATE_MANUAL = 0



PRESTRES.GBL - 13

Global Const OLE_ACTIVATE_GETFOCUS = 1
Global Const OLE_ACTIVATE_SingleCLICK = 2

'SizeModes

Global Const OLE_SIZE_CLIP = 0
Global Const OLE_SIZE_STRETCH = 1
Global Const OLE_SIZE_AUTOSIZE = 2

'DisplayTypes

Global Const OLE_DISPLAY_CONTENT = 0
Global Const OLE_DISPLAY_ICON = 1

'Update Event Constants

Global Const OLE_CHANGED = 0
Global Const OLE_SAVED = 1
Global Const OLE_CLOSED = 2
Global Const OLE_RENAMED = 3

'Special Verb Values

Global Const VERB_PRIMARY = 0
Global Const VERB_SHOW = -1
Global Const VERB_OPEN = -2
Global Const VERB_HIDE = -3
Global Const VERB_INPLACEUIACTIVATE = -4
Global Const VERB_INPLACEACTIVATE = -5

'The last two verbs are for objects that support "inside-out" activation, meaning they can be edited in-place, and that they support being left in-place-active even when the input focus moves to another control or form.

'These objects actually have 2 levels of being active. "InPlace Active" means that the object is ready for the user to click inside it and start working with it. "In-Place UI-Active" means that, in addition, if the object has any other UI associated with it, such as floating palette windows, that those windows are visible and ready for use. Any number of objects can be "In-Place Active" at a time, although only one can be "InPlace UI-Active".

'You can cause an object to move to either one of states programmatically by setting the Verb property to the appropriate verb and setting Action=OLE_ACTIVATE.

'Also, if you set AutoActivate = OLE_ACTIVATE_GETFOCUS, the server will automatically be put into "InPlace UI-Active" state when the user clicks on or tabs into the control.

'VerbFlag Bit Masks

Global Const VERBFLAG_GRAYED = &H1
Global Const VERBFLAG_DISABLED = &H2
Global Const VERBFLAG_CHECKED = &H8
Global Const VERBFLAG_SEPARATOR = &H800

'MiscFlag Bits - Or these together as desired for special behaviors

'MEMSTORAGE causes the control to use memory to store the object while it is loaded. This is faster than the default (disk-tempfile)

PRESTRES.GBL - 14

```

;
; but can consume a lot of memory for objects whose data takes
; up a lot of space, such as the bitmap for a paint program.
Global Const OLE_MISCFLAG_MEMSTORAGE = &H1

'DISABLEINPLACE overrides the control's default behavior of allowing
; in-place activation for objects that support it. If you
; are having problems activating an object inplace, you can
; force it to always activate in a separate window by setting th
; is
; bit
Global Const OLE_MISCFLAG_DISABLEINPLACE = &H2

'Common Dialog Control
'Action Property
Global Const DLG_FILE_OPEN = 1
Global Const DLG_FILE_SAVE = 2
Global Const DLG_COLOR = 3
Global Const DLG_FONT = 4
Global Const DLG_PRINT = 5
Global Const DLG_HELP = 6

'File Open/Save Dialog Flags
Global Const OFN_READONLY = &H1&
Global Const OFN_OVERWRITEPROMPT = &H2&
Global Const OFN_HIDEREADONLY = &H4&
Global Const OFN_NOCHANGEDIR = &H8&
Global Const OFN_SHOWHELP = &H10&
Global Const OFN_NOVALIDATE = &H100&
Global Const OFN_ALLOWMULTISELECT = &H200&
Global Const OFN_EXTENSIONDIFFERENT = &H400&
Global Const OFN_PATHMUSTEXIST = &H800&
Global Const OFN_FILEMUSTEXIST = &H1000&
Global Const OFN_CREATEPROMPT = &H2000&
Global Const OFN_SHAREAWARE = &H4000&
Global Const OFN_NOREADONLYRETURN = &H8000&

'Color Dialog Flags
Global Const CC_RGBINIT = &H1&
Global Const CC_FULLOPEN = &H2&
Global Const CC_PREVENTFULLOPEN = &H4&
Global Const CC_SHOWHELP = &H8&

'Fonts Dialog Flags
Global Const CF_SCREENFONTS = &H1&
Global Const CF_PRINTERFONTS = &H2&
Global Const CF_BOTH = &H3&
Global Const CF_SHOWHELP = &H4&
Global Const CF_INITTOLOGFONTSTRUCT = &H40&
Global Const CF_USESTYLE = &H80&
Global Const CF_EFFECTS = &H100&
Global Const CF_APPLY = &H200&
Global Const CF_ANSIONLY = &H400&
Global Const CF_NOVECTORFONTS = &H800&
Global Const CF_NOSIMULATIONS = &H1000&
Global Const CF_LIMITSIZE = &H2000&
Global Const CF_FIXEDPITCHONLY = &H4000&

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRESTRES.GBL - 15

```
Global Const CF_WYSIWYG = &H8000&           'must also have CF_SCREENFONTS &
CF_PRINTERFONTS
Global Const CF_FORCEFONTEXIST = &H10000
Global Const CF_SCALABLEONLY = &H20000
Global Const CF_TTONLY = &H40000
Global Const CF_NOFACESEL = &H80000
Global Const CF_MOSTYLESEL = &H100000
Global Const CF_NOSIZESEL = &H200000
```

'Printer Dialog Flags

```
Global Const PD_ALLPAGES = &H0&
Global Const PD_SELECTION = &H1&
Global Const PD_PAGENUMS = &H2&
Global Const PD_NOSELECTION = &H4&
Global Const PD_NOPAGENUMS = &H8&
Global Const PD_COLLATE = &H10&
Global Const PD_PRINTTOFILE = &H20&
Global Const PD_PRINTSETUP = &H40&
Global Const PD_NOWARNING = &H80&
Global Const PD_RETURNDC = &H100&
Global Const PD_RETURNIC = &H200&
Global Const PD_RETURNDEFAULT = &H400&
Global Const PD_SHOWHELP = &H800&
Global Const PD_USEDEVMODECOPIES = &H40000
Global Const PD_DISABLEPRINTTOFILE = &H80000
Global Const PD_HIDEPRINTTOFILE = &H100000
```

'Help Constants

```
Global Const HELP_CONTEXT = &H1           'Display topic in ulTopic
Global Const HELP_QUIT = &H2             'Terminate help
Global Const HELP_INDEX = &H3           'Display index
Global Const HELP_CONTENTS = &H3
Global Const HELP_HELPONHELP = &H4      'Display help on using help
Global Const HELP_SETINDEX = &H5        'Set the current Index for multi
index help
Global Const HELP_SETCONTENTS = &H5
Global Const HELP_CONTEXTPOPUP = &H8
Global Const HELP_FORCEFILE = &H9
Global Const HELP_KEY = &H101           'Display topic for keyword in of
fabData
Global Const HELP_COMMAND = &H102
Global Const HELP_PARTIALKEY = &H105    'Call the search engine in winhe
lp
```

'Error Constants

```
Global Const CDERR_DIALOGFAILURE = -32768
Global Const CDERR_GENERALCODES = &H7FFF
Global Const CDERR_STRUCTSIZE = &H7FFE
Global Const CDERR_INITIALIZATION = &H7FFD
Global Const CDERR_NOTEMPLATE = &H7FFC
Global Const CDERR_NOINSTANCE = &H7FFB
Global Const CDERR_LOADSTRFAILURE = &H7FFA
Global Const CDERR_FINDRESFAILURE = &H7FF9
Global Const CDERR_LOADRESFAILURE = &H7FF8
Global Const CDERR_LOCKRESFAILURE = &H7FF7
Global Const CDERR_MEMALLOCFailure = &H7FF6
```

PRESTRES.GBL - 16

Global Const CDERR_MEMLOCKFAILURE = &H7FF5
Global Const CDERR_NOHOOK = &H7FF4

'Added for CMDIALOG.VBX

Global Const CDERR_CANCEL = &H7FF3
Global Const CDERR_NODLL = &H7FF2
Global Const CDERR_ERRPROC = &H7FF1
Global Const CDERR_ALLOC = &H7FF0
Global Const CDERR_HELP = &H7FEF

Global Const PDERR_PRINTERCODES = &H6FFF
Global Const PDERR_SETUPFAILURE = &H6FFE
Global Const PDERR_PARSEFAILURE = &H6FFD
Global Const PDERR_RETDEFFAILURE = &H6FFC
Global Const PDERR_LOADDRVFAILURE = &H6FFB
Global Const PDERR_GETDEVMODEFAIL = &H6FFA
Global Const PDERR_INITFAILURE = &H6FF9
Global Const PDERR_NODEVICES = &H6FF8
Global Const PDERR_NODEFAULTPRN = &H6FF7
Global Const PDERR_DNDMMISMATCH = &H6FF6
Global Const PDERR_CREATEICFAILURE = &H6FF5
Global Const PDERR_PRINTERNOTFOUND = &H6FF4

Global Const CFERR_CHOOSFONTCODES = &H5FFF
Global Const CFERR_NOFONTS = &H5FFE

Global Const FNERR_FILENAMECODES = &H4FFF
Global Const FNERR_SUBCLASSFAILURE = &H4FFE
Global Const FNERR_INVALIDFILENAME = &H4FFD
Global Const FNERR_BUFFERTOOSMALL = &H4FFC

Global Const FRERR_FINDREPLACECODES = &H3FFF
Global Const CCERR_CHOOSCOLORCODES = &H2FFF

' Table of Contents for Visual Basic Professional

- ' 1. 3-D Controls
' (Frame/Panel/Option/Check/Command/Group Push)
' 2. Animated Button
' 3. Gauge Control
' 4. Graph Control Section
' 5. Key Status Control
' 6. Spin Button
' 7. MCI Control (Multimedia)
' 8. Masked Edit Control
' 9. Comm Control
' 10. Outline Control

'-----
' 3D Controls

'Alignment (Check Box)

Global Const SSCB_TEXT_RIGHT = 0 '0 - Text to the right

PRESTRES.GBL - 17

```

Global Const SSCB_TEXT_LEFT = 1           '1 - Text to the left

'Alignment (Option Button)
Global Const SSOB_TEXT_RIGHT = 0         '0 - Text to the right
Global Const SSOB_TEXT_LEFT = 1         '1 - Text to the left

'Alignment (Frame)
Global Const SSFR_LEFT_JUSTIFY = 0       '0 - Left justify text
Global Const SSFR_RIGHT_JUSTIFY = 1      '1 - Right justify text
Global Const SSFR_CENTER = 2            '2 - Center text

'Alignment (Panel)
Global Const SSPN_LEFT_TOP = 0           '0 - Text to left and top
Global Const SSPN_LEFT_MIDDLE = 1        '1 - Text to left and middle
Global Const SSPN_LEFT_BOTTOM = 2        '2 - Text to left and bottom
Global Const SSPN_RIGHT_TOP = 3          '3 - Text to right and top
Global Const SSPN_RIGHT_MIDDLE = 4       '4 - Text to right and middle
Global Const SSPN_RIGHT_BOTTOM = 5       '5 - Text to right and bottom
Global Const SSPN_CENTER_TOP = 6         '6 - Text to center and top
Global Const SSPN_CENTER_MIDDLE = 7      '7 - Text to center and middle
Global Const SSPN_CENTER_BOTTOM = 8      '8 - Text to center and bottom

'Autosize (Command Button)
Global Const SS_AUTOSIZE_NONE = 0        '0 - No Autosizing
Global Const SSPB_AUTOSIZE_PICTOBUT = 1  '0 - Autosize Picture to Button
Global Const SSPB_AUTOSIZE_BUTTOPIC = 2  '0 - Autosize Button to Picture

'Autosize (Ribbon Button)
Global Const SSRI_AUTOSIZE_NONE = 0      '0 - No Autosizing
Global Const SSRI_AUTOSIZE_PICTOBUT = 1  '0 - Autosize Picture to Button
Global Const SSRI_AUTOSIZE_BUTTOPIC = 2  '0 - Autosize Button to Picture

'Autosize (Panel)
Global Const SS_AUTOSIZE_NONE = 0        '0 - No Autosizing
Global Const SSPN_AUTOSIZE_WIDTH = 1     '1 - Autosize Panel width to Caption
Global Const SSPN_AUTOSIZE_HEIGHT = 2    '2 - Autosize Panel height to Caption
Global Const SSPN_AUTOSIZE_CHILD = 3     '3 - Autosize Child to Panel

'BevelInner (Panel)
Global Const SS_BEVELINNER_NONE = 0      '0 - No Inner Bevel
Global Const SS_BEVELINNER_INSET = 1     '1 - Inset Inner Bevel
Global Const SS_BEVELINNER_RAISED = 2    '2 - Raised Inner Bevel

'BevelOuter (Panel)
Global Const SS_BEVELOUTER_NONE = 0      '0 - No Outer Bevel
Global Const SS_BEVELOUTER_INSET = 1     '1 - Inset Outer Bevel
Global Const SS_BEVELOUTER_RAISED = 2    '2 - Raised Outer Bevel

'FloodType (Panel)
Global Const SS_FLOODTYPE_NONE = 0       '0 - No flood
Global Const SS_FLOODTYPE_L_TO_R = 1     '1 - Left to light
Global Const SS_FLOODTYPE_R_TO_L = 2     '2 - Right to left
Global Const SS_FLOODTYPE_T_TO_B = 3     '3 - Top to bottom
Global Const SS_FLOODTYPE_B_TO_T = 4     '4 - Bottom to top
Global Const SS_FLOODTYPE_CIRCLE = 5     '5 - Widening circle

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRESTRES.GBL - 18

```

'Font3D (Panel, Command Button, Option Button, Check Box, Frame)
Global Const SS_FONT3D_NONE = 0          '0 - No 3-D text
Global Const SS_FONT3D_RAISED_LIGHT = 1  '1 - Raised with light shading
Global Const SS_FONT3D_RAISED_HEAVY = 2  '2 - Raised with heavy shading
Global Const SS_FONT3D_INSET_LIGHT = 3   '3 - Inset with light shading
Global Const SS_FONT3D_INSET_HEAVY = 4   '4 - Inset with heavy shading
,
'PictureDnChange (Ribbon Button)
Global Const SS_PICDN_NOCHANGE = 0       '0 - Use 'Up'bitmap with no chang
e
Global Const SS_PICDN_DITHER = 1        '1 - Dither 'Up'bitmap
Global Const SS_PICDN_INVERT = 2        '2 - Invert 'Up'bitmap

'ShadowColor (Panel, Frame)
Global Const SS_SHADOW_DARKGREY = 0     '0 - Dark grey shadow
Global Const SS_SHADOW_BLACK = 1        '1 - Black shadow

'ShadowStyle (Frame)
Global Const SS_SHADOW_INSET = 0        '0 - Shadow inset
Global Const SS_SHADOW_RAISED = 1       '1 - Shadow raised

'-----
'Animated Button
'-----
'Cycle property
Global Const ANI_ANIMATED = 0
Global Const ANI_MULTISTATE = 1
Global Const ANI_TWO_STATE = 2

'Click Filter property
Global Const ANI_ANYWHERE = 0
Global Const ANI_IMAGE_AND_TEXT = 1
Global Const ANI_IMAGE = 2
Global Const ANI_TEXT = 3

'PicDrawMode Property
Global Const ANI_XPOS_YPOS = 0
Global Const ANI_AUTOSIZE = 1
Global Const ANI_STRETCH = 2

'SpecialOp Property
Global Const ANI_CLICK = 1

'TextPosition Property
Global Const ANI_CENTER = 0
Global Const ANI_LEFT = 1
Global Const ANI_RIGHT = 2
Global Const ANI_BOTTOM = 3
Global Const ANI_TOP = 4

'-----
'GAUGE
'-----
'Style Property

```

PRESTRES.GBL - 19

Global Const GAUGE_HORIZ = 0
 Global Const GAUGE_VERT = 1
 Global Const GAUGE_SEMI = 2
 Global Const GAUGE_FULL = 3

 'Graph Control

'General

Global Const G_NONE = 0
 Global Const G_DEFAULT = 0

Global Const G_OFF = 0
 Global Const G_ON = 1

Global Const G_MONO = 0
 Global Const G_COLOR = 1

'Graph Types

Global Const G_PIE2D = 1
 Global Const G_PIE3D = 2
 Global Const G_BAR2D = 3
 Global Const G_BAR3D = 4
 Global Const G_GANTT = 5
 Global Const G_LINE = 6
 Global Const G_LOGLIN = 7
 Global Const G_AREA = 8
 Global Const G_SCATTER = 9
 Global Const G_POLAR = 10
 Global Const G_HLC = 11

'Colors

Global Const G_BLACK = 0
 Global Const G_BLUE = 1
 Global Const G_GREEN = 2
 Global Const G_CYAN = 3
 Global Const G_RED = 4
 Global Const G_MAGENTA = 5
 Global Const G_BROWN = 6
 Global Const G_LIGHT_GRAY = 7
 Global Const G_DARK_GRAY = 8
 Global Const G_LIGHT_BLUE = 9
 Global Const G_LIGHT_GREEN = 10
 Global Const G_LIGHT_CYAN = 11
 Global Const G_LIGHT_RED = 12
 Global Const G_LIGHT_MAGENTA = 13
 Global Const G_YELLOW = 14
 Global Const G_WHITE = 15
 Global Const G_AUTOBW = 16

'Patterns

Global Const G_SOLID = 0
 Global Const G_HOLLOW = 1
 Global Const G_HATCH1 = 2
 Global Const G_HATCH2 = 3
 Global Const G_HATCH3 = 4



PRESTRES.GBL - 20

```

Global Const G_HATCH4 = 5
Global Const G_HATCH5 = 6
Global Const G_HATCH6 = 7
Global Const G_BITMAP1 = 16
Global Const G_BITMAP2 = 17
Global Const G_BITMAP3 = 18
Global Const G_BITMAP4 = 19
Global Const G_BITMAP5 = 20
Global Const G_BITMAP6 = 21
Global Const G_BITMAP7 = 22
Global Const G_BITMAP8 = 23
Global Const G_BITMAP9 = 24
Global Const G_BITMAP10 = 25
Global Const G_BITMAP11 = 26
Global Const G_BITMAP12 = 27
Global Const G_BITMAP13 = 28
Global Const G_BITMAP14 = 29
Global Const G_BITMAP15 = 30
Global Const G_BITMAP16 = 31

```

'Symbols

```

Global Const G_CROSS_PLUS = 0
Global Const G_CROSS_TIMES = 1
Global Const G_TRIANGLE_UP = 2
Global Const G_SOLID_TRIANGLE_UP = 3
Global Const G_TRIANGLE_DOWN = 4
Global Const G_SOLID_TRIANGLE_DOWN = 5
Global Const G_SQUARE = 6
Global Const G_SOLID_SQUARE = 7
Global Const G_DIAMOND = 8
Global Const G_SOLID_DIAMOND = 9

```

'Line Styles

```

Global Const G_SOLID = 0
Global Const G_DASH = 1
Global Const G_DOT = 2
Global Const G_DASHDOT = 3
Global Const G_DASHDOTDOT = 4

```

'Grids

```

Global Const G_HORIZONTAL = 1
Global Const G_VERTICAL = 2

```

'Statistics

```

Global Const G_MEAN = 1
Global Const G_MIN_MAX = 2
Global Const G_STD_DEV = 4
Global Const G_BEST_FIT = 8

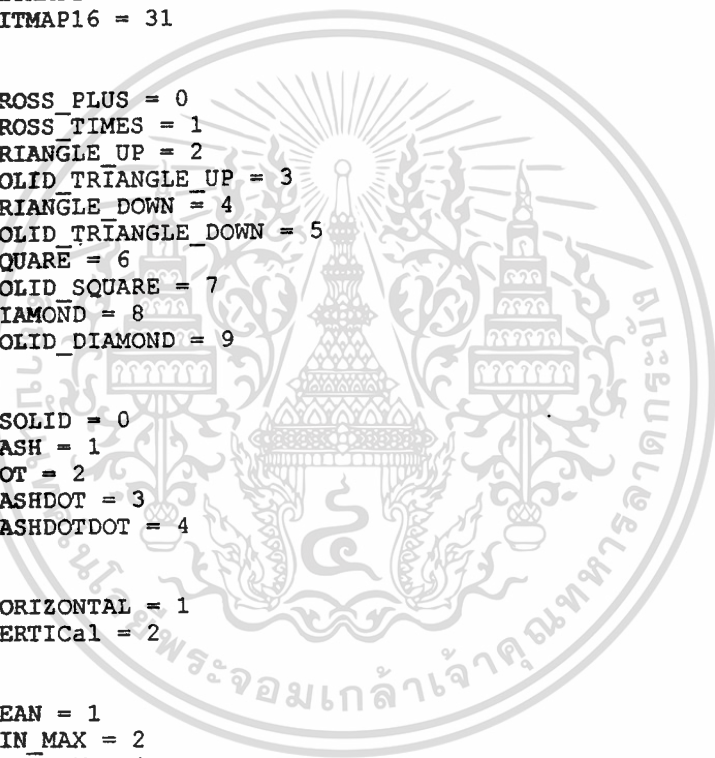
```

'Data Arrays

```

Global Const G_GRAPH_DATA = 1
Global Const G_COLOR_DATA = 2
Global Const G_EXTRA_DATA = 3
Global Const G_LABEL_TEXT = 4
Global Const G_LEGEND_TEXT = 5
Global Const G_PATTERN_DATA = 6
Global Const G_SYMBOL_DATA = 7

```



PRESTRES.GBL - 21

Global Const G_XPOS_DATA = 8
Global Const G_ALL_DATA = 9

'Draw Mode

Global Const G_NO_ACTION = 0
Global Const G_CLEAR = 1
Global Const G_DRAW = 2
Global Const G_BLIT = 3
Global Const G_COPY = 4
Global Const G_PRINT = 5
Global Const G_WRITE = 6

'Print Options

Global Const G_BORDER = 2

'Pie Chart Options

Global Const G_NO_LINES = 1
Global Const G_COLORED = 2
Global Const G_PERCENTS = 4

'Bar Chart Options

Global Const G_HORIZONTAL = 1
Global Const G_STACKED = 2
Global Const G_PERCENTAGE = 4
Global Const G_Z_CLUSTERED = 6

'Gantt Chart Options

Global Const G_SPACED_BARS = 1

'Line/Polar Chart Options

Global Const G_SYMBOLS = 1
Global Const G_STICKS = 2
Global Const G_LINES = 4

'Area Chart Options

Global Const G_ABSOLUTE = 1
Global Const G_PERCENT = 2

'HLC Chart Options

Global Const G_NO_CLOSE = 1
Global Const G_NO_HIGH_LOW = 2

'Key Status Control

'Style

Global Const KEYSTAT_CAPSLOCK = 0
Global Const KEYSTAT_NUMLOCK = 1
Global Const KEYSTAT_INSERT = 2
Global Const KEYSTAT_SCROLLLOCK = 3

'MCI Control (Multimedia)

'NOTE:

PRESTRES.GBL - 22

'Please use the updated Multimedia constants
'in the WINMMSYS.TXT file from the \VB\WINAPI
'subdirectory.

'Mode Property

'Global Const MCI_MODE_NOT_OPEN = 11
'Global Const MCI_MODE_STOP = 12
'Global Const MCI_MODE_PLAY = 13
'Global Const MCI_MODE_RECORD = 14
'Global Const MCI_MODE_SEEK = 15
'Global Const MCI_MODE_PAUSE = 16
'Global Const MCI_MODE_READY = 17

'NotifyValue Property

'Global Const MCI_NOTIFY_SUCCESSFUL = 1
'Global Const MCI_NOTIFY_SUPERSEDED = 2
'Global Const MCI_ABORTED = 4
'Global Const MCI_FAILURE = 8

'Orientation Property

'Global Const MCI_ORIENT_HORZ = 0
'Global Const MCI_ORIENT_VERT = 1

'RecordMode Property

'Global Const MCI_RECORD_INSERT = 0
'Global Const MCI_RECORD_OVERWRITE = 1

'TimeFormat Property

'Global Const MCI_FORMAT_MILLISECONDS = 0
'Global Const MCI_FORMAT_HMS = 1
'Global Const MCI_FORMAT_MSF = 2
'Global Const MCI_FORMAT_FRAMES = 3
'Global Const MCI_FORMAT_SMPTE_24 = 4
'Global Const MCI_FORMAT_SMPTE_25 = 5
'Global Const MCI_FORMAT_SMPTE_30 = 6
'Global Const MCI_FORMAT_SMPTE_30DROP = 7
'Global Const MCI_FORMAT_BYTES = 8
'Global Const MCI_FORMAT_SAMPLES = 9
'Global Const MCI_FORMAT_TMSF = 10

'-----
'Spin Button
'-----

'SpinOrientation

Global Const SPIN_VERTICAL = 0
Global Const SPIN_HORIZONTAL = 1

'-----
'Masked Edit Control
'-----

'ClipMode

Global Const ME_INCLIT = 0
Global Const ME_EXCLIT = 1

PRESTRES.GBL - 23

```

'-----
'Comm Control
'-----
'Handshaking
Global Const MSCOMM_HANDSHAKE_NONE = 0
Global Const MSCOMM_HANDSHAKE_XONXOFF = 1
Global Const MSCOMM_HANDSHAKE_RTS = 2
Global Const MSCOMM_HANDSHAKE_RTSXONXOFF = 3

'Event constants
Global Const MSCOMM_EV_SEND = 1
Global Const MSCOMM_EV_RECEIVE = 2
Global Const MSCOMM_EV_CTS = 3
Global Const MSCOMM_EV_DSR = 4
Global Const MSCOMM_EV_CD = 5
Global Const MSCOMM_EV_RING = 6
Global Const MSCOMM_EV_EOF = 7

'Error code constants
Global Const MSCOMM_ER_BREAK = 1001
Global Const MSCOMM_ER_CTSTO = 1002
Global Const MSCOMM_ER_DSRTO = 1003
Global Const MSCOMM_ER_FRAME = 1004
Global Const MSCOMM_ER_OVERRUN = 1006
Global Const MSCOMM_ER_CDTO = 1007
Global Const MSCOMM_ER_RXOVER = 1008
Global Const MSCOMM_ER_RXPARITY = 1009
Global Const MSCOMM_ER_TXFULL = 1010

'-----
' MAPI SESSION CONTROL CONSTANTS
'-----
'Action
Global Const SESSION_SIGNON = 1
Global Const SESSION_SIGNOFF = 2

'-----
' MAPI MESSAGE CONTROL CONSTANTS
'-----
'Action
Global Const MESSAGE_FETCH = 1           ' Load all messages from message store
Global Const MESSAGE_SENDDLG = 2         ' Send mail bring up default mapi dialog
Global Const MESSAGE_SEND = 3            ' Send mail without default mapi dialog
Global Const MESSAGE_SAVEMSG = 4         ' Save message in the compose buffer
Global Const MESSAGE_COPY = 5            ' Copy current message to compose buffer
Global Const MESSAGE_COMPOSE = 6         ' Initialize compose buffer (previous data is lost)
Global Const MESSAGE_REPLY = 7           ' Fill Compose buffer as REPLY
Global Const MESSAGE_REPLYALL = 8        ' Fill Compose buffer as REPLY

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRESTRES.GBL - 24

```

ALL
Global Const MESSAGE_FORWARD = 9      ' Fill Compose buffer as FORWARD
D
Global Const MESSAGE_DELETE = 10      ' Delete current message
Global Const MESSAGE_SHOWADBOOK = 11  ' Show Address book
Global Const MESSAGE_SHOWDETAILS = 12  ' Show details of the current recipient
Global Const MESSAGE_RESOLVENAME = 13  ' Resolve the display name of the recipient
Global Const RECIPIENT_DELETE = 14     ' Fill Compose buffer as FORWARD
Global Const ATTACHMENT_DELETE = 15    ' Delete current message

```

' ERROR CONSTANT DECLARATIONS (MAPI CONTROLS)
'-----

```

Global Const SUCCESS_SUCCESS = 32000
Global Const MAPI_USER_ABORT = 32001
Global Const MAPI_E_FAILURE = 32002
Global Const MAPI_E_LOGIN_FAILURE = 32003
Global Const MAPI_E_DISK_FULL = 32004
Global Const MAPI_E_INSUFFICIENT_MEMORY = 32005
Global Const MAPI_E_ACCESS_DENIED = 32006
Global Const MAPI_E_TOO_MANY_SESSIONS = 32008
Global Const MAPI_E_TOO_MANY_FILES = 32009
Global Const MAPI_E_TOO_MANY_RECIPIENTS = 32010
Global Const MAPI_E_ATTACHMENT_NOT_FOUND = 32011
Global Const MAPI_E_ATTACHMENT_OPEN_FAILURE = 32012
Global Const MAPI_E_ATTACHMENT_WRITE_FAILURE = 32013
Global Const MAPI_E_UNKNOWN_RECIPIENT = 32014
Global Const MAPI_E_BAD_RECIPTYPE = 32015
Global Const MAPI_E_NO_MESSAGES = 32016
Global Const MAPI_E_INVALID_MESSAGE = 32017
Global Const MAPI_E_TEXT_TOO_LARGE = 32018
Global Const MAPI_E_INVALID_SESSION = 32019
Global Const MAPI_E_TYPE_NOT_SUPPORTED = 32020
Global Const MAPI_E_AMBIGUOUS_RECIPIENT = 32021
Global Const MAPI_E_MESSAGE_IN_USE = 32022
Global Const MAPI_E_NETWORK_FAILURE = 32023
Global Const MAPI_E_INVALID_EDITFIELDS = 32024
Global Const MAPI_E_INVALID_RECIPS = 32025
Global Const MAPI_E_NOT_SUPPORTED = 32026

Global Const CONTROL_E_SESSION_EXISTS = 32050
Global Const CONTROL_E_INVALID_BUFFER = 32051
Global Const CONTROL_E_INVALID_READ_BUFFER_ACTION = 32052
Global Const CONTROL_E_NO_SESSION = 32053
Global Const CONTROL_E_INVALID_RECIPIENT = 32054
Global Const CONTROL_E_INVALID_COMPOSE_BUFFER_ACTION = 32055
Global Const CONTROL_E_FAILURE = 32056
Global Const CONTROL_E_NO_RECIPIENTS = 32057
Global Const CONTROL_E_NO_ATTACHMENTS = 32058

```

' MISCELLANEOUS GLOBAL CONSTANT DECLARATIONS (MAPI CONTROLS)
'-----

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRESTRES.GBL - 25

```

'-----
Global Const RECIPTYPE_ORIG = 0
Global Const RECIPTYPE_TO = 1
Global Const RECIPTYPE_CC = 2
Global Const RECIPTYPE_BCC = 3

Global Const ATTACHTYPE_DATA = 0
Global Const ATTACHTYPE_EOLE = 1
Global Const ATTACHTYPE_SOLE = 2

'-----
' Outline
'-----
' PictureType
Global Const MSOUTLINE_PICTURE_CLOSED = 0
Global Const MSOUTLINE_PICTURE_OPEN = 1
Global Const MSOUTLINE_PICTURE_LEAF = 2

'Outline Control Error Constants
Global Const MSOUTLINE_BADPICFORMAT = 32000
Global Const MSOUTLINE_BADINDENTATION = 32001
Global Const MSOUTLINE_MEM = 32002
Global Const MSOUTLINE_PARENTNOTEEXPANDED = 32003

''''WinFlag''''
Global Const WF_PMODE = &H1
Global Const WF_CPU286 = &H2
Global Const WF_CPU386 = &H4
Global Const WF_CPU486 = &H8
Global Const WF_STANDARD = &H10
Global Const WF_WIN286 = &H10
Global Const WF_ENHANCED = &H20
Global Const WF_WIN386 = &H20
Global Const WF_CPU086 = &H40
Global Const WF_CPU186 = &H80
Global Const WF_LARGEFRAME = &H100
Global Const WF_SMALLFRAME = &H200
Global Const WF_80x87 = &H400

''''SystemMetric''''
Global Const SM_CXSCREEN = 0
Global Const SM_CYSCREEN = 1
Global Const SM_CXVSCROLL = 2
Global Const SM_CYHSCROLL = 3
Global Const SM_CYCAPTION = 4
Global Const SM_CXBORDER = 5
Global Const SM_CYBORDER = 6
Global Const SM_CXDLGFRAME = 7
Global Const SM_CYDLGFRAME = 8
Global Const SM_CYVTHUMB = 9
Global Const SM_CXHTHUMB = 10
Global Const SM_CXICON = 11
Global Const SM_CYICON = 12
Global Const SM_CXCURSOR = 13
Global Const SM_CYCURSOR = 14

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRESTRES.GBL - 26

Global Const SM_CYMENU = 15
Global Const SM_CXFULLSCREEN = 16
Global Const SM_CYFULLSCREEN = 17
Global Const SM_CYKANJIWINDOW = 18
Global Const SM_MOUSEPRESENT = 19
Global Const SM_CYVSCROLL = 20
Global Const SM_CXHSCROLL = 21
Global Const SM_DEBUG = 22
Global Const SM_SWAPBUTTON = 23
Global Const SM_RESERVED1 = 24
Global Const SM_RESERVED2 = 25
Global Const SM_RESERVED3 = 26
Global Const SM_RESERVED4 = 27
Global Const SM_CXMIN = 28
Global Const SM_CYMIN = 29
Global Const SM_CXSIZE = 30
Global Const SM_CYSIZE = 31
Global Const SM_CXFRAME = 32
Global Const SM_CYFRAME = 33
Global Const SM_CXMINTRACK = 34
Global Const SM_CYMINTRACK = 35



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRANSDAT.BAS - 1

```

Function Find_Bottom (BottomY As Integer, SC As Integer, XY As Integer) As Integer
    Dim MinY As Single
    Dim MaxY As Single
    Dim v As Integer
    Dim I As Integer
    Dim Found As Integer
    v = False
    Found = False
    MinY = Column(CProj, BottomY).Co_Y
    For I = 1 To SNodeNum(CProj)
        If (Column(CProj, I).Co_Y > MinY) And In_Length(XY, I, Column(CProj, SC).Co_X - Ec, Column(CProj, SC).Co_X + Ec) Then 'find column inlength and more than column
            If Not v Then
                MaxY = Column(CProj, I).Co_Y
                Find_Bottom = I
                v = True
                Found = True
            End If
            If Column(CProj, I).Co_Y <= MaxY Then 'find nearest column
                MaxY = Column(CProj, I).Co_Y
                Find_Bottom = I
                Found = True
            End If
        Else
            If Not Found Then
                Find_Bottom = 0 'Find_Bottom = 0 is Not Founded
            End If
        End If
    Next I
End Function

Function FInd_Left (LeftX As Integer, SC As Integer, XY As Integer) As Integer
    Dim MinX As Single
    Dim MaxX As Single
    Dim v As Integer
    Dim I As Integer
    Dim Found As Integer
    v = False
    Found = False
    MinX = Column(CProj, LeftX).Co_X
    For I = 1 To SNodeNum(CProj)
        If (Column(CProj, I).Co_X > MinX) And In_Length(XY, I, Column(CProj, SC).Co_Y - Ec, Column(CProj, SC).Co_Y + Ec) Then
            If Not v Then
                MaxX = Column(CProj, I).Co_X
                FInd_Left = I
                v = True
                Found = True
            End If
            If Column(CProj, I).Co_X <= MaxX Then
                MaxX = Column(CProj, I).Co_X
                FInd_Left = I
            End If
        End If
    Next I
End Function

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRANSDAT.BAS - 2

```

    Found = True
End If
Else
    If Not Found Then
        Find_Left = 0
    End If
End If
Next I
End Function

```

Function Find_Right (RightX As Integer, SC As Integer, XY As Integer) As Integer

```

Dim MinX As Single
Dim MaxX As Single
Dim I As Integer
Dim v As Integer
Dim Found As Integer
v = False
Found = False
MaxX = Column(CProj, RightX).Co_X
For I = 1 To SNodeNum(CProj)
    If (Column(CProj, I).Co_X < MaxX) And In_Length(XY, I, Column(CProj,
SC).Co_Y - Ec, Column(CProj, SC).Co_Y + Ec) Then
        If Not v Then
            MinX = Column(CProj, I).Co_X
            Find_Right = I
            Found = True
            v = True
        End If
        If Column(CProj, I).Co_X >= MinX Then
            MinX = Column(CProj, I).Co_X
            Find_Right = I
            Found = True
        End If
    Else
        If Not Found Then
            Find_Right = 0
        End If
    End If
Next I
End Function

```

Function Find_Top (BottomY As Integer, SC As Integer, XY As Integer) As Integer

```

Dim MinY As Single
Dim MaxY As Single
Dim v As Integer
Dim I As Integer
Dim Found As Integer
v = False
Found = False
MaxY = Column(CProj, BottomY).Co_Y
For I = 1 To SNodeNum(CProj)
    If (Column(CProj, I).Co_Y < MaxY) And In_Length(XY, I, Column(CProj,
SC).Co_X - Ec, Column(CProj, SC).Co_X + Ec) Then
        If Not v Then
            MinY = Column(CProj, I).Co_Y

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRANSDAT.BAS - 3

```

Find_Top = I
Found = True
v = True
End If
If Column(CProj, I).Co_Y >= MinY Then
MinY = Column(CProj, I).Co_Y
Find_Top = I
Found = True
End If
Else
If Not Found Then
Find_Top = 0
End If
End If
Next I
End Function

```

```

Function Find_X (X As Integer) As Integer
Dim MinX As Single
Dim v As Integer
v = False
If X = 0 Then
For I = 1 To SNodeNum(CProj)
If Column(CProj, I).Y_Flag = 1 Then
If Not v Then
MaxX = Column(CProj, I).Co_X
Find_X = I
v = True
End If
If Column(CProj, I).Co_X < MaxX Then
Find_X = I
End If
End If
Next I
ElseIf X > 0 Then
MinX = Column(CProj, X).Co_X
For I = 1 To SNodeNum(CProj)
If Column(CProj, I).Y_Flag = 1 Then
If Column(CProj, I).Co_X > MinX Then
If Not v Then
MaxX = Column(CProj, I).Co_X
Find_X = I
v = True
End If
If Column(CProj, I).Co_X < MaxX Then
MaxX = Column(CProj, I).Co_X
Find_X = I
End If
End If
Next I
End If
End If
End Function

```

```

Function Find_Y (Y As Integer) As Integer
Dim MinY As Single
Dim v As Integer

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRANSDAT.BAS - 4

```

v = False
If Y = 0 Then
  For I = 1 To SNodeNum(CProj)
    If Column(CProj, I).X_Flag = 1 Then
      If Not v Then
        MaxY = Column(CProj, I).Co_Y
        Find Y = I
        v = True
      End If
      If Column(CProj, I).Co_Y < MaxY Then
        Find Y = I
      End If
    End If
  Next I
ElseIf Y > 0 Then
  MinY = Column(CProj, Y).Co_Y
  For I = 1 To SNodeNum(CProj)
    If Column(CProj, I).X_Flag = 1 Then
      If Column(CProj, I).Co_Y > MinY Then
        If Not v Then
          MaxY = Column(CProj, I).Co_Y
          Find Y = I
          v = True
        End If
        If Column(CProj, I).Co_Y < MaxY Then
          MaxY = Column(CProj, I).Co_Y
          Find Y = I
        End If
      End If
    End If
  Next I
End If
End Function

Function In_Length (XY, I As Integer, Min As Single, Max As Single) As Integer
  If XY Then
    If ((Column(CProj, I).Co_X >= Min) And (Column(CProj, I).Co_X <= Max)) Then
  ) Then
    In_Length = True
  Else
    In_Length = False
  End If
  ElseIf Not (XY) Then
    If ((Column(CProj, I).Co_Y >= Min) And (Column(CProj, I).Co_Y <= Max))
  ) Then
    In_Length = True
  Else
    In_Length = False
  End If
  End If
End Function

Function Max (a As Single, b As Single)
  Max = a
  If b > a Then
    Max = b
  
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRANSDAT.BAS - 5

End If
End Function

```

Sub Show_Element ()
  Dim axis As Integer
  Dim N As Integer
  N = 0
  For I = 1 To SRow(CProj)
    For J = 1 To RowEle(I)
      N = N + 1
      frmMain.grdElement.Row = N
      frmMain.grdElement.Col = 1: frmMain.grdElement.Text = "X"
      frmMain.grdElement.Col = 2: frmMain.grdElement.Text = Format$(I)
      frmMain.grdElement.Col = 3: frmMain.grdElement.Text = Format$(J)
      frmMain.grdElement.Col = 4: frmMain.grdElement.Text = Format$(EleCol
Row(CProj, 0, I, J).Start)
      frmMain.grdElement.Col = 5: frmMain.grdElement.Text = Format$(EleCol
Row(CProj, 0, I, J).End)
      frmMain.grdElement.Col = 6: frmMain.grdElement.Text = Format$(EleCol
Row(CProj, 0, I, J).Length)
      frmMain.grdElement.Col = 7: frmMain.grdElement.Text = Format$(EleCol
Row(CProj, 0, I, J).L1)
      frmMain.grdElement.Col = 8: frmMain.grdElement.Text = Format$(EleCol
Row(CProj, 0, I, J).L2)
    Next J
  Next I
  For I = 1 To SCol(CProj)
    For J = 1 To ColEle(I)
      N = N + 1
      frmMain.grdElement.Row = N
      frmMain.grdElement.Col = 1: frmMain.grdElement.Text = "Y"
      frmMain.grdElement.Col = 2: frmMain.grdElement.Text = Format$(I)
      frmMain.grdElement.Col = 3: frmMain.grdElement.Text = Format$(J)
      frmMain.grdElement.Col = 4: frmMain.grdElement.Text = Format$(EleCol
Row(CProj, 1, I, J).Start)
      frmMain.grdElement.Col = 5: frmMain.grdElement.Text = Format$(EleCol
Row(CProj, 1, I, J).End)
      frmMain.grdElement.Col = 6: frmMain.grdElement.Text = Format$(EleCol
Row(CProj, 1, I, J).Length)
      frmMain.grdElement.Col = 7: frmMain.grdElement.Text = Format$(EleCol
Row(CProj, 1, I, J).L1)
      frmMain.grdElement.Col = 8: frmMain.grdElement.Text = Format$(EleCol
Row(CProj, 1, I, J).L2)
    Next J
  Next I
End Sub

```

```

Sub Trans_Element (ColOrRow As Integer, ByVal ColRow As Integer) ' ColOrR
ow = True Cal Column Dir
  Const YE = 1
  Const XE = 0
  If ColOrRow Then
    For I = 1 To ColEle(ColRow)
      Element(CProj, I) = EleColRow(CProj, YE, ColRow, I)
      LoadData(CProj, I) = SLoad(CProj, YE, ColRow, I)
      LoadData(CProj, I).ElementRef = I
      Node(CProj, I).Co_X = Column(CProj, Element(CProj, I).Start).Co_X

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRANSDAT.BAS - 6

```

Node(CProj, I).Co_Y = Column(CProj, Element(CProj, I).Start).Co_Y
Node(CProj, I).SupportType = Column(CProj, Element(CProj, I).Start).
SupportType
Element(CProj, I).CStart = EleColRow(CProj, YE, ColRow, I).Start
Element(CProj, I).Start = I
Node(CProj, I + 1).Co_X = Column(CProj, Element(CProj, I).End).Co_X
Node(CProj, I + 1).Co_Y = Column(CProj, Element(CProj, I).End).Co_Y
Node(CProj, I + 1).SupportType = Column(CProj, Element(CProj, I).End
).SupportType
Element(CProj, I).CEnd = EleColRow(CProj, YE, ColRow, I).End
Element(CProj, I).End = I + 1
Next I
NodeNum(CProj) = ColEle(ColRow) + 1
ElementNum(CProj) = ColEle(ColRow)
LoadNum(CProj) = ColEle(ColRow)
Elseif Not ColOrRow Then
For I = 1 To RowEle(ColRow)
Element(CProj, I) = EleColRow(CProj, XE, ColRow, I)
LoadData(CProj, I) = SLoad(CProj, XE, ColRow, I)
LoadData(CProj, I).ElementRef = I
Node(CProj, I).Co_X = Column(CProj, Element(CProj, I).Start).Co_X
Node(CProj, I).Co_Y = Column(CProj, Element(CProj, I).Start).Co_Y
Node(CProj, I).SupportType = Column(CProj, Element(CProj, I).Start).
SupportType
Element(CProj, I).CStart = EleColRow(CProj, XE, ColRow, I).Start
Element(CProj, I).Start = I
Node(CProj, I + 1).Co_X = Column(CProj, Element(CProj, I).End).Co_X
Node(CProj, I + 1).Co_Y = Column(CProj, Element(CProj, I).End).Co_Y
Node(CProj, I + 1).SupportType = Column(CProj, Element(CProj, I).End
).SupportType
Element(CProj, I).CEnd = EleColRow(CProj, XE, ColRow, I).End
Element(CProj, I).End = I + 1
Next I
NodeNum(CProj) = RowEle(ColRow) + 1
ElementNum(CProj) = RowEle(ColRow)
LoadNum(CProj) = RowEle(ColRow)
End If
End Sub

Sub Trans_Element ()
Dim I As Integer
Dim J As Integer
Dim K As Integer
Dim EleCol As Integer
Dim EleRow As Integer
Dim MinX As Integer
Dim MinY As Integer
Dim L1 As Single
Dim L2 As Single
Dim SColColumn As Integer
Dim SRowColumn As Integer
Const Ec = 3 'eccentric of column equal 3 meters
Const XE = 0 'Code for X axis
Const YE = 1 'Code for Y axis
SCol(CProj) = 0 'Number of Col
EleCol = 0 '1-100 is Column Number
SColColumn = 0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRANSDAT.BAS - 7

```

'''' Find Element in Column & Row ,EleColRow
For I = 1 To SNodeNum(CProj)

'''' In Y axis
If ((Column(CProj, I).Y_Flag = 1) And (Find_X(SColColumn) = I)) Then
'Check Next Right Column
SColColumn = I
MinY = I
SCol(CProj) = SCol(CProj) + 1
EleCol = 0
For J = 1 To SNodeNum(CProj)
If In_Length(True, J, Column(CProj, SColColumn).Co_X - Ec, Column(C
Proj, SColColumn).Co_X + Ec) And (Find_Bottom(MinY, SColColumn, True) = J)
Then 'Check Next Element
EleCol = EleCol + 1
EleColRow(CProj, YE, SCol(CProj), EleCol).Start = MinY
EleColRow(CProj, YE, SCol(CProj), EleCol).End = J
EleColRow(CProj, YE, SCol(CProj), EleCol).Length = Column(CProj, J
).Co_Y - Column(CProj, MinY).Co_Y
MinY = J
J = 1 '*****
End If
Next J
ColEle(SCol(CProj)) = EleCol
I = 1 '*****
End If
Next I

'''' In Y axis
SRow(CProj) = 0
EleRow = 0
SRowColumn = 0
For I = 1 To SNodeNum(CProj)
If ((Column(CProj, I).X_Flag = 1) And (Find_Y(SRowColumn) = I)) Then
'Check Next Top Row
SRowColumn = I
MinX = I
SRow(CProj) = SRow(CProj) + 1
EleRow = 0
For J = 1 To SNodeNum(CProj)
If In_Length(False, J, Column(CProj, I).Co_Y - Ec, Column(CProj, I)
.Co_Y + Ec) And (Find_Left(MinX, SRowColumn, False) = J) Then 'Check Ne
xt Row Element
EleRow = EleRow + 1
EleColRow(CProj, XE, SRow(CProj), EleRow).Start = MinX
EleColRow(CProj, XE, SRow(CProj), EleRow).End = J
EleColRow(CProj, XE, SRow(CProj), EleRow).Length = Column(CProj, J
).Co_X - Column(CProj, MinX).Co_X
MinX = J
J = 1 '*****
End If
Next J
RowEle(SRow(CProj)) = EleRow
I = 1 '*****
End If
Next I

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRANSDAT.BAS - 8

```

'''' Transform To Element Only Y axis ,SElement
ElementYNum(CProj) = 0
For I = 1 To SCol(CProj)
  For J = 1 To ColEle(I)
    ElementYNum(CProj) = ElementYNum(CProj) + 1
    SElement(CProj, YE, ElementYNum(CProj)) = EleColRow(CProj, YE, I, J)
  Next J
Next I

```

```

'''' Transform To Element Only X axis ,SElement
ElementXNum(CProj) = 0
For I = 1 To SRow(CProj)
  For J = 1 To RowEle(I)
    ElementXNum(CProj) = ElementXNum(CProj) + 1
    SElement(CProj, XE, ElementXNum(CProj)) = EleColRow(CProj, XE, I, J)
  Next J
Next I

```

```

'''' Find L1,L2 in Row
For I = 1 To SRow(CProj)
  For J = 1 To RowEle(I)
    L1 = 0
    L2 = 0
    For K = 1 To ElementYNum(CProj)
      If (EleColRow(CProj, XE, I, J).Start = SElement(CProj, YE, K).End)
Or (EleColRow(CProj, XE, I, J).End = SElement(CProj, YE, K).End) Then
        If L1 < SElement(CProj, YE, K).Length Then
          L1 = SElement(CProj, YE, K).Length
        End If
      End If
      If (EleColRow(CProj, XE, I, J).Start = SElement(CProj, YE, K).Start
) Or (EleColRow(CProj, XE, I, J).End = SElement(CProj, YE, K).Start) Then
        If L2 < SElement(CProj, YE, K).Length Then
          L2 = SElement(CProj, YE, K).Length
        End If
      End If
    Next K
    EleColRow(CProj, XE, I, J).L1 = L1
    EleColRow(CProj, XE, I, J).L2 = L2
  Next J
Next I

```

```

'''' Find L1,L2 in Column
For I = 1 To SCol(CProj)
  For J = 1 To ColEle(I)
    L1 = 0
    L2 = 0
    For K = 1 To ElementXNum(CProj)
      If (EleColRow(CProj, YE, I, J).Start = SElement(CProj, XE, K).End)
Or (EleColRow(CProj, YE, I, J).End = SElement(CProj, XE, K).End) Then
        If L1 < SElement(CProj, XE, K).Length Then
          L1 = SElement(CProj, XE, K).Length
        End If
      End If
      If (EleColRow(CProj, YE, I, J).Start = SElement(CProj, XE, K).Start
) Or (EleColRow(CProj, YE, I, J).End = SElement(CProj, XE, K).Start) Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRANSDAT.BAS - 9

```

    If L2 < SElement(CProj, XE, K).Length Then
        L2 = SElement(CProj, XE, K).Length
    End If
End If
Next K
EleColRow(CProj, YE, I, J).L1 = L1
EleColRow(CProj, YE, I, J).L2 = L2
Next J
Next I

''' Transform To All Element in project,EElement
For I = 1 To ElementYNum(CProj)
    EElement(CProj, I) = SElement(CProj, 1, I)
Next I
For I = 1 To ElementXNum(CProj)
    EElement(CProj, ElementYNum(CProj) + I) = SElement(CProj, 0, I)
Next I

''' Replace L1,L2
ElementYNum(CProj) = 0
For I = 1 To SCol(CProj)
    For J = 1 To ColEle(I)
        ElementYNum(CProj) = ElementYNum(CProj) + 1
        SElement(CProj, YE, ElementYNum(CProj)) = EleColRow(CProj, YE, I, J)
    Next J
Next I

''' Transform To Element Only X axis ,SElement
ElementXNum(CProj) = 0
For I = 1 To SRow(CProj)
    For J = 1 To RowEle(I)
        ElementXNum(CProj) = ElementXNum(CProj) + 1
        SElement(CProj, XE, ElementXNum(CProj)) = EleColRow(CProj, XE, I, J)
    Next J
Next I
Show_Element
''' Try output element conjunction
'Load frmOutText(CProj)
'frmOutText(CProj).Show
'frmOutText(CProj).Cls
'For J = 1 To (ElementXNum(CProj) + ElementYNum(CProj))
' CStart = EElement(CProj, J).Start
' CEnd = EElement(CProj, J).End
' frmOutText(CProj).Print "Start : End" + Format$(CStart) + " : " + F
ormat$(CEnd)
'Next J
Show_DataScreenE
End Sub

Sub Trans_UniformLoad (SUniform As Single)
Const XE = 0
Const YE = 1
For I = 1 To SCol(CProj)
    For J = 1 To ColEle(I)
        SLoad(CProj, YE, I, J).Type = 2
        SLoad(CProj, YE, I, J).Start = 0
        SLoad(CProj, YE, I, J).End = EleColRow(CProj, YE, I, J).Length
    Next J
Next I

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRANSDAT.BAS - 10

```

      SLoad(CProj, YE, I, J).ElementRef = J
      SLoad(CProj, YE, I, J).StartLoad = (1 / 2) * SUniform * (EleColRow(C
Proj, YE, I, J).L2 + EleColRow(CProj, YE, I, J).L1)
    Next J
  Next I
  For I = 1 To SRow(CProj)
    For J = 1 To RowEle(I)
      SLoad(CProj, XE, I, J).Type = 2
      SLoad(CProj, XE, I, J).Start = 0
      SLoad(CProj, XE, I, J).End = EleColRow(CProj, XE, I, J).Length
      SLoad(CProj, XE, I, J).ElementRef = J
      SLoad(CProj, XE, I, J).StartLoad = (1 / 2) * SUniform * (EleColRow(C
Proj, XE, I, J).L2 + EleColRow(CProj, XE, I, J).L1)
    Next J
  Next I
End Sub

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COMPUTE.BAS - 1

```

Sub BCal ()
  Dim I As Integer
  Cal_BEndMoment
  For I = 1 To ElementNum(CProj)
    RL(CProj, I) = Cal_BEndReaction(I, ML(CProj, I), MR(CProj, I), True)
    RR(CProj, I) = Cal_BEndReaction(I, ML(CProj, I), MR(CProj, I), False)
  Next I
End Sub

Function Cal_BBeta1 (emax As Single, eb As Single) As Single
  Cal_BBeta1 = emax / eb
End Function

Function Cal_BBeta2 (emax As Single, eb As Single) As Single
  Cal_BBeta2 = emax / eb
End Function

Function Cal_BCOF (E As Integer, LeftOrRight As Integer) As Single
  Dim N As Integer
  If LeftOrRight Then
    Cal_BCOF = (1 / 2)
    N = Element(CProj, E).Start
  ElseIf Not (LeftOrRight) Then
    Cal_BCOF = (1 / 2)
    N = Element(CProj, E).End
  End If
  If Node(CProj, N).SupportType = 3 Then
    Cal_BCOF = 0
  End If
End Function

Function Cal_BDeflection () As Single
End Function

Function Cal_BDF (E As Integer, StartOrEnd As Integer) As Single
  Dim I As Integer
  Dim SumDF As Single
  SumDF = 0
  If StartOrEnd Then
    For I = 0 To ElementNum(CProj)
      If (Element(CProj, E).Start = Element(CProj, I).Start) Or (Element(CProj, E).Start = Element(CProj, I).End) Then
        SumDF = SumDF + Cal_BK(I)
      End If
    Next I
  End If
  If Not (SumDF = 0) Then
    Cal_BDF = (Cal_BK(E) / SumDF)
  Else
    Cal_BDF = 1
  End If
  If Node(CProj, Element(CProj, E).Start).SupportType = 2 Then
    Cal_BDF = 0
  End If
  ElseIf Not (StartOrEnd) Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COMPUTE.BAS - 2

```

For I = 0 To ElementNum(CProj)
  If (Element(CProj, E).End = Element(CProj, I).Start) Or (Element(CProj, E).End = Element(CProj, I).End) Then
    SumDF = SumDF + Cal_BK(I)
  End If
Next I
If Not (SumDF = 0) Then
  Cal_BDF = (Cal_BK(E) / SumDF)
Else
  Cal_BDF = 1
End If
If Node(CProj, Element(CProj, E).End).SupportType = 2 Then
  Cal_BDF = 0
End If
End If
End Function

```

```

Sub Cal_BEndMoment ()
Const ErrorCheck = .01
Dim ErrorValue As Single
Dim I As Integer
ReDim FEML(ElementNum(CProj)) As Single
ReDim FEMR(ElementNum(CProj)) As Single
ReDim DFL(ElementNum(CProj)) As Single
ReDim DFR(ElementNum(CProj)) As Single
'ReDim ML(CProj, ElementNum(CProj)) As Single
'ReDim MR(CProj, ElementNum(CProj)) As Single
'ReDim COFL(CProj, ElementNum(CProj)) As Single
'ReDim COFR(CProj, ElementNum(CProj)) As Single
ReDim COML(ElementNum(CProj)) As Single
ReDim COMR(ElementNum(CProj)) As Single
ReDim DML(ElementNum(CProj)) As Single
ReDim DMR(ElementNum(CProj)) As Single
ReDim MNode(NodeNum(CProj)) As Single
For I = 1 To ElementNum(CProj)
  FEML(I) = Cal_BFEM(I, True)
  FEMR(I) = Cal_BFEM(I, False)
  DFL(I) = Cal_BDF(I, True)
  DFR(I) = Cal_BDF(I, False)
  COFL(CProj, I) = Cal_BCOF(I, True)
  COFR(CProj, I) = Cal_BCOF(I, False)
  ML(CProj, I) = FEML(I)
  MR(CProj, I) = FEMR(I)
  DML(I) = 0
  DMR(I) = 0
  COML(I) = 0
  COMR(I) = 0
Next I
'Distribution Moment
Do
ErrorValue = 0
'Sum DM
For I = 1 To NodeNum(CProj)
  MNode(I) = 0
  For J = 1 To ElementNum(CProj)
    If (Element(CProj, J).Start) = I Then
      MNode(I) = (MNode(I) + ML(CProj, J))
    End If
  Next J
Next I

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COMPUTE.BAS - 3

```

ElseIf (Element(CProj, J).End = I) Then
  MNode(I) = (MNode(I) + MR(CProj, J))
End If
Next J
Next I
'DM
For I = 1 To NodeNum(CProj)
  For J = 1 To ElementNum(CProj)
    If (Element(CProj, J).Start = I) Then
      DML(J) = (((-1) * MNode(I)) * DFL(J))
    ElseIf (Element(CProj, J).End = I) Then
      DMR(J) = (((-1) * MNode(I)) * DFR(J))
    End If
  Next J
Next I
For I = 1 To ElementNum(CProj)
  If Abs(ErrorValue) < Abs(DML(I)) Then
    ErrorValue = DML(I)
  End If
  If Abs(ErrorValue) < Abs(DMR(I)) Then
    ErrorValue = DMR(I)
  End If
Next I
'Carry over moment
For I = 1 To ElementNum(CProj)
  COML(I) = DMR(I) * COFL(CProj, I)
  COMR(I) = DML(I) * COFR(CProj, I)
Next I
'Sum Moment
If (Abs(ErrorValue) < ErrorCheck) Then
  For I = 1 To ElementNum(CProj)
    ML(CProj, I) = ML(CProj, I) + DML(I)
    MR(CProj, I) = MR(CProj, I) + DMR(I)
  Next I
Else
  For I = 1 To ElementNum(CProj)
    ML(CProj, I) = ML(CProj, I) + DML(I) + COML(I)
    MR(CProj, I) = MR(CProj, I) + DMR(I) + COMR(I)
  Next I
End If
Loop Until (Abs(ErrorValue) < ErrorCheck)
Load frmOutText(CProj)
frmOutText(CProj).Show
frmOutText(CProj).Cls
For I = 1 To ElementNum(CProj)
  frmOutText(CProj).Print "Fixed end moment Left element" + Format$(I)
+ " = " + Format$(Cal_BFEM(I, True), "0.00")
  frmOutText(CProj).Print "Fixed end moment Right element" + Format$(I)
+ " = " + Format$(Cal_BFEM(I, False), "0.00")
  frmOutText(CProj).Print "DF Factor Left " + Format$(DFL(I), "0.000")
  frmOutText(CProj).Print "DF Factor Right " + Format$(DFR(I), "0.000")
  frmOutText(CProj).Print "End Moment Left " + Format$(ML(CProj, I), "0
.00")
  frmOutText(CProj).Print "End Moment Right " + Format$(MR(CProj, I), "
0.00")
Next I
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COMPUTE.BAS - 4

Function Cal_BEndReaction (E As Integer, ML As Single, MR As Single, LeftOrRight As Integer) As Single

```
Dim RL As Single
Dim RR As Single
Dim M As Single
Dim p As Single
Dim W As Single
Dim W1 As Single
Dim W2 As Single
Dim a As Single
Dim b As Single
Dim L As Single
```

RL = 0

RR = 0

If LeftOrRight Then

For I = 1 To LoadNum(CProj)

If LoadData(CProj, I).ElementRef = E Then

If Node(CProj, Element(CProj, E).Start).SupportType = 3 Then

Select Case LoadData(CProj, I).Type

Case 1:

a = LoadData(CProj, I).Start

p = LoadData(CProj, I).StartLoad

L = Element(CProj, E).Length

RL = RL + p

Case 2:

a = LoadData(CProj, I).Start

b = LoadData(CProj, I).End

W1 = LoadData(CProj, I).StartLoad

L = Element(CProj, E).Length

RL = RL + (W1 * (b - a))

Case 3:

a = LoadData(CProj, I).Start

b = LoadData(CProj, I).End

W1 = LoadData(CProj, I).StartLoad

W2 = LoadData(CProj, I).EndLoad

L = Element(CProj, E).Length

RL = RL + ((1 / 2) * (W1 + W2) * (b - a))

Case 4:

L = Element(CProj, E).Length

M = LoadData(CProj, I).StartLoad

RL = RL

Case 5:

End Select

Else

Select Case LoadData(CProj, I).Type

Case 1:

a = LoadData(CProj, I).Start

p = LoadData(CProj, I).StartLoad

L = Element(CProj, E).Length

RL = RL + ((p * (L - a)) / L)

Case 2:

a = LoadData(CProj, I).Start

b = LoadData(CProj, I).End

W1 = LoadData(CProj, I).StartLoad

L = Element(CProj, E).Length

RL = RL + ((W * (b - a) * ((b - a) / 2) + L - b)) / L)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COMPUTE.BAS - 5

```

Case 3:
  a = LoadData(CProj, I).Start
  b = LoadData(CProj, I).End
  W1 = LoadData(CProj, I).StartLoad
  W2 = LoadData(CProj, I).EndLoad
  L = Element(CProj, E).Length
  RL = RL + (((1 / 2) * (W1 + W2) * (b - a)) * (L - b + ((1
/ 3) * W2)) / (W1 + W2))) / L
Case 4:
  L = Element(CProj, E).Length
  M = LoadData(CProj, I).StartLoad
  RL = RL + ((M / L) * (-1))
Case 5:
End Select
End If
End If
Next I
L = Element(CProj, E).Length
If Not (Node(CProj, Element(CProj, E).End).SupportType = 3) Then
  RL = RL + ((ML + MR) / L) * (-1)
End If
Cal_BEndReaction = RL
ElseIf Not (LeftOrRight) Then
For I = 1 To LoadNum(CProj)
  If LoadData(CProj, I).ElementRef = E Then
    If Node(CProj, Element(CProj, E).End).SupportType = 3 Then
      Select Case LoadData(CProj, I).Type
      Case 1:
        a = LoadData(CProj, I).Start
        p = LoadData(CProj, I).StartLoad
        L = Element(CProj, E).Length
        RR = RR + p
      Case 2:
        a = LoadData(CProj, I).Start
        b = LoadData(CProj, I).End
        W1 = LoadData(CProj, I).StartLoad
        L = Element(CProj, E).Length
        RR = RR + (W1 * (b - a))
      Case 3:
        a = LoadData(CProj, I).Start
        b = LoadData(CProj, I).End
        W1 = LoadData(CProj, I).StartLoad
        W2 = LoadData(CProj, I).EndLoad
        L = Element(CProj, E).Length
        RR = RR + ((1 / 2) * (W1 + W2) * (b - a))
      Case 4:
        L = Element(CProj, E).Length
        M = LoadData(CProj, I).StartLoad
        RR = RR + 0
      Case 5:
      End Select
    Else
      Select Case LoadData(CProj, I).Type
      Case 1:
        a = LoadData(CProj, I).Start
        p = LoadData(CProj, I).StartLoad
        L = Element(CProj, E).Length

```

COMPUTE.BAS - 6

```

    RR = RR + ((p * a) / L)
Case 2:
  a = LoadData(CProj, I).Start
  b = LoadData(CProj, I).End
  W1 = LoadData(CProj, I).StartLoad
  L = Element(CProj, E).Length
  RR = RR + ((W * (b - a) * (a + ((b - a) / 2))) / L)
Case 3:
  a = LoadData(CProj, I).Start
  b = LoadData(CProj, I).End
  W1 = LoadData(CProj, I).StartLoad
  W2 = LoadData(CProj, I).EndLoad
  L = Element(CProj, E).Length
  RR = RR + (((((1 / 2) * (W1 + W2) * (b - a) * (a + ((W2 + ((1 / 3
) * W1)) / (W1 + W2)))))) / L)
Case 4:
  L = Element(CProj, E).Length
  M = LoadData(CProj, I).StartLoad
  RR = RR + (M / L)
Case 5:
End Select
End If
End If
Next I
L = Element(CProj, E).Length
If Not (Node(CProj, Element(CProj, E).End).SupportType <> 3) Then
  RR = RR + ((ML + MR) / L)
End If
Cal BEndReaction = RR
End If
End Function

```

```

Function Cal_BFEM (E As Integer, LeftOrRight As Integer) As Single
Dim I As Integer
Dim a As Single
Dim b As Single
Dim C As Single
Dim L As Single
Dim W As Single
Dim W1 As Single
Dim W2 As Single
Dim M As Single
Dim FEML As Single
Dim FEMR As Single
FEML = 0
FEMR = 0
If LeftOrRight Then
For I = 1 To LoadNum(CProj)
If LoadData(CProj, I).ElementRef = E Then
If Node(CProj, Element(CProj, E).Start).SupportType = 3 Then
Select Case LoadData(CProj, I).Type
Case 1: 'Point Load
W = LoadData(CProj, I).StartLoad
a = LoadData(CProj, I).Start
b = Element(CProj, E).Length - LoadData(CProj, I).Start
L = Element(CProj, E).Length
FEML = FEML - (W * a)

```

COMPUTE.BAS - 7

Case 2: 'Uniform Load

a = LoadData(CProj, I).Start

b = LoadData(CProj, I).End

L = Element(CProj, E).Length

M = L - ((a + b) / 2)

W = (LoadData(CProj, I).StartLoad) * (b - a)

p = b - a

FEML = FEML - (W * ((a + b) / 2))

Case 3: 'Triangle Load

a = LoadData(CProj, I).Start

b = LoadData(CProj, I).End

W = (LoadData(CProj, I).StartLoad) * (b - a)

W1 = (1 / 2) * (LoadData(CProj, I).EndLoad - LoadData(CProj, I).S
tartLoad) * (b - a)

FEML = FEML - (W * ((a + b) / 2))

FEML = FEML - (W1 * (((2 * b) + a) / 3))

Case 4: 'Moment at Point

a = LoadData(CProj, I).Start

b = Element(CProj, E).Length - LoadData(CProj, I).End

L = Element(CProj, E).Length

M = LoadData(CProj, I).StartLoad

FEML = FEML + -M

Case 5: 'Function Load

End Select

Else

Select Case LoadData(CProj, I).Type

Case 1: 'Point Load

W = LoadData(CProj, I).StartLoad

a = LoadData(CProj, I).Start

b = Element(CProj, E).Length - LoadData(CProj, I).Start

L = Element(CProj, E).Length

FEML = FEML + (((W * a * (b ^ 2)) / (L ^ 2)) * (-1))

'FEMR = FEMR + ((W * (a ^ 2) * b) / (L ^ 2))

Case 2: 'Uniform Load

a = LoadData(CProj, I).Start

b = LoadData(CProj, I).End

L = Element(CProj, E).Length

M = L - ((a + b) / 2)

W = (LoadData(CProj, I).StartLoad) * (b - a)

p = b - a

FEML = FEML - (((W / (24 * L)) * (((24 * (M ^ 3)) / L) - ((6 * b * (p ^ 2)) / L) + ((3 * (p ^ 3)) / L) + (4 * (p ^ 2)) - (24 * (M ^ 2)))) * (-1)))

'FEMR = FEMR + (W / (24 * L)) * (((24 * (m ^ 3)) / L) - ((6 * b * (p ^ 2)) / L) + ((3 * (p ^ 3)) / L) + (2 * (p ^ 2)) - (48 * (m ^ 2)) + (24 * m * L)))

Case 3: 'Triangle Load

a = LoadData(CProj, I).Start

b = LoadData(CProj, I).End

L = Element(CProj, E).Length

M = L - ((a + b) / 2)

N = L - ((a + (2 * b)) / 3)

W = (LoadData(CProj, I).StartLoad) * (b - a)

W1 = (1 / 2) * (LoadData(CProj, I).EndLoad - LoadData(CProj, I).S
tartLoad) * (b - a)

p = b - a

FEML = FEML + (((W / (24 * L)) * (((24 * (M ^ 3)) / L) - ((6 * b

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COMPUTE.BAS - 8

```

* (p ^ 2) / L) + ((3 * (p ^ 3)) / L) + (4 * (p ^ 2)) - (24 * (M ^ 2))))
* (-1))
  FEML = FEML + ((W1 / L) * ((N ^ 3) / L) + ((p ^ 2) / 9) + ((51
* (p ^ 3)) / (810 * L)) - (((1 / 6) * (p ^ 2) * b) / L) - (N ^ 2)))) * (-1
)
  'FEMR = FEMR + (W / (24 * L)) * (((24 * (m ^ 3)) / L) - ((6 * b
* (p ^ 2) / L) + ((3 * (p ^ 3)) / L) + (2 * (p ^ 2)) - (48 * (m ^ 2)) + (
24 * m * L))) + ((W1 / L) * ((n ^ 3) / L) - ((p ^ 2) / 18) + ((51 * (p ^
3)) / (810 * L)) - (((1 / 6) * (p ^ 2) * b) / L) - (2 * (n ^ 2)) + (n * L
))

  Case 4: 'Moment at Point
  a = LoadData(CProj, I).Start
  b = Element(CProj, E).Length - LoadData(CProj, I).End
  L = Element(CProj, E).Length
  M = LoadData(CProj, I).StartLoad
  FEML = FEML + ((M * b) * ((3 * a) - L)) / (L ^ 2)
  'FEMR = FEMR + ((m * a) * ((3 * b) - L)) / (L ^ 2)
  Case 5: 'Function Load
  End Select
End If
End If
Next I
Cal_BFEM = FEML
ElseIf Not (LeftOrRight) Then
  For I = 1 To LoadNum(CProj)
  If LoadData(CProj, I).ElementRef = E Then
  If Node(CProj, Element(CProj, E).End).SupportType = 3 Then
  Select Case LoadData(CProj, I).Type
  Case 1: 'Point Load
  W = LoadData(CProj, I).StartLoad
  a = Element(CProj, E).Length - LoadData(CProj, I).Start
  L = Element(CProj, E).Length
  FEML = FEML + (W * a)
  Case 2: 'Uniform Load
  a = Element(CProj, E).Length - LoadData(CProj, I).End
  b = Element(CProj, E).Length - LoadData(CProj, I).Start
  L = Element(CProj, E).Length
  M = L - ((a + b) / 2)
  W = (LoadData(CProj, I).StartLoad) * (b - a)
  FEMR = FEMR + (W * ((a + b) / 2))
  Case 3: 'Triangle Load
  a = Element(CProj, E).Length - LoadData(CProj, I).End
  b = Element(CProj, E).Length - LoadData(CProj, I).Start
  L = Element(CProj, E).Length
  W = (LoadData(CProj, I).EndLoad) * (b - a)
  W1 = (1 / 2) * (LoadData(CProj, I).StartLoad - LoadData(CProj, I)
.EndLoad) * (b - a)
  FEMR = FEMR - (W * ((a + b) / 2))
  FEMR = FEMR - (W1 * (((2 * b) + a) / 3))
  Case 4: 'Moment at Point
  a = Element(CProj, E).Length - LoadData(CProj, I).Start
  M = LoadData(CProj, I).StartLoad
  FEMR = FEMR - M
  Case 5: 'Function Load
  End Select
Else
  Select Case LoadData(CProj, I).Type

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COMPUTE.BAS - 9

Case 1: 'Point Load

```
W = LoadData(CProj, I).StartLoad
a = LoadData(CProj, I).Start
b = Element(CProj, E).Length - LoadData(CProj, I).Start
L = Element(CProj, E).Length
'FEML = FEML + ((W * a * (b ^ 2)) / (L ^ 2)) * (-1)
FEMR = FEMR + ((W * (a ^ 2) * b) / (L ^ 2))
```

Case 2: 'Uniform Load

```
a = LoadData(CProj, I).Start
b = LoadData(CProj, I).End
L = Element(CProj, E).Length
M = L - ((a + b) / 2)
W = (LoadData(CProj, I).StartLoad) * (b - a)
p = b - a
'FEML = FEML - (W / (24 * L)) * (((24 * (m ^ 3)) / L) - ((6 * b
* (p ^ 2)) / L) + ((3 * (p ^ 3)) / L) + (4 * (p ^ 2)) - (24 * (m ^ 2))))
FEMR = FEMR + ((W / (24 * L)) * (((24 * (M ^ 3)) / L) - ((6 * b
* (p ^ 2)) / L) + ((3 * (p ^ 3)) / L) + (2 * (p ^ 2)) - (48 * (M ^ 2)) + (
24 * M * L))))
```

Case 3: 'Triangle Load

```
a = LoadData(CProj, I).Start
b = LoadData(CProj, I).End
L = Element(CProj, E).Length
M = L - ((a + b) / 2)
N = L - ((a + (2 * b)) / 3)
W = (LoadData(CProj, I).StartLoad) * (b - a)
W1 = (1 / 2) * (LoadData(CProj, I).EndLoad - LoadData(CProj, I).S
tartLoad) * (b - a)
p = b - a
'FEML = FEML - (W / (24 * L)) * (((24 * (m ^ 3)) / L) - ((6 * b
* (p ^ 2)) / L) + ((3 * (p ^ 3)) / L) + (4 * (p ^ 2)) - (24 * (m ^ 2))) +
((W1 / L) * (((n ^ 3) / L) + ((p ^ 2) / 9) + ((51 * (p ^ 3)) / (810 * L))
- (((1 / 6) * (p ^ 2) * b) / L) - (n ^ 2)))
FEMR = FEMR + (W / (24 * L)) * (((24 * (M ^ 3)) / L) - ((6 * b *
(p ^ 2)) / L) + ((3 * (p ^ 3)) / L) + (2 * (p ^ 2)) - (48 * (M ^ 2)) + (2
4 * M * L)))
FEMR = FEMR + ((W1 / L) * (((N ^ 3) / L) - ((p ^ 2) / 18) + ((51
* (p ^ 3)) / (810 * L)) - (((1 / 6) * (p ^ 2) * b) / L) - (2 * (N ^ 2)) +
(N * L)))
```

Case 4: 'Moment at Point

```
a = LoadData(CProj, I).Start
b = Element(CProj, E).Length - LoadData(CProj, I).End
L = Element(CProj, E).Length
M = LoadData(CProj, I).StartLoad
'FEML = FEML + ((m * b) * ((3 * a) - L)) / (L ^ 2)
FEMR = FEMR + ((M * a) * ((3 * b) - L)) / (L ^ 2)
```

Case 5: 'Function Load

End Select

End If

End If

Next I

.Cal_BFEM = FEMR

End If

End Function

Function Cal_BK (E As Integer) As Single

COMPUTE.BAS - 10

```

Dim Ec As Single
Ec = GeneralData(CProj).SEc
Cal BK = ((4 * Ec * Cal_BSecondInertia(E)) / Element(CProj, E).Length)
If (Node(CProj, Element(CProj, E).Start).SupportType = 3) Or (Node(CPr
oj, Element(CProj, E).End).SupportType = 3) Then
    Cal BK = 0
End If
End Function

```

```

Function Cal_BMoment (E As Integer, x As Single, ML As Single, MR As Singl
e) As Single

```

```

Dim M As Single
Dim p As Single
Dim W As Single
Dim W1 As Single
Dim W2 As Single
Dim a As Single
Dim b As Single
Dim L As Single
Dim Mx As Single
Dim Wx As Single

```

```
'RL = 0
```

```
'RR = 0
```

```
Mx = 0
```

```
For I = 1 To LoadNum(CProj)
```

```
  If LoadData(CProj, I).ElementRef = E Then
```

```
    Select Case LoadData(CProj, I).Type
```

```
      Case 1:
```

```
        a = LoadData(CProj, I).Start
```

```
        p = LoadData(CProj, I).StartLoad
```

```
        L = Element(CProj, E).Length
```

```
        If x <= a Then
```

```
          Mx = Mx + (((p * (L - a)) / L) * x)
```

```
        End If
```

```
        If x > a Then
```

```
          Mx = Mx + (((p * a) / L) * (L - x))
```

```
        End If
```

```
      Case 2:
```

```
        a = LoadData(CProj, I).Start
```

```
        b = LoadData(CProj, I).End
```

```
        W = LoadData(CProj, I).StartLoad
```

```
        L = Element(CProj, E).Length
```

```
        If x <= a Then
```

```
          Mx = Mx + (((W * (b - a) * (L - b + ((b - a) / 2))) / L) * x)
```

```
        End If
```

```
        If (x > a) And (x < b) Then
```

```
          Mx = Mx + (((W * (b - a) * (L - b + ((b - a) / 2))) / L) * x) - (
```

```
W * (x - a) * (1 / 2) * (x - a)))
```

```
        End If
```

```
        If x >= b Then
```

```
          Mx = Mx + (((W * (b - a) * (a + ((b - a) / 2))) / L) * (L - x))
```

```
        End If
```

```
      Case 3:
```

```
        a = LoadData(CProj, I).Start
```

```
        b = LoadData(CProj, I).End
```

```
        W1 = LoadData(CProj, I).StartLoad
```

```
        W2 = LoadData(CProj, I).EndLoad
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COMPUTE.BAS - 11

```

L = Element(CProj, E).Length
Wx = (W1 + (((W2 - W1) * (x - a)) / (b - a)))
If x <= a Then
  Mx = Mx + (((((1 / 2) * (W1 + W2) * (b - a)) * (L - b + (((W1 + ((
1 / 3) * W2)) / (W1 + W2))) * (b - a))) / L) * x)
ElseIf (x > a) And (x < b) Then
  Mx = Mx + (((((1 / 2) * (W1 + W2) * (b - a)) * (L - b + (((W1 + ((
1 / 3) * W2)) / (W1 + W2))) * (b - a))) / L) * x)
  Mx = Mx - ((1 / 2) * (W1 + Wx) * (b - a) * (((W1 + ((1 / 3) * W2))
/ (W1 + W2))) * (x - a)
ElseIf (x > b) Then
  Mx = Mx + (((((1 / 2) * (W1 + W2) * (b - a)) * (a + (((W2 + ((1 /
3) * W1)) / (W1 + W2))) * (b - a))) / L) * (L - x))
End If
Case 4:
  a = LoadData(CProj, I).Start
  M = LoadData(CProj, I).StartLoad
  L = Element(CProj, E).Length
  If x <= a Then
    Mx = Mx - ((M / L) * x)
  ElseIf x > a Then
    Mx = Mx + ((M / L) * (L - x))
  End If
Case 5:
End Select
End If
Next I
L = Element(CProj, E).Length
Mx = Mx - (((ML + MR) / L) * x) + ML
Cal_BMoment = Mx
End Function

```

```

Function Cal_BSecondInertia (E As Integer) As Single
  Dim b As Single
  Dim d As Single
  Select Case Section(CProj, E).Type
  Case 1:
    b = Section(CProj, E).b
    d = Section(CProj, E).d
    Cal_BSecondInertia = (1 / 12) * b * (d ^ 3)
  Case 2:
  Case 3:
  Case 4:
  End Select
End Function

```

```

Function Cal_BShear (E As Integer, x As Single) As Single
  Dim M As Single
  Dim p As Single
  Dim W As Single
  Dim W1 As Single
  Dim W2 As Single
  Dim Wx As Single
  Dim a As Single
  Dim b As Single
  Dim L As Single

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COMPUTE.BAS - 12

```

Dim V As Single
V = 0
For I = 1 To LoadNum(CProj)
  If LoadData(CProj, I).ElementRef = E Then
    Select Case LoadData(CProj, I).Type
    Case 1:
      a = LoadData(CProj, I).Start
      p = LoadData(CProj, I).StartLoad
      L = Element(CProj, E).Length
      If x <= a Then
        V = V + ((p * (L - a)) / L)
      End If
      If x > a Then
        V = V - ((p * a) / L)
      End If
    Case 2:
      a = LoadData(CProj, I).Start
      b = LoadData(CProj, I).End
      W = LoadData(CProj, I).StartLoad
      L = Element(CProj, E).Length
      If x <= a Then
        V = V + (W * (b - a) * (L - b + ((b - a) / 2))) / L
      End If
      If (x > a) And (x < b) Then
        V = V + ((W * (b - a) * (L - b + ((b - a) / 2))) / L) - (W * (x - a
)))
      End If
      If x >= b Then
        V = V - ((W * (b - a) * (a + ((b - a) / 2))) / L)
      End If
    Case 3:
      a = LoadData(CProj, I).Start
      b = LoadData(CProj, I).End
      W1 = LoadData(CProj, I).StartLoad
      W2 = LoadData(CProj, I).EndLoad
      Wx = W1 + (((W2 - W1) * (x - a)) / (b - a))
      L = Element(CProj, E).Length
      If x <= a Then
        V = V + ((1 / 2) * (W1 + W2) * (b - a) * (L - b + ((b - a) * (W1 +
((1 / 3) * W2)) / (W1 + W2)))) / L)
      End If
      If (x > a) And (x < b) Then
        V = V + (((1 / 2) * (W1 + W2) * (b - a) * (L - b + ((b - a) * ((W
1 + ((1 / 3) * W2)) / (W1 + W2)))) / L) - ((1 / 2) * (Wx + W1) * (x - a))
)
      End If
      If (x >= b) Then
        V = V + (((1 / 2) * (W1 + W2) * (b - a) * (L - b + (W1 + ((1 / 3)
* W2)) / (W1 + W2)) / L) - ((1 / 2) * (b - a) * (W1 + W2)))
      End If
    Case 4:
      a = LoadData(CProj, I).Start
      M = LoadData(CProj, I).StartLoad
      L = Element(CProj, E).Length
      If x <= a Then
        V = V - (M / L)
      End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COMPUTE.BAS ~ 13

```

    If x > a Then
      V = V - (M / L)
    End If
    Case 5:
    End Select
  End If
Next I
Cal_BShear = V - ((ML(CProj, E) + MR(CProj, E)) / Element(CProj, E).Length)
End Function

Function Fc_ToEc (fc_) As Single
  Fc_ToEc = 4730 * Sqr(fc_)
End Function

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CRTDATA.BAS - 1

```
Sub Action_Icon (Index As Integer, Enabled As Integer)
```

```
Select Case Enabled
```

```
Case True:
```

```
Select Case Index
```

```
Case 0:
```

```
frmMain.imgTool1(Index).Enabled = True
```

```
Case 1:
```

```
frmMain.imgTool1(Index).Enabled = True
```

```
Case 2:
```

```
frmMain.imgTool1(Index).Enabled = True
```

```
Case 3:
```

```
frmMain.imgTool1(Index).Enabled = True
```

```
Case 4:
```

```
frmMain.imgTool1(Index).Enabled = True
```

```
Case 5:
```

```
frmMain.imgTool1(Index).Enabled = True
```

```
Case 6:
```

```
frmMain.imgTool1(Index).Enabled = True
```

```
Case 7:
```

```
frmMain.imgTool1(Index).Enabled = True
```

```
End Select
```

```
Case False:
```

```
Select Case Index
```

```
Case 0:
```

```
frmMain.imgTool1(Index).Enabled = False
```

```
Case 1:
```

```
frmMain.imgTool1(Index).Enabled = False
```

```
Case 2:
```

```
frmMain.imgTool1(Index).Enabled = False
```

```
Case 3:
```

```
frmMain.imgTool1(Index).Enabled = False
```

```
Case 4:
```

```
frmMain.imgTool1(Index).Enabled = False
```

```
Case 5:
```

```
frmMain.imgTool1(Index).Enabled = False
```

```
Case 6:
```

```
frmMain.imgTool1(Index).Enabled = False
```

```
Case 7:
```

```
frmMain.imgTool1(Index).Enabled = False
```

```
End Select
```

```
End Select
```

```
End Sub
```

```
Sub Do_Tool2 (ColNum As Integer, RowNum As Integer)
```

```
Const Span = 5
```

```
Const TwipsPerMeter = 100
```

```
Const I_X = 1000
```

```
Const I_Y = 1000
```

```
For I = 1 To ColNum
```

```
For J = 1 To RowNum
```

```
Column(CProj, (I - 1) * J + J).Co_X = Span * (I - 1) * TwipsPerMeter
```

```
Column(CProj, (I - 1) * J + J).Co_Y = Span * (J - 1) * TwipsPerMeter
```

```
'frmDataInfo(CProj).Line (I_X + 0 + Column(CProj, (I - 1) * J + J).C
```

```
o_X, I_Y + 0 + Column(CProj, (I - 1) * J + J).Co_Y)-(I_X + 50 + Column(CPr
```

```
oJ, (I - 1) * J + J).Co_X, I_Y + 50 + Column(CProj, (I - 1) * J + J).Co_Y)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CRTDATA.BAS - 2

```
, ColumnColor, BF
  Next J
  Next I
End Sub
```

```
Sub Draw_SNode (ArrX As Integer, ArrY As Integer, Node As Integer)
  Const I_X = 1000
  Const I_Y = 1000
  For I = 0 To ArrY - 1
    For J = 0 To ArrX - 1
      'frmData(CProj).Line (I_X + 0 + Column(CProj, Current + I * J + J).C
o_X, I_Y + 0 + Column(CProj, I + J + J).Co_Y)-(I_X + 50 + Column(CProj, I
* J + J).Co_X, I_Y + 50 + Column(CProj, I * J + J).Co_Y), ColumnColor, BF
    Next J
  Next I
Exit Sub
End Sub
```

```
Sub Initial_Menu (Index As Integer)
  Select Case Index
  Case 1: 'initial in enter program
    'file
    frmMain.mnuFile_New.Enabled = True
    frmMain.mnuFile_Open.Enabled = True
    frmMain.mnuFile_Save.Enabled = False
    frmMain.mnuFile_SaveAs.Enabled = False
    frmMain.mnuFile_Print_Prestress_Profile.Enabled = False
    frmMain.mnuFile_Print_Prestress_LProfile.Enabled = False
    frmMain.mnuFile_Print_Prestress_LCompute.Enabled = False
    frmMain.mnuFile_Print_Flat_LCompute.Enabled = False
    frmMain.mnuFile_Print_Geometric.Enabled = False
    frmMain.mnuFile_Print_Shear.Enabled = False
    frmMain.mnuFile_Print_Moment.Enabled = False
    'input
    frmMain.mnuInput_Node.Enabled = False
    frmMain.mnuInput_Element.Enabled = False
    frmMain.mnuInput_Load.Enabled = False
    frmMain.mnuInput_Material.Enabled = False
    frmMain.mnuInput_Information = False
    'Design
    frmMain.mnuDesign_Prestress.Enabled = False
    frmMain.mnuDesign_Flat.Enabled = False
    'Result
    frmMain.mnuResult_Prestress_Profile.Enabled = False
    frmMain.mnuResult_Prestress_LProfile.Enabled = False
    frmMain.mnuResult_Prestress_LCompute.Enabled = False
    frmMain.mnuResult_Flat_LCompute.Enabled = False
    frmMain.mnuResult_Geometric.Enabled = False
    frmMain.mnuResult_Shear.Enabled = False
    frmMain.mnuResult_Moment.Enabled = False
    'Option
    frmMain.mnuOption_SaveSetting.Enabled = True
    'Window
    frmMain.mnuWindow_ArrangeIcon.Enabled = True
    frmMain.mnuWindow_Close.Enabled = False
    'Help
    frmMain.mnuHelp_System.Enabled = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CRTDATA.BAS - 3

```

frmMain.mnuHelp_About.Enabled = True
'menu icon
Action_Icon 0, True
Action_Icon 1, True
Action_Icon 2, False
Action_Icon 3, False
Action_Icon 4, False
Action_Icon 5, False
Action_Icon 6, False
Action_Icon 7, False
Case 2: 'Open or new file
'file
frmMain.mnuFile_New.Enabled = True
frmMain.mnuFile_Open.Enabled = True
frmMain.mnuFile_Save.Enabled = True
frmMain.mnuFile_SaveAs.Enabled = True
frmMain.mnuFile_Print_Prestress_Profile.Enabled = False
frmMain.mnuFile_Print_Prestress_LProfile.Enabled = False
frmMain.mnuFile_Print_Prestress_LCompute.Enabled = False
frmMain.mnuFile_Print_Flat_LCompute.Enabled = False
frmMain.mnuFile_Print_Geometric.Enabled = False
frmMain.mnuFile_Print_Shear.Enabled = False
frmMain.mnuFile_Print_Moment.Enabled = False
'input
frmMain.mnuInput_Node.Enabled = True
frmMain.mnuInput_Element.Enabled = True
frmMain.mnuInput_Load.Enabled = True
frmMain.mnuInput_Material.Enabled = True
frmMain.mnuInput_Information = True
'Design
frmMain.mnuDesign_Prestress.Enabled = True
frmMain.mnuDesign_Flat.Enabled = True
'Result
frmMain.mnuResult_Prestress_Profile.Enabled = False
frmMain.mnuResult_Prestress_LProfile.Enabled = False
frmMain.mnuResult_Prestress_LCompute.Enabled = False
frmMain.mnuResult_Flat_LCompute.Enabled = False
frmMain.mnuResult_Geometric.Enabled = False
frmMain.mnuResult_Shear.Enabled = False
frmMain.mnuResult_Moment.Enabled = False
'Option
frmMain.mnuOption_SaveSetting.Enabled = True
'Window
frmMain.mnuWindow_ArrangeIcon.Enabled = True
frmMain.mnuWindow_Close.Enabled = True
'Help
frmMain.mnuHelp_System.Enabled = True
frmMain.mnuHelp_About.Enabled = True
'menu icon
Action_Icon 0, True
Action_Icon 1, True
Action_Icon 2, True
Action_Icon 3, True
Action_Icon 4, True
Action_Icon 5, True
Action_Icon 6, False
Action_Icon 7, False

```

CRTDATA.BAS - 4

Case 3: 'Prestress design

```

'file
frmMain.mnuFile_New.Enabled = True
frmMain.mnuFile_Open.Enabled = True
frmMain.mnuFile_Save.Enabled = True
frmMain.mnuFile_SaveAs.Enabled = True
frmMain.mnuFile_Print_Prestress_Profile.Enabled = True
frmMain.mnuFile_Print_Prestress_LProfile.Enabled = True
frmMain.mnuFile_Print_Prestress_LCompute.Enabled = True
frmMain.mnuFile_Print_Flat_LCompute.Enabled = False
frmMain.mnuFile_Print_Geometric.Enabled = True
frmMain.mnuFile_Print_Shear.Enabled = True
frmMain.mnuFile_Print_Moment.Enabled = True
'input
frmMain.mnuInput_Node.Enabled = True
frmMain.mnuInput_Element.Enabled = True
frmMain.mnuInput_Load.Enabled = True
frmMain.mnuInput_Material.Enabled = True
frmMain.mnuInput_Information = True
'Design
frmMain.mnuDesign_Prestress.Enabled = True
frmMain.mnuDesign_Flat.Enabled = True
'Result
frmMain.mnuResult_Prestress_Profile.Enabled = True
frmMain.mnuResult_Prestress_LProfile.Enabled = True
frmMain.mnuResult_Prestress_LCompute.Enabled = True
frmMain.mnuResult_Flat_LCompute.Enabled = False
frmMain.mnuResult_Geometric.Enabled = True
frmMain.mnuResult_Shear.Enabled = True
frmMain.mnuResult_Moment.Enabled = True
'Option
frmMain.mnuOption_SaveSetting.Enabled = True
'Window
frmMain.mnuWindow_ArrangeIcon.Enabled = True
frmMain.mnuWindow_Close.Enabled = True
'Help
frmMain.mnuHelp_System.Enabled = True
frmMain.mnuHelp_About.Enabled = True
'menu icon
Action_Icon 0, True
Action_Icon 1, True
Action_Icon 2, True
Action_Icon 3, True
Action_Icon 4, True
Action_Icon 5, True
Action_Icon 6, True
Action_Icon 7, True

```

Case 4: 'Flat design

```

'file
frmMain.mnuFile_New.Enabled = True
frmMain.mnuFile_Open.Enabled = True
frmMain.mnuFile_Save.Enabled = True
frmMain.mnuFile_SaveAs.Enabled = True
frmMain.mnuFile_Print_Prestress_Profile.Enabled = False
frmMain.mnuFile_Print_Prestress_LProfile.Enabled = False
frmMain.mnuFile_Print_Prestress_LCompute.Enabled = False
frmMain.mnuFile_Print_Flat_LCompute.Enabled = True

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CRTDATA.BAS - 5

```

frmMain.mnuFile_Print_Geometric.Enabled = True
frmMain.mnuFile_Print_Shear.Enabled = True
frmMain.mnuFile_Print_Moment.Enabled = True
'input
frmMain.mnuInput_Node.Enabled = True
frmMain.mnuInput_Element.Enabled = True
frmMain.mnuInput_Load.Enabled = True
frmMain.mnuInput_Material.Enabled = True
frmMain.mnuInput_Information = True
'Design
frmMain.mnuDesign_Prestress.Enabled = True
frmMain.mnuDesign_Flat.Enabled = True
'Result
frmMain.mnuResult_Prestress_Profile.Enabled = False
frmMain.mnuResult_Prestress_LProfile.Enabled = False
frmMain.mnuResult_Prestress_LCompute.Enabled = False
frmMain.mnuResult_Flat_LCompute.Enabled = True
frmMain.mnuResult_Geometric.Enabled = True
frmMain.mnuResult_Shear.Enabled = True
frmMain.mnuResult_Moment.Enabled = True
'Option
frmMain.mnuOption_SaveSetting.Enabled = True
'Window
frmMain.mnuWindow_ArrangeIcon.Enabled = True
frmMain.mnuWindow_Close.Enabled = True
'Help
frmMain.mnuHelp_System.Enabled = True
frmMain.mnuHelp_About.Enabled = True
'menu icon
Action_Icon 0, True
Action_Icon 1, True
Action_Icon 2, True
Action_Icon 3, True
Action_Icon 4, True
Action_Icon 5, True
Action_Icon 6, True
Action_Icon 7, True
End Select
End Sub

Sub Show_DataScreen ()
Dim TPMX As Integer
Dim TPMY As Integer
Dim XMax As Integer
Dim XMin As Integer
Dim YMax As Integer
Dim YMin As Integer
Dim InitX As Integer
Dim InitY As Integer
Dim Col As Integer
Dim CStart As Integer
Dim CEnd As Integer

frmData(CProj).Cls
Col = 50
InitX = 1500
InitY = 1000

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CRTDATA.BAS - 6

```

XMax = Column(CProj, 1).Co_X
XMin = Column(CProj, 1).Co_X
YMax = Column(CProj, 1).Co_Y
YMin = Column(CProj, 1).Co_Y
For I = 1 To SNodeNum(CProj)
  If XMax < Column(CProj, I).Co_X Then
    XMax = Column(CProj, I).Co_X
  End If
  If XMin > Column(CProj, I).Co_X Then
    XMin = Column(CProj, I).Co_X
  End If
  If YMax < Column(CProj, I).Co_Y Then
    YMax = Column(CProj, I).Co_Y
  End If
  If YMin > Column(CProj, I).Co_Y Then
    YMin = Column(CProj, I).Co_Y
  End If
Next I
TPMX = (frmData(CProj).Width - 3000) \ (XMax - XMin)
TPMY = (frmData(CProj).Height - 2000) \ (YMax - YMin)
If TPMX < TPMY Then
  TPMY = TPMX
Else
  TPMX = TPMY
End If
For I = 1 To SNodeNum(CProj)
  frmData(CProj).Line (((Column(CProj, I).Co_X - MinX) * TPMX) - Col +
  InitX, ((Column(CProj, I).Co_Y - MinY) * TPMY) - Col + InitY)-(((Column(CP
roj, I).Co_X - MinX) * TPMX) + Col + InitX, ((Column(CProj, I).Co_Y - MinY
) * TPMY) + Col + InitY), ColumnColor, BF
  frmData(CProj).Print "[" + Format$(Column(CProj, I).Co_X, "#.00") + "
," + Format$(Column(CProj, I).Co_Y, "#.00") + "]"
Next I
End Sub

```

```

Sub Show_DataScreenE ()
Dim TPMX As Integer
Dim TPMY As Integer
Dim XMax As Integer
Dim XMin As Integer
Dim YMax As Integer
Dim YMin As Integer
Dim InitX As Integer
Dim InitY As Integer
Dim Col As Integer
Dim CStart As Integer
Dim CEnd As Integer

```

```

ElementColor = 30
ColumnColor = 34

```

```

frmOutGraph(CProj).Cls
Col = 50
InitX = 1500
InitY = 1000
XMax = Column(CProj, 1).Co_X
XMin = Column(CProj, 1).Co_X

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CRTDATA.BAS - 7

```

YMax = Column(CProj, 1).Co_Y
YMin = Column(CProj, 1).Co_Y
For I = 1 To SNodeNum(CProj)
  If XMax < Column(CProj, I).Co_X Then
    XMax = Column(CProj, I).Co_X
  End If
  If XMin > Column(CProj, I).Co_X Then
    XMin = Column(CProj, I).Co_X
  End If
  If YMax < Column(CProj, I).Co_Y Then
    YMax = Column(CProj, I).Co_Y
  End If
  If YMin > Column(CProj, I).Co_Y Then
    YMin = Column(CProj, I).Co_Y
  End If
Next I
TPMX = (frmOutGraph(CProj).Width - 3000) \ (XMax - XMin)
TPMY = (frmOutGraph(CProj).Height - 2000) \ (YMax - YMin)
If TPMX < TPMY Then
  TPMY = TPMX
Else
  TPMX = TPMY
End If
For I = 1 To SNodeNum(CProj)
  frmOutGraph(CProj).Line (((Column(CProj, I).Co_X - MinX) * TPMX) - Co
1 + InitX, ((Column(CProj, I).Co_Y - MinY) * TPMY) - Col + InitY)-(((Column
n(CProj, I).Co_X - MinX) * TPMX) + Col + InitX, ((Column(CProj, I).Co_Y -
MinY) * TPMY) + Col + InitY), ColumnColor, BF
  frmOutGraph(CProj).Print Format$(I, "@@@") + "[" + Format$(Column(CPr
oj, I).Co_X, "#.00") + "," + Format$(Column(CProj, I).Co_Y, "#.00") + "]"
Next I
''' Draw Element Conjunction in X axis
For I = 1 To SRow(CProj)
  For J = 1 To RowEle(I)
    CStart = EleColRow(CProj, 0, I, J).Start
    CEnd = EleColRow(CProj, 0, I, J).End
    X1 = ((Column(CProj, CStart).Co_X - MinX) * TPMX) + InitX
    Y1 = ((Column(CProj, CStart).Co_Y - MinY) * TPMY) + InitY
    X2 = ((Column(CProj, CEnd).Co_X - MinX) * TPMX) + InitX
    Y2 = ((Column(CProj, CEnd).Co_Y - MinY) * TPMY) + InitY
    frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(ElementColor)
  Next J
Next I
''' Draw Element Conjunction in Y axis
For I = 1 To SCol(CProj)
  For J = 1 To ColEle(I)
    CStart = EleColRow(CProj, 1, I, J).Start
    CEnd = EleColRow(CProj, 1, I, J).End
    X1 = ((Column(CProj, CStart).Co_X - MinX) * TPMX) + InitX
    Y1 = ((Column(CProj, CStart).Co_Y - MinY) * TPMY) + InitY
    X2 = ((Column(CProj, CEnd).Co_X - MinX) * TPMX) + InitX
    Y2 = ((Column(CProj, CEnd).Co_Y - MinY) * TPMY) + InitY
    frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(ElementColor)
  Next J
Next I
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FCOMPUTE.BAS - 1

```

Function Cal_Fb0 (Column As ColumnType, Index As Integer) As Single
  Dim Cal_Fb0t As Single
  Dim c1 As Single
  Dim c2 As Single
  Dim ed As Single
  Dim ed1 As Single
  Dim ed2 As Single
  ed1 = d(CProj)
  ed2 = Column.DropThick + ed1
  ed = ed1
  If Index = 0 Then 'none
    c1 = Column.c1
    c2 = Column.c2
    ed = ed1
  ElseIf Index = 1 Then 'Drop panel
    c1 = Column.DropX
    c2 = Column.DropY
    ed = ed1
  ElseIf Index = 2 Then 'Drop Capital
    c1 = Column.CapitalX
    c2 = Column.CapitalY
    ed = ed2
  End If
  'Interior
  If (Column.X_Flag + Column.Y_Flag) = 0 Then
    Cal_Fb0t = (c1 + ed + c2 + ed) * 2
  'Corner
  ElseIf (Column.X_Flag <> 0) And (Column.Y_Flag <> 0) Then
    Cal_Fb0t = (c1 + ed + c2)
  'Edge Y
  ElseIf (Column.X_Flag = 0) And ((Column.Y_Flag = 1) Or (Column.Y_Flag
= 2)) Then
    Cal_Fb0t = (c1 + ed + (2 * c2) + ed)
  'Edge X
  ElseIf (Column.Y_Flag = 0) And ((Column.X_Flag = 1) Or (Column.X_Flag
= 2)) Then
    Cal_Fb0t = ((c1 * 2) + ed + c2 + ed)
  End If
  Cal_Fb0 = Cal_Fb0t
End Function

Function Cal_FColumnK (CN As Integer, XY As Integer, h As Single) As Singl
e 'h is thickness
  Dim Kec As Single
  Dim Kt As Single
  Dim Kcc As Single
  Dim Kc As Single
  Dim a As Single
  Dim b As Single
  Dim R As Single
  Dim s As Single
  Dim c As Single
  Dim d As Single
  Dim Y_ As Single
  Dim X2 As Single

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FCOMPUTE.BAS - 2

```

Dim CC1 As Single 'length of column as x axis
Dim CC2 As Single 'length of column as y axis
Dim CDx As Single 'Drop x length
Dim CDy As Single 'Drop y Length
Dim CDt As Single 'Drop Thick
Dim CCx As Single 'Capital x Length
Dim CCy As Single 'Capital y Length
Dim CCL As Single 'Column length
Dim Cs As Single
Dim Cd As Single
Dim C3 As Single
Dim C4 As Single
Dim C5 As Single
Dim C6 As Single
Dim C7 As Single
Dim FA As Single
Dim FB As Single
Dim FR As Single
Dim FC As Single
Dim FD As Single
Dim CR As Single
ReDim SColRow(MaxFileNum) As Integer
lc = Column(CProj, CN).Length
If XY Then
  SColRow(CProj) = ElementXNum(CProj)
Else
  SColRow(CProj) = ElementYNum(CProj)
End If
For I = 1 To SColRow(CProj) 'find transverse length
  If (SElement(CProj, XY * (-1), I).Start = Element(CProj, E).CStart) Or
  r (SElement(CProj, XY * (-1), E).End = Element(CProj, E).CStart) Then
    L2 = (SElement(CProj, XY * (-1), I).Length / 2)
  End If
Next I
If XY Then 'Along x axis
  CC1 = Column(CProj, CN).c1
  CC2 = Column(CProj, CN).c2
  CDx = Column(CProj, CN).DropX
  CDy = Column(CProj, CN).DropY
  CDt = Column(CProj, CN).DropThick
  CCx = Column(CProj, CN).CapitalX
  CCy = Column(CProj, CN).CapitalY
  CCL = Column(CProj, CN).Length
  a = ((h + CDt) / 2) / lc
  b = 0
  R = 0
  s = 0
  c = 0
  d = (1 - a)
  FA = 0
  FB = 1
  FR = 1
  FC = 1
  FD = 0
  C3 = (((-FB) * a * a * (3 - (2 * a)) * (1 - FA)) - ((b * b) * (3 - (2
* b)) * (1 - FB)) + ((R ^ 2) * (3 - (2 * R)) * (1 - FR)) - ((s ^ 2) * (3
- (2 * s)) * (1 - FR))) + ((c ^ 2) * (3 - (2 * c)) * (1 - FC)) + ((FC * d *

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FCOMPUTE.BAS - 3

```

d) * (3 - (2 * d)) * (1 - FD)) + (FC * FD)) / 6
C4 = (((-FB * (a ^ 3)) * (1 - FA)) - ((b ^ 3) * (1 - FB)) + ((R ^ 3)
* (1 - FR)) - ((s ^ 3) * (1 - FR)) + ((c ^ 3) * (1 - FC)) + ((FC * (d ^ 3)
) * (1 - FD)) + (FC * FD)) / 3
C5 = ((((-FB) * (a ^ 3)) * (4 - (3 * a)) * (1 - FA)) - ((b ^ 3) * (4 -
- (3 * b)) * (1 - FB)) + ((R ^ 3) * (4 - (3 * R)) * (1 - FR)) - ((s ^ 3) *
(4 - (3 * s)) * (1 - FR)) + ((c ^ 3) * (4 - (3 * c)) * (1 - FC)) + ((FC *
(d ^ 3)) * (4 - (3 * d)) * (1 - FD)) + (FC * FD)) / 24
C6 = ((((-FB) * a) * (3 - (3 * a) + (a ^ 2)) * (1 - FA)) - (b * (3 -
(3 * b) + (b ^ 2)) * (1 - FB)) + (R * (3 - (3 * R) + (R ^ 2)) * (1 - FR))
- (s * (3 - (3 * s) + (s ^ 2)) * (1 - FR)) + (c * (3 - (3 * c) + (c ^ 2))
* (1 - FC)) + ((FC * d) * (3 - (3 * d) + (d ^ 2)) * (1 - FD)) + (FC * FD))
/ 3
C7 = (((-FB) * (a ^ 2) * (6 - (8 * a) + (3 * (a ^ 2))) * (1 - FA)) -
((b ^ 2) * (6 - (8 * b) + (3 * (b ^ 2))) * (1 - FB)) + ((R ^ 2) * (6 - (8
* R) + (3 * (R ^ 2))) * (1 - FR)) - ((s ^ 2) * (6 - (8 * s) + (3 * (s ^ 2)
)) * (1 - FR)) + ((c ^ 2) * (6 - (8 * c) + (3 * (c ^ 2))) * (1 - FC)) + ((
FC * (d ^ 2)) * (6 - (8 * d) + (3 * (d ^ 2))) * (1 - FD)) + (FC * FD)) / 2
4
CR = (C6 * C4) - (C3 * C3)
Kcc = C4 / CR
Kc = ((Kcc * GeneralData(CProj).CEc) * ((CC2 * (CC1 ^ 3)) / 12)) / 1c
*****
'to be continued tomorrow
If frmMain.optSystem(1).Value Then
Cs = ((1 - (.63 * (h / CC1))) * ((h ^ 3) * CC1) / 3))
Cd = (1 - (.63 * ((h + CDt) / CC1))) * (((h + CDt) ^ 3) * CC1) / 3)
X2 = (L2 - CDy) / 2
Kt = (GeneralData(CProj).CEc / (((L2 * (1 - ((CC2 / L2) ^ 5))) / (20
* Cd))) + (((L2 * ((1 - (X2 / L2)) ^ 5)) / 20) * ((1 / Cs) - (1 / Cd))))
Else
Kt = ((9 * GeneralData(CProj).CEc * ((1 - (.63 * (h / CC1))) * ((h
^ 3) * CC1) / 3))) / (L2 * ((1 - (CC2 / L2)) ^ 3))
End If
End If
If Not (XY) Then 'Along Y axis
CC1 = Column(CProj, CN).c1
CC2 = Column(CProj, CN).c2
CDx = Column(CProj, CN).DropX
CDy = Column(CProj, CN).DropY
CDt = Column(CProj, CN).DropThick
CCx = Column(CProj, CN).CapitalX
CCy = Column(CProj, CN).CapitalY
CCL = Column(CProj, CN).Length
a = ((h + CDt) / 2) / 1c
b = 0
R = 0
s = 0
c = 0
d = (1 - a)
FA = 0
FB = 1
FR = 1
FC = 1
FD = 0
C3 = (((-FB) * a * a * (3 - (2 * a)) * (1 - FA)) - ((b * b) * (3 - (2
* b)) * (1 - FB)) + ((R ^ 2) * (3 - (2 * R)) * (1 - FR)) - ((s ^ 2) * (3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FCOMPUTE.BAS - 4

```

- (2 * s) * (1 - FR)) + ((c ^ 2) * (3 - (2 * c)) * (1 - FC)) + ((FC * d *
d) * (3 - (2 * d)) * (1 - FD)) + (FC * FD)) / 6
C4 = (((-FB * (a ^ 3)) * (1 - FA)) - ((b ^ 3) * (1 - FB)) + ((R ^ 3)
* (1 - FR)) - ((s ^ 3) * (1 - FR)) + ((c ^ 3) * (1 - FC)) + ((FC * (d ^ 3)
) * (1 - FD)) + (FC * FD)) / 3
C5 = ((((-FB) * (a ^ 3)) * (4 - (3 * a)) * (1 - FA)) - ((b ^ 3) * (4
- (3 * b)) * (1 - FB)) + ((R ^ 3) * (4 - (3 * R)) * (1 - FR)) - ((s ^ 3) *
(4 - (3 * s)) * (1 - FR)) + ((c ^ 3) * (4 - (3 * c)) * (1 - FC)) + ((FC *
(d ^ 3)) * (4 - (3 * d)) * (1 - FD)) + (FC * FD)) / 24
C6 = ((((-FB) * a) * (3 - (3 * a) + (a ^ 2)) * (1 - FA)) - (b * (3 -
(3 * b) + (b ^ 2)) * (1 - FB)) + (R * (3 - (3 * R) + (R ^ 2)) * (1 - FR))
- (s * (3 - (3 * s) + (s ^ 2)) * (1 - FR)) + (c * (3 - (3 * c) + (c ^ 2))
* (1 - FC)) + ((FC * d) * (3 - (3 * d) + (d ^ 2)) * (1 - FD)) + (FC * FD))
/ 3
C7 = ((((-FB) * (a ^ 2) * (6 - (8 * a) + (3 * (a ^ 2)))) * (1 - FA)) -
((b ^ 2) * (6 - (8 * b) + (3 * (b ^ 2)))) * (1 - FB)) + ((R ^ 2) * (6 - (8
* R) + (3 * (R ^ 2)))) * (1 - FR)) - ((s ^ 2) * (6 - (8 * s) + (3 * (s ^ 2)
)) * (1 - FR)) + ((c ^ 2) * (6 - (8 * c) + (3 * (c ^ 2)))) * (1 - FC)) + ((
FC * (d ^ 2)) * (6 - (8 * d) + (3 * (d ^ 2)))) * (1 - FD)) + (FC * FD)) / 2
4
CR = (C6 * C4) - (C3 * C3)
Kcc = C4 / CR
Kc = ((Kcc * GeneralData(CProj).CEc) * ((CC2 * (CC1 ^ 3)) / 12)) / 1c
If frmMain.optSystem(0).Value = True Then
Cs = ((1 - (.63 * (h / CC2))) * ((h ^ 3) * CC2) / 3)
Cd = (1 - (.63 * ((h + CDt) / CC2))) * (((h + CDt) ^ 3) * CC2) / 3
X2 = (L2 - CDx) / 2
Kt = (GeneralData(CProj).CEc / (((L2 * (1 - ((CC1 / L2) ^ 5))) / (20
* Cd))) + (((L2 * ((1 - (X2 / L2)) ^ 5)) / 20) * ((1 / Cs) - (1 / Cd))))
Else
Kt = ((9 * GeneralData(CProj).CEc * ((1 - (.63 * (h / CC2))) * ((h
^ 3) * CC2) / 3))) / (L2 * ((1 - (CC1 / L2)) ^ 3)))
End If
End If
Kec = ((4 * Kc * Kt) / (2 * Kt + 2 * Kc))
Cal_FColumnK = Kec
End Function

Function Cal_FColumnShear (N As Integer, Index As Integer, d As Single, Wu
As Single) 'Check Dia & Punch Shear at Column
Dim b0 As Single
Dim L1 As Single
Dim L2 As Single
Dim lf As Single
Dim XNum As Integer
Dim YNum As Integer
Dim c1 As Single 'Column Width ,Length in x axis
Dim c2 As Single 'Column Thick ,Length in y axis
Dim X_Flag As Integer
Dim Y_Flag As Integer
Dim EdgeIndex As Integer
c1 = Column(CProj, N).c1
c2 = Column(CProj, N).c2
X_Flag = Column(CProj, N).X_Flag
Y_Flag = Column(CProj, N).Y_Flag
L1 = 0
L2 = 0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FCOMPUTE.BAS - 5

```

lf = 0
XNum = 0
YNum = 0
Select Case Index
Case 1: 'Punch
  L1 = 0
  L2 = 0
  lf = 0
  XNum = 0
  YNum = 0
  For I = 1 To ElementXNum(CProj)
    'Span Then Conjunc That Column in X axis Must Less Or Equal 2 Span
    If (SElement(CProj, 0, I).Start = N) Or (SElement(CProj, 0, I).End =
N) Then
      L1 = L1 + (SElement(CProj, 0, I).Length / 2)
      XNum = XNum + 1
    End If
  Next I
  'Span Then Conjunc That Column in Y axis Must Less Or Equal 2 Span
  For I = 1 To ElementYNum(CProj)
    If (SElement(CProj, 1, I).Start = N) Or (SElement(CProj, 1, I).End =
N) Then
      L2 = L2 + (SElement(CProj, 1, I).Length / 2)
      YNum = YNum + 1
    End If
  Next I
  If (((X_Flag = 1) Or (X_Flag = 2)) And ((Y_Flag = 1) Or (Y_Flag = 2)))
) Then
    EdgeIndex = 1
  ElseIf (((X_Flag = 1) Or (X_Flag = 2)) And (Y_Flag = 0)) Or ((X_Flag
= 0) And ((Y_Flag = 1) Or (Y_Flag = 2))) Then
    EdgeIndex = 2
  ElseIf ((X_Flag = 0) And (Y_Flag = 0)) Then
    EdgeIndex = 3
  End If
  Select Case EdgeIndex
  Case 1: 'punch corner
    vn = (Wu * ((L1 * L2) - ((c1 + (d / 2)) * (c2 + d / 2)))) / .85
  Case 2: 'punch edge
    vn = (Wu * ((L1 * L2) - ((c1 + (XNum * d / 2)) * (c2 + (YNum * d / 2
)))) / .85
  Case 3: 'punch interior
    vn = (Wu * ((L1 * L2) - (c1 + d) * (c2 + d))) / .85
  End Select
Case 2: 'Dia
  lf = 0
  XNum = 0
  YNum = 0
  For I = 1 To ElementXNum(CProj) + ElementYNum(CProj)
    If ((EElement(CProj, I).Start = N) Or (EElement(CProj, I).End = N))
And (lf < EElement(CProj, I).Length) Then
      lf = (EElement(CProj, I).Length / 2)
      XNum = 1
    Else
      'XNum = 0
    End If
    If ((EElement(CProj, I).Start = N) Or (EElement(CProj, I).End = N))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

FCOMPUTE.BAS - 6

And (lf < EElement(CProj, I).Length) Then
  lf = (EElement(CProj, I).Length / 2)
  YNum = 1
  Else
  'YNum = 0
  End If
Next I
If (((X_Flag = 1) Or (X_Flag = 2)) And ((Y_Flag = 1) Or (Y_Flag = 2)))
) Then
  EdgeIndex = 1
  ElseIf (((X_Flag = 1) Or (X_Flag = 2)) And (Y_Flag = 0)) Or ((X_Flag
= 0) And ((Y_Flag = 1) Or (Y_Flag = 2)))) Then
  EdgeIndex = 2
  ElseIf ((X_Flag = 0) And (Y_Flag = 0)) Then
  EdgeIndex = 3
  End If
  Select Case EdgeIndex      'Check Shear For 1 m strip
  Case 1: 'dia corner
    vn = (Wu * (lf - d - (XNum * c1) - (YNum * c2))) / .85
  Case 2: 'dia edge
    vn = (Wu * (lf - d - (XNum * (c1 / 2)) - (YNum * (c2 / 2)))) / .85
  Case 3: 'dia interior
    vn = (Wu * (lf - d - (XNum * c1 / 2) - (YNum * c2 / 2))) / .85
  End Select
End Select
Cal_FColumnShear = vn
End Function

Function Cal_FCover (Thk As Single) As Single
Dim Cover As Single
If ((Thk > 0) And (Thk <= 12)) Then
  Cover = .025
End If
If ((Thk > 12) And (Thk <= 20)) Then
  Cover = .03
End If
If ((Thk > 20) And (Thk <= 40)) Then
  Cover = .04
End If
If (Thk > 40) Then
  Cover = .05
End If
Cal_FCover = Cover
End Function

Function Cal_FDF (E As Integer, XY As Integer, LR As Integer) As Single
Dim I As Integer
Dim SumDF As Single
SumDF = 0
If LR Then
  For I = 1 To ElementNum(CProj)
    If (Element(CProj, E).Start = Element(CProj, I).Start) Or (Element(C
Proj, E).Start = Element(CProj, I).End) Then
      SumDF = SumDF + Cal_FK(I, XY)
    End If
  Next I

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FDCOMPUTE.BAS - 7

```

ElseIf Not (LR) Then
  For I = 1 To ElementNum(CProj)
    If (Element(CProj, E).End = Element(CProj, I).Start) Or (Element(CProj, E).End = Element(CProj, I).End) Then
      SumDF = SumDF + Cal_FK(I, XY)
    End If
  Next I
End If
If Not (SumDF = 0) Then
  Cal_FDF = (Cal_FK(E, XY) / SumDF)
Else
  Cal_FDF = 1
End If
End Function

```

```

Sub Cal_FDdropCapital (CN As Integer, h As Single) 'h is Thickness
  Dim Thick As Single
  Dim Lmax As Single 'Now Lmax is Maximum Length of Span
  'For Adapt Lmax is Maximum Length That conjunc The
  Column
  Dim d As Single 'if true is find lmax in x axis or y axis around column
  Lmax = 0
  For I = 1 To ElementXNum(CProj) + ElementYNum(CProj)
    If ((Element(CProj, I).Start = CN) Or (Element(CProj, I).End = CN))
    And (Element(CProj, I).Length > Lmax) Then
      Lmax = Element(CProj, I).Length
    End If
  Next I
  Column(CProj, CN).DropThick = .375 * h
  Column(CProj, CN).DropX = Lmax / 3
  Column(CProj, CN).DropY = Lmax / 3
  Column(CProj, CN).CapitalX = .225 * Lmax
  Column(CProj, CN).CapitalY = .225 * Lmax
  If frmMain.chkCHead(0).Value Then
    Column(CProj, CN).Type = 1
  End If
  If frmMain.chkCHead(1).Value Then
    Column(CProj, CN).Type = 2
  End If
  If frmMain.chkCHead(0).Value And frmMain.chkCHead(1).Value Then
    Column(CProj, CN).Type = 3
  End If
  If frmMain.optSystem(0).Value Then
    Column(CProj, CN).Type = 0
  End If
  If Column(CProj, CN).Type = 3 Then
    Column(CProj, CN).DropThick = .25 * h
  End If
  Select Case Column(CProj, CN).Type
  Case 0:
    Column(CProj, CN).DropThick = 0
    Column(CProj, CN).DropX = 0
    Column(CProj, CN).DropY = 0
    Column(CProj, CN).CapitalX = 0
    Column(CProj, CN).CapitalY = 0
  Case 1:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FCOMPUTE.BAS - 8

```

    Column(CProj, CN).CapitalX = 0
    Column(CProj, CN).CapitalY = 0
Case 2:
    Column(CProj, CN).DropThick = 0
    Column(CProj, CN).DropX = 0
    Column(CProj, CN).DropY = 0
Case 3:
End Select
End Sub

Sub Cal_FEndMoment (ColOrRow As Integer, ColRow As Integer)
Dim Pattern As Integer
Dim axis As Integer
Const ErrorCheck = 1
Dim ErrorValue As Single
Dim I As Integer
If ColOrRow Then
    axis = 0
ElseIf Not ColOrRow Then
    axis = 1
End If
ReDim FEML(ElementNum(CProj)) As Single
ReDim FEMR(ElementNum(CProj)) As Single
ReDim DFL(ElementNum(CProj)) As Single
ReDim DFR(ElementNum(CProj)) As Single
ReDim COML(ElementNum(CProj)) As Single
ReDim COMR(ElementNum(CProj)) As Single
ReDim DML(ElementNum(CProj)) As Single
ReDim DMR(ElementNum(CProj)) As Single
ReDim MNode(NodeNum(CProj)) As Single
ReDim MMaxL(ElementNum(CProj)) As Single
ReDim MMaxR(ElementNum(CProj)) As Single
For I = 1 To ElementNum(CProj)
    MMaxL(I) = 0
    MMaxR(I) = 0
Next I
For Pattern = 1 To 3
For I = 1 To ElementNum(CProj)
    R = Cal_FK(I, Not (ColOrRow))
    FEML(I) = Cal_FFEM(axis, I, True, Pattern)
    FEMR(I) = Cal_FFEM(axis, I, False, Pattern)
    DFL(I) = Cal_FDF(I, Not (ColOrRow), True)
    DFR(I) = Cal_FDF(I, Not (ColOrRow), False)
    COFL(CProj, I) = COFL(CProj, I)
    COFR(CProj, I) = COFR(CProj, I)
    ML(CProj, I) = FEML(I)
    MR(CProj, I) = FEMR(I)
    DML(I) = 0
    DMR(I) = 0
    COML(I) = 0
    COMR(I) = 0
Next I
'Distribution Moment
Do
ErrorValue = 0
'Sum DM
For I = 1 To NodeNum(CProj)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FCOMPUTE.BAS - 9

```

MNode(I) = 0
For J = 1 To ElementNum(CProj)
  If (Element(CProj, J).Start) = I Then
    MNode(I) = (MNode(I) + ML(CProj, J))
  ElseIf (Element(CProj, J).End = I) Then
    MNode(I) = (MNode(I) + MR(CProj, J))
  End If
Next J
Next I
'DM
For I = 1 To NodeNum(CProj)
  For J = 1 To ElementNum(CProj)
    If (Element(CProj, J).Start = I) Then
      DML(J) = (((-1) * MNode(I)) * DFL(J))
    ElseIf (Element(CProj, J).End = I) Then
      DMR(J) = (((-1) * MNode(I)) * DFR(J))
    End If
  Next J
Next I
For I = 1 To ElementNum(CProj)
  If Abs(ErrorValue) < Abs(DML(I)) Then
    ErrorValue = DML(I)
  End If
  If Abs(ErrorValue) < Abs(DMR(I)) Then
    ErrorValue = DMR(I)
  End If
Next I
'Carry over moment
For I = 1 To ElementNum(CProj)
  COML(I) = DMR(I) * COFL(CProj, I)
  COMR(I) = DML(I) * COFR(CProj, I)
Next I
'Sum Moment
If (Abs(ErrorValue) < ErrorCheck) Then
  For I = 1 To ElementNum(CProj)
    ML(CProj, I) = ML(CProj, I) + DML(I)
    MR(CProj, I) = MR(CProj, I) + DMR(I)
  Next I
Else
  For I = 1 To ElementNum(CProj)
    ML(CProj, I) = ML(CProj, I) + DML(I) + COML(I)
    MR(CProj, I) = MR(CProj, I) + DMR(I) + COMR(I)
  Next I
End If
Loop Until (Abs(ErrorValue) < ErrorCheck)
For I = 1 To ElementNum(CProj)
  If Abs(MMaxL(I)) < Abs(ML(CProj, I)) Then
    MMaxL(I) = ML(CProj, I)
  End If
  If Abs(MMaxR(I)) < Abs(MR(CProj, I)) Then
    MMaxR(I) = MR(CProj, I)
  End If
  MD(CProj, I) = (1 / 8) * ((DeadCo * DeadLoad(CProj)) + (1.8 * LiveLoad(CProj))) * ((Element(CProj, I).L1 + Element(CProj, I).L2) / 2) * (Element(CProj, I).Length) ^ 2
Next I
Next Pattern

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FCOMPUTE.BAS - 10

```

For I = 1 To ElementNum(CProj)
  ML(CProj, I) = MMaxL(I)
  MR(CProj, I) = MMaxR(I)
  MD(CProj, I) = MD(CProj, I) - (ML(CProj, I) + MR(CProj, I)) / 2
Next I

```

```

''' Show End Moment at Programmed
'Load frmOutText(CProj)
'frmOutText(CProj).Show
'frmOutText(CProj).Cls
'For I = 1 To ElementNum(CProj)
' frmOutText(CProj).Print "Fixed end moment Left element" + Format$(I)
+ " = " + Format$(FEML(I), "0.00")
' frmOutText(CProj).Print "Fixed end moment Right element" + Format$(I)
) + " = " + Format$(FEMR(I), "0.00")
' frmOutText(CProj).Print "DF Factor Left " + Format$(DFL(I), "0.000")
' frmOutText(CProj).Print "DF Factor Right " + Format$(DFR(I), "0.000"
)
' frmOutText(CProj).Print "End Moment Left " + Format$(ML(CProj, I), "
0.00")
' frmOutText(CProj).Print "End Moment Right " + Format$(MR(CProj, I),
"0.00")
'Next I
End Sub

```

```

Function Cal_FFEM (axis As Integer, E As Integer, LeftOrRight As Integer,
Pattern As Integer) As Single
  Dim L1 As Single
  Dim L2 As Single
  Dim L3 As Single
  Dim Wu As Single
  Dim N As Integer
  Wu = Cal_FWu(E, LeftOrRight, Pattern)
  L1 = Element(CProj, E).Length
  L2 = ((Element(CProj, E).L1 / 2) + (Element(CProj, E).L2 / 2))
  If LeftOrRight Then
    Cal_FFEM = (-1) * FEMCL(CProj, E) * Wu * L2 * L1 ^ 2
  ElseIf Not LeftOrRight Then
    Cal_FFEM = FEMCL(CProj, E) * Wu * L2 * L1 ^ 2
  End If
End Function

```

```

Function Cal_FK (E As Integer, XY As Integer) As Single 'Element of Slab s1
tiffness
  Dim EC As Single
  Dim Ks As Single
  Dim ka As Single
  Dim kb As Single
  Dim L1 As Single
  Dim L2 As Single
  Dim a As Single
  Dim b As Single
  Dim R As Single
  Dim s As Single
  Dim c As Single
  Dim d As Single
  Dim I1 As Single

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

F COMPUTE.BAS - 11

```

Dim I2 As Single
Dim I3 As Single
Dim I4 As Single
Dim I5 As Single
Dim I6 As Single
Dim Y_ As Single
Dim CC1 As Single
Dim CC2 As Single
Dim CDx As Single
Dim CDy As Single
Dim CDt As Single
Dim CCx As Single
Dim CCy As Single
Dim CCL As Single
Dim c1 As Single
Dim c2 As Single
Dim C3 As Single
Dim C4 As Single
Dim C5 As Single
Dim C6 As Single
Dim C7 As Single
Dim FA As Single
Dim FB As Single
Dim FR As Single
Dim FC As Single
Dim FD As Single
Dim CR As Single
Dim h As Single 'Thickness
Dim NodeA As Integer
Dim NodeB As Integer
Dim ElementNum As Integer
h = Thick(CProj)
EC = GeneralData(CProj).CEC
L1 = Element(CProj, E).Length
L2 = ((Element(CProj, E).L1 / 2) + (Element(CProj, E).L2 / 2))
NodeA = Element(CProj, E).CStart
NodeB = Element(CProj, E).CEnd
If XY Then
  a = (Column(CProj, NodeA).c1 / 2) / L1
  b = (Column(CProj, NodeA).CapitalX / 2) / L1
  R = (Column(CProj, NodeA).DropX / 2) / L1
  s = (L1 - (Column(CProj, NodeB).DropX / 2)) / L1
  c = (L1 - (Column(CProj, NodeB).CapitalX / 2)) / L1
  d = (L1 - (Column(CProj, NodeB).c1 / 2)) / L1
  CC1 = Column(CProj, NodeA).c1
  CC2 = Column(CProj, NodeA).c2
  CDx = Column(CProj, NodeA).DropX
  CDy = Column(CProj, NodeA).DropY
  CDt = Column(CProj, NodeA).DropThick
  CCx = Column(CProj, NodeA).CapitalX
  CCy = Column(CProj, NodeA).CapitalY
  CCL = Column(CProj, NodeA).Length
  Y_ = ((h * L2) * ((h / 2) + CDt) + (CDy * (CDt ^ 2) * .5)) / ((h * L2)
) + (CDy * CDt)
  I6 = (L2 * (h ^ 3)) / 12
  I1 = I6 + ((h * L2) * ((h / 2) - Y_) ^ 2) + ((CDy * (CDt ^ 3)) / 12)
) + (CDy * CDt * (Y_ - (CDt / 2)))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FCOMPUTE.BAS - 12

```

I2 = I1 / ((1 - (CC2 / L2)) ^ 2)
I4 = I2
CC1 = Column(CProj, NodeB).c1
CC2 = Column(CProj, NodeB).c2
CDx = Column(CProj, NodeB).DropX
CDy = Column(CProj, NodeB).DropY
CDt = Column(CProj, NodeB).DropThick
CCx = Column(CProj, NodeB).CapitalX
CCy = Column(CProj, NodeB).CapitalY
CCL = Column(CProj, NodeB).Length
Y_ = ((h * L2) * ((h / 2) + CDt) + (CDy * (CDt ^ 2) * .5)) / ((h * L2)
) + (CDy * CDt)
I1 = I6 + ((h * L2) * (((h / 2) - Y_) ^ 2)) + ((CDy * (CDt ^ 3)) / 12
) + (CDy * CDt * (Y_ - (CDt / 2)))
I3 = I1 / ((1 - (CC2 / L2)) ^ 2)
I5 = I3
FA = (1 - (Column(CProj, NodeA).c2 / L2)) ^ 2
FB = I1 / I2
FR = I1 / I6
FC = I1 / I3
FD = (1 - (Column(CProj, NodeB).c2 / L2)) ^ 2
C3 = (((-FB) * a * a * (3 - (2 * a)) * (1 - FA)) - ((b * b) * (3 - (2
* b)) * (1 - FB)) + ((R ^ 2) * (3 - (2 * R)) * (1 - FR)) - ((s ^ 2) * (3
- (2 * s)) * (1 - FR)) + ((c ^ 2) * (3 - (2 * c)) * (1 - FC)) + ((FC * d *
d) * (3 - (2 * d)) * (1 - FD)) + (FC * FD)) / 6
C4 = (((-FB) * (a ^ 3)) * (1 - FA)) - ((b ^ 3) * (1 - FB)) + ((R ^ 3)
* (1 - FR)) - ((s ^ 3) * (1 - FR)) + ((c ^ 3) * (1 - FC)) + ((FC * (d ^ 3)
) * (1 - FD)) + (FC * FD)) / 3
C5 = (((-FB) * (a ^ 3)) * (4 - (3 * a)) * (1 - FA)) - ((b ^ 3) * (4
- (3 * b)) * (1 - FB)) + ((R ^ 3) * (4 - (3 * R)) * (1 - FR)) - ((s ^ 3) *
(4 - (3 * s)) * (1 - FR)) + ((c ^ 3) * (4 - (3 * c)) * (1 - FC)) + ((FC *
(d ^ 3)) * (4 - (3 * d)) * (1 - FD)) + (FC * FD)) / 24
C6 = (((-FB) * a) * (3 - (3 * a) + (a ^ 2)) * (1 - FA)) - (b * (3 -
(3 * b) + (b ^ 2)) * (1 - FB)) + (R * (3 - (3 * R) + (R ^ 2)) * (1 - FR))
- (s * (3 - (3 * s) + (s ^ 2)) * (1 - FR)) + (c * (3 - (3 * c) + (c ^ 2))
* (1 - FC)) + ((FC * d) * (3 - (3 * d) + (d ^ 2)) * (1 - FD)) + (FC * FD))
/ 3
C7 = (((-FB) * (a ^ 2) * (6 - (8 * a) + (3 * (a ^ 2)))) * (1 - FA)) -
((b ^ 2) * (6 - (8 * b) + (3 * (b ^ 2)))) * (1 - FB)) + ((R ^ 2) * (6 - (8
* R) + (3 * (R ^ 2)))) * (1 - FR)) - ((s ^ 2) * (6 - (8 * s) + (3 * (s ^ 2)
))) * (1 - FR)) + ((c ^ 2) * (6 - (8 * c) + (3 * (c ^ 2)))) * (1 - FC)) + ((
FC * (d ^ 2)) * (6 - (8 * d) + (3 * (d ^ 2)))) * (1 - FD)) + (FC * FD)) / 2
4
CR = (C6 * C4) - (C3 * C3)
COFL(CProj, E) = C3 / C4
COFR(CProj, E) = C3 / C6
ka = C4 / CR
kb = C6 / CR
FEMCL(CProj, E) = ((C7 * C4) - (C5 * C3)) / CR
FEMCR(CProj, E) = ((C6 * C5) - (C7 * C3)) / CR
Ks = ((ka * (GeneralData(CProj).SEc) * (L2 * (h ^ 3))) / 12) / L1
End If
If Not (XY) Then
a = (Column(CProj, NodeA).c2 / 2) / L1
b = (Column(CProj, NodeA).CapitalY / 2) / L1
R = (Column(CProj, NodeA).DropY / 2) / L1
s = (L1 - (Column(CProj, NodeB).DropY / 2)) / L1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FCOMPUTE.BAS - 13

```

c = (L1 - (Column(CProj, NodeB).CapitalY / 2)) / L1
d = (L1 - (Column(CProj, NodeB).c2 / 2)) / L1
CC1 = Column(CProj, NodeA).c2
CC2 = Column(CProj, NodeA).c1
CDx = Column(CProj, NodeA).DropY
CDy = Column(CProj, NodeA).DropX
CDt = Column(CProj, NodeA).DropThick
CCx = Column(CProj, NodeA).CapitalY
CCy = Column(CProj, NodeA).CapitalX
CCL = Column(CProj, NodeA).Length
Y_ = ((h * L2) * ((h / 2) + CDt) + (CDy * (CDt ^ 2) * .5)) / ((h * L2
) + (CDy * CDt))
I6 = (L2 * (h ^ 3)) / 12
I1 = I6 + ((h * L2) * (((h / 2) - Y_) ^ 2)) + ((CDy * (CDt ^ 3)) / 12
) + (CDy * CDt * (Y_ - (CDt / 2)))
I2 = I1 / ((1 - (CC2 / L2)) ^ 2)
I4 = I2
CC1 = Column(CProj, NodeB).c2
CC2 = Column(CProj, NodeB).c1
CDx = Column(CProj, NodeB).DropY
CDy = Column(CProj, NodeB).DropX
CDt = Column(CProj, NodeB).DropThick
CCx = Column(CProj, NodeB).CapitalY
CCy = Column(CProj, NodeB).CapitalX
CCL = Column(CProj, NodeB).Length
Y_ = ((h * L2) * ((h / 2) + CDt) + (CDy * (CDt ^ 2) * .5)) / ((h * L2
) + (CDy * CDt))
I1 = I6 + ((h * L2) * (((h / 2) - Y_) ^ 2)) + ((CDy * (CDt ^ 3)) / 12
) + (CDy * CDt * (Y_ - (CDt / 2)))
I3 = I1 / ((1 - (CC2 / L2)) ^ 2)
I5 = I3
FA = (1 - (Column(CProj, NodeA).c1 / L2)) ^ 2
FB = I1 / I2
FR = I1 / I6
FC = I1 / I3
FD = (1 - (Column(CProj, NodeB).c1 / L2)) ^ 2
C3 = (((-FB) * a * a * (3 - (2 * a)) * (1 - FA)) - ((b * b) * (3 - (2
* b)) * (1 - FB)) + ((R ^ 2) * (3 - (2 * R)) * (1 - FR)) - ((s ^ 2) * (3
- (2 * s)) * (1 - FR)) + ((c ^ 2) * (3 - (2 * c)) * (1 - FC)) + ((FC * d *
d) * (3 - (2 * d)) * (1 - FD)) + (FC * FD)) / 6
C4 = (((-FB * (a ^ 3)) * (1 - FA)) - ((b ^ 3) * (1 - FB)) + ((R ^ 3)
* (1 - FR)) - ((s ^ 3) * (1 - FR)) + ((c ^ 3) * (1 - FC)) + ((FC * (d ^ 3)
) * (1 - FD)) + (FC * FD)) / 3
C5 = ((((-FB) * (a ^ 3)) * (4 - (3 * a)) * (1 - FA)) - ((b ^ 3) * (4
- (3 * b)) * (1 - FB)) + ((R ^ 3) * (4 - (3 * R)) * (1 - FR)) - ((s ^ 3) *
(4 - (3 * s)) * (1 - FR)) + ((c ^ 3) * (4 - (3 * c)) * (1 - FC)) + ((FC *
(d ^ 3)) * (4 - (3 * d)) * (1 - FD)) + (FC * FD)) / 24
C6 = ((((-FB) * a) * (3 - (3 * a) + (a ^ 2)) * (1 - FA)) - (b * (3 -
(3 * b) + (b ^ 2)) * (1 - FB)) + (R * (3 - (3 * R) + (R ^ 2)) * (1 - FR))
- (s * (3 - (3 * s) + (s ^ 2)) * (1 - FR)) + (c * (3 - (3 * c) + (c ^ 2))
* (1 - FC)) + ((FC * d) * (3 - (3 * d) + (d ^ 2)) * (1 - FD)) + (FC * FD))
/ 3
C7 = (((-FB) * (a ^ 2) * (6 - (8 * a) + (3 * (a ^ 2))) * (1 - FA)) -
((b ^ 2) * (6 - (8 * b) + (3 * (b ^ 2))) * (1 - FB)) + ((R ^ 2) * (6 - (8
* R) + (3 * (R ^ 2))) * (1 - FR)) - ((s ^ 2) * (6 - (8 * s) + (3 * (s ^ 2)
)) * (1 - FR)) + ((c ^ 2) * (6 - (8 * c) + (3 * (c ^ 2))) * (1 - FC)) + ((
FC * (d ^ 2)) * (6 - (8 * d) + (3 * (d ^ 2))) * (1 - FD)) + (FC * FD)) / 2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

F COMPUTE.BAS - 14

```

4
CR = (C6 * C4) - (C3 * C3)
COFL(CProj, E) = C3 / C4
COFR(CProj, E) = C3 / C6
ka = C4 / CR
kb = C6 / CR
FEMCL(CProj, E) = ((C7 * C4) - (C5 * C3)) / CR
FEMCR(CProj, E) = ((C6 * C5) - (C7 * C3)) / CR
Ks = ((ka * (GeneralData(CProj).SEc) * (L2 * (h ^ 3))) / 12) / L1
End If
Cal_FK = Ks
End Function

Function Cal_FMoment (Index As Integer, ElementNum As Integer, Element As
ElementType) As Single
Dim w As Single
Dim L1 As Single
Dim L2 As Single
Select Case Index
Case 1: 'Left

Case 2: 'Mid
Case 3: 'Right
End Select
End Function

Sub Cal_FReinforce ()
End Sub

Function Cal_FThick () As Single
Dim Lmax As Single
Dim Thick As Single
Dim c As Single
Dim E As Integer
Lmax = 0
For I = 1 To ElementXNum(CProj) 'find lmax in x axis
If SElement(CProj, 0, I).Length > Lmax Then
Lmax = SElement(CProj, 0, I).Length
c = Column(CProj, SElement(CProj, 0, I).Start).c1
End If
Next I
For I = 1 To ElementYNum(CProj) 'find lmax in y axis
If SElement(CProj, 1, I).Length > Lmax Then
Lmax = SElement(CProj, 1, I).Length
c = Column(CProj, SElement(CProj, 1, I).Start).c2
End If
Next I
h = ((Lmax - c) * (.8 + ((2.844 * GeneralData(CProj).RFs_) / 40000)))
/ 36)
Select Case GeneralData(CProj).SlabIndex 'thick for flat plate or flat
slab

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FCOMPUTE.BAS - 15

```

Case 0:
  h = h
Case 1, 2, 3:
  h = 1.1 * h
End Select
Dim x As Integer
h = h * 100
x = Int(h)
If h - x > 0 Then
  h = x + 1
ElseIf h - x = 0 Then
  h = x
End If
h = h / 100
Cal_FThick = h
End Function

```

Function Cal_FWu (E As Integer, LeftOrRight As Integer, Pattern As Integer
) As Single 'Check By Load Pattern 3 Case

```

Dim Wu As Single
Select Case Pattern
Case 0:
  Wu = DeadLoad(CProj)
Case 1:
  Wu = 1.8 * LiveLoad(CProj) + DeadCo * DeadLoad(CProj)
Case 2:
  If (E Mod 2) = 1 Then
    Wu = DeadCo * DeadLoad(CProj) + .75 * LiveCo * LiveLoad(CProj)
  ElseIf (E Mod 2) = 0 Then
    Wu = DeadCo * DeadLoad(CProj)
  End If
Case 3:
  If (E Mod 2) = 1 Then
    Wu = DeadCo * DeadLoad(CProj)
  ElseIf (E Mod 2) = 0 Then
    Wu = DeadCo * DeadLoad(CProj) + .75 * LiveCo * LiveLoad(CProj)
  End If
Case 4:
  Wu = DeadCo * DeadLoad(CProj) + LiveCo * LiveLoad(CProj) + DeadCo * SuperLoad(CProj)
Case 5:
  Wu = DeadLoad(CProj) + LiveLoad(CProj) + SuperLoad(CProj)
End Select
Cal_FWu = Wu
End Function

```

Function Check_FFShear (Diag As Integer, b0 As Single, d As Single, N As Integer) As Single

```

Dim Vc As Single 'Diagonal check diag = True
Dim vc1 As Single
Dim vc2 As Single
Dim vc3 As Single
Dim Bs As Single
Dim AAs As Single
Dim c1 As Single
Dim c2 As Single

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FCOMPUTE.BAS - 16

```

Dim X_Flag As Integer
Dim Y_Flag As Integer
c1 = Column(CProj, N).c1
c2 = Column(CProj, N).c2
X_Flag = Column(CProj, N).X_Flag
Y_Flag = Column(CProj, N).Y_Flag
Dim EdgeIndex As Integer
If (((X_Flag = 1) Or (X_Flag = 2)) And ((Y_Flag = 1) Or (Y_Flag = 2)))
Then
EdgeIndex = 1 'Corner
ElseIf (((X_Flag = 1) Or (X_Flag = 2)) And (Y_Flag = 0)) Or ((X_Flag
= 0) And ((Y_Flag = 1) Or (Y_Flag = 2))) Then
EdgeIndex = 2 'Edge
ElseIf ((X_Flag = 0) And (Y_Flag = 0)) Then
EdgeIndex = 3 'Interior
End If
Select Case EdgeIndex
Case 1: 'Corner
AAs = 20
Case 2: 'Edge
AAs = 30
Case 3: 'Interior
AAs = 40
End Select
Bs = (c1 / c2)
If Bs < 1 Then
Bs = (Bs ^ -1)
End If
If Diag Then
Vc = (.53) * Sqr(GeneralData(CProj).SFc_) * b0 * d
Vc = Vc * 10000
End If
'punch checking
vc1 = (.53 + (1.06 / Bs)) * Sqr(GeneralData(CProj).SFc_) * b0 * d
vc2 = .265 * ((AAs / (d * b0)) + 2) * Sqr(GeneralData(CProj).SFc_) * b
0 * d
vc3 = 1.06 * Sqr(GeneralData(CProj).SFc_) * b0 * d
If Not (Diag) Then 'Find minimum vc
Vc = vc1
If vc2 < Vc Then
Vc = vc2
End If
If vc3 < Vc Then
Vc = vc3
End If
Vc = Vc * 10000
End If
Check_FFShear = Vc
End Function

```

```

Function Check_FShear (Diag As Integer, vn As Single, b0 As Single, d As S
ingle, N As Integer) As Integer 'Unit Shear That Checked If True Check Pas
s, If False Check Not Pass

```

```

Dim Vc As Single 'Diagonal check diag = True
Dim vc1 As Single
Dim vc2 As Single
Dim vc3 As Single

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FCOMPUTE.BAS - 17

```

Dim Bs As Single
Dim AAs As Single
Dim c1 As Single
Dim c2 As Single
Dim X_Flag As Integer
Dim Y_Flag As Integer
c1 = Column(CProj, N).c1
c2 = Column(CProj, N).c2
X_Flag = Column(CProj, N).X_Flag
Y_Flag = Column(CProj, N).Y_Flag
Dim EdgeIndex As Integer
If (((X_Flag = 1) Or (X_Flag = 2)) And ((Y_Flag = 1) Or (Y_Flag = 2)))
) Then
EdgeIndex = 1      'Corner
ElseIf (((X_Flag = 1) Or (X_Flag = 2)) And (Y_Flag = 0)) Or ((X_Flag
= 0) And ((Y_Flag = 1) Or (Y_Flag = 2))) Then
EdgeIndex = 2      'Edge
ElseIf ((X_Flag = 0) And (Y_Flag = 0)) Then
EdgeIndex = 3      'Interior
End If
Select Case EdgeIndex
Case 1: 'Corner
AAs = 20
Case 2: 'Edge
AAs = 30
Case 3: 'Interior
AAs = 40
End Select
Bs = (c1 / c2)
If Bs < 1 Then
Bs = (Bs ^ -1)
End If
If Diag Then
Vc = (.53) * Sqr(GeneralData(CProj).SFC_) * b0 * d
Vc = Vc * 10000
End If
'punch checking
vc1 = (.53 + (1.06 / Bs)) * Sqr(GeneralData(CProj).SFC_) * b0 * d
vc2 = .265 * (((AAs * d) / b0) + 2) * Sqr(GeneralData(CProj).SFC_) * b
0 * d
vc3 = 1.06 * Sqr(GeneralData(CProj).SFC_) * b0 * d
If Not (Diag) Then 'Find minimum vc
Vc = vc1
If vc2 < Vc Then
Vc = vc2
End If
If vc3 < Vc Then
Vc = vc3
End If
Vc = Vc * 10000
End If
If Vc >= vn Then
Check_FSshear = True
Else
Check_FSshear = False
End If
End Function

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FDESIGN.BAS - 1

```

Sub Design_FSlab ()
  Dim DiagIndex As Integer
  Dim Cov As Single
  Dim Lmax As Single
  Dim Wu As Single 'Kg/m^2 1.4DL+1.7LL
  Dim b0 As Single 'length of punching
  Dim I As Integer
  Dim J As Integer
  'shear
  'Column
  Dim Vc0 As Single
  Dim Vn0 As Single
  'Drop
  Dim Vc1 As Single
  Dim Vn1 As Single
  'Capital
  Dim Vc2 As Single
  Dim Vn2 As Single
  'Diagnal
  Dim Vc3 As Single
  Dim Vn3 As Single

  'M design
  Dim MDL As Single
  Dim MDD As Single
  Dim MDR As Single
  'M design at column strip
  Dim MCL As Single
  Dim MCD As Single
  Dim MCR As Single
  'M design at middle strip
  Dim MML1 As Single
  Dim MMD1 As Single
  Dim MMR1 As Single
  Dim MML2 As Single
  Dim MMD2 As Single
  Dim MMR2 As Single
  ''''Area of reinforce steel
  'Half middle strip 1
  Dim MLB1 As Single
  Dim MMB1 As Single
  Dim MRB1 As Single
  Dim MLT1 As Single
  Dim MMT1 As Single
  Dim MRT1 As Single
  'Half middle strip 2
  Dim MLB2 As Single
  Dim MMB2 As Single
  Dim MRB2 As Single
  Dim MLT2 As Single
  Dim MMT2 As Single
  Dim MRT2 As Single
  'column strip
  Dim CLB As Single
  Dim CMB As Single

```



FDESIGN.BAS - 2

```

Dim CRB As Single
Dim CLT As Single
Dim CMT As Single
Dim CRT As Single
'width of strip
Dim b As Single
Dim b1 As Single
Dim b2 As Single

```

```

Dim RNum As Integer 'data read number
Dim WNum As Integer 'data write number
Dim W As Single
Dim X As Single
Dim LT As Single 'L transverse
Dim L As Single
Dim de As Single 'Thick effective

```

```

Access Data

```

```

''' Set Constanst

```

```

Trans Element

```

```

Thick(CProj) = Cal_FThick()

```

```

Cov = Cal_FCover(Thick(CProj))

```

```

DeadLoad(CProj) = Thick(CProj) * 2400

```

```

Wu = DeadCo * DeadLoad(CProj) + LiveCo * LiveLoad(CProj)

```

```

Trans_UniformLoad Wu 'Find Load Each Element

```

```

d(CProj) = Thick(CProj) - Cov 'Try d Initial

```

```

'''Display

```

```

frmMain.txtThick.Text = Format$(Thick(CProj) * 100, "0.000")

```

```

frmMain.txtDL.Text = Format$(Thick(CProj) * 2400, "0.00")

```

```

For I = 1 To SNodeNum(CProj)

```

```

    Cal_FDropCapital I, Thick(CProj)

```

```

Next I

```

```

****

```

```

DiagIndex = 1 'Check Punch

```

```

For I = 1 To SNodeNum(CProj) 'Test Shear ,Find For Slab Thickness,Check

```

```

k For All Column

```

```

    DiagIndex = 1

```

```

    If Column(CProj, I).SupportType <> 2 Then 'If Not Free,Mean That is Column support

```

```

        Select Case GeneralData(CProj).SlabIndex

```

```

            Case 0:

```

```

                de = d(CProj)

```

```

                b0 = Cal_Fb0(Column(CProj, I), 0)

```

```

                Do While Not (Check_FShear(False, Cal_FColumnShear(I, DiagIndex, de

```

```

, Wu), b0, de, I)) 'If Not pass Revise Thick ,False Is Diagonal Check

```

```

                    Thick(CProj) = Thick(CProj) + .01

```

```

                    frmMain.txtThick.Text = Format$(Thick(CProj) * 100, "0.00") 'Display

```

```

                    y in txtThick ,Thick

```

```

                    frmMain.txtDL.Text = Format$(Thick(CProj) * 2400, "0.00") 'Display

```

```

                    in txtDL ,Dead Load

```

```

                    DeadLoad(CProj) = Thick(CProj) * 2400

```

```

                    Wu = (DeadCo * DeadLoad(CProj)) + (LiveCo * LiveLoad(CProj))

```

```

                    d(CProj) = Thick(CProj) - Cov

```

```

                    de = d(CProj)

```

```

                    b0 = Cal_Fb0(Column(CProj, I), 0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FDESGN.BAS - 3

```

Loop
Case 1:
Cal_FDropCapital I, Thick(CProj)
de = d(CProj)
b0 = Cal_Fb0(Column(CProj, I), 1)'Check Drop panel
Do While Not (Check_FShear(False, Cal_FColumnShear(I, DiagIndex, d(
CProj), Wu), b0, d(CProj), I)) 'If Not pass Revise Thick ,False Is Diago
nal Check
Thick(CProj) = Thick(CProj) + .01
frmMain.txtThick.Text = Format$(Thick(CProj) * 100, "0.00")'Displa
y in txtThick ,Thick
frmMain.txtDL.Text = Format$(Thick(CProj) * 2400, "0.00") 'Display
in txtDL ,Dead Load
DeadLoad(CProj) = Thick(CProj) * 2400
Wu = (DeadCo * DeadLoad(CProj)) + (LiveCo * LiveLoad(CProj))
d(CProj) = Thick(CProj) - Cov
Cal_FDropCapital I, Thick(CProj)
de = d(CProj)
b0 = Cal_Fb0(Column(CProj, I), 1)'Check Drop panel
Loop
Case 2:
Cal_FDropCapital I, Thick(CProj)
de = d(CProj)
b0 = Cal_Fb0(Column(CProj, I), 2)'Check Capital
Do While Not (Check_FShear(False, Cal_FColumnShear(I, DiagIndex, d(
CProj), Wu), b0, d(CProj), I)) 'If Not pass Revise Thick ,False Is Diago
nal Check
Thick(CProj) = Thick(CProj) + .01
frmMain.txtThick.Text = Format$(Thick(CProj) * 100, "0.00")'Displa
y in txtThick ,Thick
frmMain.txtDL.Text = Format$(Thick(CProj) * 2400, "0.00") 'Display
in txtDL ,Dead Load
DeadLoad(CProj) = Thick(CProj) * 2400
Wu = (DeadCo * DeadLoad(CProj)) + (LiveCo * LiveLoad(CProj))
d(CProj) = Thick(CProj) - Cov
Cal_FDropCapital I, Thick(CProj)
de = d(CProj)
b0 = Cal_Fb0(Column(CProj, I), 2)'Check Capital
Loop
Case 3:
Cal_FDropCapital I, Thick(CProj)
de = d(CProj)
b0 = Cal_Fb0(Column(CProj, I), 1)'Check Drop panel
Do While Not (Check_FShear(False, Cal_FColumnShear(I, DiagIndex, d(
CProj), Wu), b0, d(CProj), I)) 'If Not pass Revise Thick ,False Is Diago
nal Check
Thick(CProj) = Thick(CProj) + .01
frmMain.txtThick.Text = Format$(Thick(CProj) * 100, "0.00")'Displa
y in txtThick ,Thick
frmMain.txtDL.Text = Format$(Thick(CProj) * 2400, "0.00") 'Display
in txtDL ,Dead Load
DeadLoad(CProj) = Thick(CProj) * 2400
Wu = (DeadCo * DeadLoad(CProj)) + (LiveCo * LiveLoad(CProj))
d(CProj) = Thick(CProj) - Cov
Cal_FDropCapital I, Thick(CProj)
de = d(CProj)
b0 = Cal_Fb0(Column(CProj, I), 1)'Check Drop panel

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FDESIGN.BAS - 4

```

Loop
Cal_FDropCapital I, Thick(CProj)
de = d(CProj) + Column(CProj, I).DropThick
b0 = Cal_Fb0(Column(CProj, I), 2)'Check capital
Do While Not (Check_FShear(False, Cal_FColumnShear(I, DiagIndex, d(
CProj), Wu), b0, d(CProj), I)) 'If Not pass Revise Thick ,False Is Diago
nal Check
Thick(CProj) = Thick(CProj) + .01
frmMain.txtThick.Text = Format$(Thick(CProj) * 100, "0.00")'Displa
y in txtThick ,Thick
frmMain.txtDL.Text = Format$(Thick(CProj) * 2400, "0.00") 'Display
in txtDL ,Dead Load
DeadLoad(CProj) = Thick(CProj) * 2400
Wu = (DeadCo * DeadLoad(CProj)) + (LiveCo * LiveLoad(CProj))
d(CProj) = Thick(CProj) - Cov
Cal_FDropCapital I, Thick(CProj)
de = d(CProj) + Column(CProj, I).DropThick
b0 = Cal_Fb0(Column(CProj, I), 2)'Check capital
Loop
End Select
End If
'check diagonal in column
DiagIndex = 2 'Check Diagonal
If Column(CProj, I).SupportType <> 2 Then 'If Not Free,Mean That is C
olumn support
b0 = 1
Do While Not (Check_FShear(True, Cal_FColumnShear(I, DiagIndex, d(CP
roj), Wu), b0, d(CProj), I)) 'If Not pass Revise Thick ,False Is Diagona
l Check
Thick(CProj) = Thick(CProj) + .01
frmMain.txtThick.Text = Format$(Thick(CProj) * 100, "0.00")'Display
in txtThick ,Thick
frmMain.txtDL.Text = Format$(Thick(CProj) * 2400, "0.00") 'Display
in txtDL ,Dead Load
DeadLoad(CProj) = Thick(CProj) * 2400
Wu = (DeadCo * DeadLoad(CProj)) + (LiveCo * LiveLoad(CProj))
d(CProj) = Thick(CProj) - Cov
Loop
End If
Next I

''''Find Thick Complete
DeadLoad(CProj) = Thick(CProj) * 2400
frmMain.txtThick.Text = Format$(Thick(CProj) * 100, "0.00")'Display in
txtThick ,Thick
frmMain.txtDL.Text = Format$(DeadLoad(CProj), "0.00") 'Display in txtD
L ,Dead Load

'''' 'Find Drop And Capital
For I = 1 To SNodeNum(CProj)
Cal_FDropCapital I, Thick(CProj)
Next I

Trans_UniformLoad Wu
''''Write Data output in File

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FDESIGN.BAS - 5

```

WNum = 0
Open DirApp + "temp.swp" For Random Access Read Write Lock Read Write
As #20 Len = 4
Put #20, WNum + 1, Thick(CProj)
Put #20, WNum + 2, Cov
WNum = WNum + 2
''' Puncing Shear
DiagIndex = 1 'Check Punch
For I = 1 To SNodeNum(CProj) 'Test Shear ,Find For Slab Thickness,Ch
ck For All Column
DiagIndex = 1'Check Punch
If Column(CProj, I).SupportType <> 2 Then 'If Not Free,Mean That is
Column support
Select Case GeneralData(CProj).SlabIndex
Case 0:
de = d(CProj)
b0 = Cal_Fb0(Column(CProj, I), 0)
Vc0 = Check_FFShear(False, b0, de, I) 'If Not pass Revise Thick ,F
also Is Diagonal Check
Vn0 = Cal_FColumnShear(I, DiagIndex, de, Wu)
Vc1 = 0
Vn1 = 0
Vc2 = 0
Vn2 = 0
Case 1:
de = d(CProj)
b0 = Cal_Fb0(Column(CProj, I), 1)'Check Drop panel
Vc0 = 0
Vn0 = 0
Vc1 = Check_FFShear(False, b0, d(CProj), I)
Vn1 = Cal_FColumnShear(I, DiagIndex, d(CProj), Wu)
Vc2 = 0
Vn2 = 0
Vc3 = 0
Vn3 = 0
Case 2:
de = d(CProj)
b0 = Cal_Fb0(Column(CProj, I), 2)'Check Capital
Vc0 = 0
Vn0 = 0
Vc1 = 0
Vn1 = 0
Vc2 = Check_FFShear(False, b0, d(CProj), I)
Vn2 = Cal_FColumnShear(I, DiagIndex, d(CProj), Wu)
Vc3 = 0
Vn3 = 0
Case 3:
Vc0 = 0
Vn0 = 0
de = d(CProj)
b0 = Cal_Fb0(Column(CProj, I), 1)'Check Drop panel
Vc1 = Check_FFShear(False, b0, d(CProj), I)
Vn1 = Cal_FColumnShear(I, DiagIndex, d(CProj), Wu)
de = d(CProj) + Column(CProj, I).DropThick
b0 = Cal_Fb0(Column(CProj, I), 2)'Check capital
Vc2 = Check_FFShear(False, b0, d(CProj), I)
Vn2 = Cal_FColumnShear(I, DiagIndex, d(CProj), Wu)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FDESIGN.BAS - 6

```

End Select
End If
'check diagonal in column
DiagIndex = 2 'Check Diagonal
If Column(CProj, I).SupportType <> 2 Then 'If Not Free,Mean That is
Column support
b0 = 1
Vc3 = (Check_FFShear(True, b0, d(CProj), I)) 'If Not pass Revise
Thick ,False Is Diagonal Check
Vn3 = Cal_FColumnShear(I, DiagIndex, d(CProj), Wu)
End If
Put #20, WNum + 1, Vc0
Put #20, WNum + 2, Vn0
Put #20, WNum + 3, Vc1
Put #20, WNum + 4, Vn1
Put #20, WNum + 5, Vc2
Put #20, WNum + 6, Vn2
Put #20, WNum + 7, Vc3
Put #20, WNum + 8, Vn3
WNum = WNum + 8
Next I

''' Write end moment & mid span moment and transfer to design moment
at edge of column
For I = 1 To SRow(CProj)
  Trans_BElement False, I 'False is calculate in x axis
  Cal_FEndMoment False, I
  For J = 1 To RowEle(I)
    Put #20, WNum + 1, ML(CProj, J)
    Put #20, WNum + 2, MD(CProj, J)
    Put #20, WNum + 3, MR(CProj, J)
    X = (Column(CProj, EleColRow(CProj, 0, I, J).Start).c1 + Column(CPr
oj, EleColRow(CProj, 0, I, J).Start).CapitalX) / 2
    LT = (EleColRow(CProj, 0, I, J).L1 + EleColRow(CProj, 0, I, J).L2)
/ 2
    W = (DeadCo * DeadLoad(CProj) + LiveCo * LiveLoad(CProj)) 'Ultimate
Load
    MDL = Cal_BSshear(J, 0) * X + ML(CProj, J) - ((W * LT * X ^ 2) / 2)
    Put #20, WNum + 4, MDL
    MDD = MD(CProj, J)
    Put #20, WNum + 5, MDD
    X = (Column(CProj, EleColRow(CProj, 0, I, J).Start).c1 + Column(CPr
oj, EleColRow(CProj, 0, I, J).Start).CapitalX) / 2
    LT = (EleColRow(CProj, 0, I, J).L1 + EleColRow(CProj, 0, I, J).L2)
    L = EleColRow(CProj, 0, I, J).Length
    W = (DeadCo * DeadLoad(CProj) + LiveCo * LiveLoad(CProj)) 'Ultimate
Load
    MDR = (-1) * Cal_BSshear(J, L) * X - MR(CProj, J) - ((W * LT * X ^ 2
) / 2)
    Put #20, WNum + 6, MDR
    WNum = WNum + 6
  Next J
Next I
For I = 1 To SCol(CProj)
  Trans_BElement True, I 'False is calculate in x axis

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FDESIGN.BAS - 7

```

Cal_FEndMoment True, I
For J = 1 To ColEle(I)
  Put #20, WNum + 1, ML(CProj, J)
  Put #20, WNum + 2, MD(CProj, J)
  Put #20, WNum + 3, MR(CProj, J)
  X = (Column(CProj, EleColRow(CProj, 1, I, J).Start).c1 + Column(CProj,
oj, EleColRow(CProj, 1, I, J).Start).CapitalX) / 2
  LT = (EleColRow(CProj, 1, I, J).L1 + EleColRow(CProj, 1, I, J).L2)
  W = (DeadCo * DeadLoad(CProj) + LiveCo * LiveLoad(CProj)) 'Ultimate
Load
  MDL = Cal_BSshear(J, 0) * X + ML(CProj, J) - (W * LT * X ^ 2) / 2)
  Put #20, WNum + 4, MDL
  MDD = MD(CProj, J)
  Put #20, WNum + 5, MDD
  X = (Column(CProj, EleColRow(CProj, 1, I, J).Start).c1 + Column(CProj,
oj, EleColRow(CProj, 1, I, J).Start).CapitalX) / 2
  LT = (EleColRow(CProj, 1, I, J).L1 + EleColRow(CProj, 1, I, J).L2)
  L = EleColRow(CProj, 1, I, J).Length
  W = (DeadCo * DeadLoad(CProj) + LiveCo * LiveLoad(CProj)) 'Ultimate
Load
  MDR = (-1) * Cal_BSshear(J, L) * X - MR(CProj, J) - (W * LT * X ^ 2
) / 2)
  Put #20, WNum + 6, MDR
  WNum = WNum + 6
Next J
Next I

'Distribution design moment at column strip & middle strip
RNum = 2 + 8 * SNodeNum(CProj)
For I = 1 To SRow(CProj)
  For J = 1 To RowEle(I)
    Get #20, RNum + 4, MDL
    Get #20, RNum + 5, MDD
    Get #20, RNum + 6, MDR
    RNum = RNum + 6

    '''' Find design moment at Row strip x axis

    '''' Left side
    'Interior
    If (Column(CProj, EleColRow(CProj, 0, I, J).Start).X_Flag + Column(
CProj, EleColRow(CProj, 0, I, J).Start).Y_Flag) = 0 Then
      MCL = .75 * MDL
      MCD = .6 * MDD
      MML1 = .125 * MDL
      MMD1 = .2 * MDD
      MML2 = .125 * MDL
      MMD2 = .2 * MDD
    'Corner
    ElseIf (Column(CProj, EleColRow(CProj, 0, I, J).Start).X_Flag <> 0)
And (Column(CProj, EleColRow(CProj, 0, I, J).Start).Y_Flag <> 0) Then
      MCL = MDL
      MCD = .6 * MDD
      MML1 = 0
      MMD1 = .4 * MDD
      MML2 = 0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FDESIGN.BAS - 8

```

MMD2 = .4 * MDD
'Edge Y
ElseIf (Column(CProj, EleColRow(CProj, 0, I, J).Start).X_Flag = 0)
And ((Column(CProj, EleColRow(CProj, 0, I, J).Start).Y_Flag = 1) Or (Column(CProj, EleColRow(CProj, 0, I, J).Start).Y_Flag = 2)) Then
MCL = .75 * MDL
MCD = .6 * MDD
MML1 = 0
MMD1 = 0
MML2 = 0
MMD2 = .4 * MDD
'Edge X
ElseIf (Column(CProj, EleColRow(CProj, 0, I, J).Start).Y_Flag = 0)
And ((Column(CProj, EleColRow(CProj, 0, I, J).Start).X_Flag = 1) Or (Column(CProj, EleColRow(CProj, 0, I, J).Start).X_Flag = 2)) Then
MCL = MDL
MCD = .6 * MDD
MML1 = 0
MMD1 = .2 * MDD
MML2 = 0
MMD2 = .2 * MDD
End If

''' Right side
'Interior
If (Column(CProj, EleColRow(CProj, 0, I, J).End).X_Flag + Column(CProj, EleColRow(CProj, 0, I, J).End).Y_Flag = 0) Then
MCR = .75 * MDR
MMR1 = .125 * MDR
MMR2 = .125 * MDR
'Corner
ElseIf (Column(CProj, EleColRow(CProj, 0, I, J).End).X_Flag <> 0) And (Column(CProj, EleColRow(CProj, 0, I, J).End).Y_Flag <> 0) Then
MCR = MDR
MMR1 = 0
MMR2 = 0
'Edge Y
ElseIf (Column(CProj, EleColRow(CProj, 0, I, J).End).X_Flag = 0) And ((Column(CProj, EleColRow(CProj, 0, I, J).End).Y_Flag = 1) Or (Column(CProj, EleColRow(CProj, 0, I, J).End).Y_Flag = 2)) Then
MCR = .75 * MDR
MMR1 = .25 * MDR
MMR2 = .25 * MDR
'Edge X
ElseIf (Column(CProj, EleColRow(CProj, 0, I, J).End).Y_Flag = 0) And ((Column(CProj, EleColRow(CProj, 0, I, J).End).X_Flag = 1) Or (Column(CProj, EleColRow(CProj, 0, I, J).End).X_Flag = 2)) Then
MCR = MDR
MMR1 = 0
MMR2 = 0
End If
Put #20, WNum + 1, MCL
Put #20, WNum + 2, MCD
Put #20, WNum + 3, MCR
Put #20, WNum + 4, MML1
Put #20, WNum + 5, MMD1
Put #20, WNum + 6, MMR1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FDESIGN.BAS - 9

```

Put #20, WNum + 7, MML2
Put #20, WNum + 8, MMD2
Put #20, WNum + 9, MMR2
WNum = WNum + 9
Next J
Next I

RNum = 2 + 8 * SNodeNum(CProj) + 6 * ElementXNum(CProj)
For I = 1 To SCol(CProj)
  For J = 1 To ColEle(I)
    Get #20, RNum + 4, MDL
    Get #20, RNum + 5, MDD
    Get #20, RNum + 6, MDR
    RNum = RNum + 6
    '''' Find design moment at Column strip y axis

    '''' Left side
    'Interior
    If (Column(CProj, EleColRow(CProj, 1, I, J).Start).X_Flag + Column(
CProj, EleColRow(CProj, 1, I, J).Start).Y_Flag) = 0 Then
      MCL = .75 * MDL
      MCD = .6 * MDD
      MML1 = .125 * MDL
      MMD1 = .2 * MDD
      MML2 = .125 * MDL
      MMD2 = .2 * MDD
    'Corner
    ElseIf (Column(CProj, EleColRow(CProj, 1, I, J).Start).X_Flag <> 0)
And (Column(CProj, EleColRow(CProj, 1, I, J).Start).Y_Flag <> 0) Then
      MCL = MDL
      MCD = .6 * MDD
      MML1 = 0
      MMD1 = .4 * MDD
      MML2 = 0
      MMD2 = .4 * MDD
    'Edge Y
    ElseIf (Column(CProj, EleColRow(CProj, 1, I, J).Start).X_Flag = 0)
And ((Column(CProj, EleColRow(CProj, 1, I, J).Start).Y_Flag = 1) Or (Column
n(CProj, EleColRow(CProj, 1, I, J).Start).Y_Flag = 2)) Then
      MCL = MDL
      MCD = .6 * MDD
      MML1 = 0
      MMD1 = .2 * MDD
      MML2 = 0
      MMD2 = .2 * MDD
    'Edge X
    ElseIf (Column(CProj, EleColRow(CProj, 1, I, J).Start).Y_Flag = 0)
And ((Column(CProj, EleColRow(CProj, 1, I, J).Start).X_Flag = 1) Or (Column
n(CProj, EleColRow(CProj, 1, I, J).Start).X_Flag = 2)) Then
      MCL = .75 * MDL
      MCD = .6 * MDD
      MML1 = .25 * MDL
      MMD1 = .4 * MDL
      MML2 = .25 * MDL
      MMD2 = .4 * MDD
    End If
  
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FDESIGN.BAS - 10

```

''' Right side
'Interior
If (Column(CProj, EleColRow(CProj, 1, I, J).End).X_Flag + Column(CP
roj, EleColRow(CProj, 1, I, J).End).Y_Flag) = 0 Then
MCR = .75 * MDR
MMR1 = .125 * MDR
MMR2 = .125 * MDR
'Corner
ElseIf (Column(CProj, EleColRow(CProj, 1, I, J).End).X_Flag <> 0) A
nd (Column(CProj, EleColRow(CProj, 1, I, J).End).Y_Flag <> 0) Then
MCR = MDR
MMR1 = 0
MMR2 = 0
'Edge Y
ElseIf (Column(CProj, EleColRow(CProj, 1, I, J).End).X_Flag = 0) An
d ((Column(CProj, EleColRow(CProj, 1, I, J).End).Y_Flag = 1) Or (Column(CP
roj, EleColRow(CProj, 1, I, J).End).Y_Flag = 2)) Then
MCR = MDR
MMR1 = 0
MMR2 = 0
'Edge X
ElseIf (Column(CProj, EleColRow(CProj, 1, I, J).End).Y_Flag = 0) An
d ((Column(CProj, EleColRow(CProj, 1, I, J).End).X_Flag = 1) Or (Column(CP
roj, EleColRow(CProj, 1, I, J).End).X_Flag = 2)) Then
MCR = .75 * MDR
MMR1 = .25 * MDR
MMR2 = .25 * MDR
End If
Put #20, WNum + 1, MCL
Put #20, WNum + 2, MCD
Put #20, WNum + 3, MCR
Put #20, WNum + 4, MML1
Put #20, WNum + 5, MMD1
Put #20, WNum + 6, MMR1
Put #20, WNum + 7, MML2
Put #20, WNum + 8, MMD2
Put #20, WNum + 9, MMR2
WNum = WNum + 9
Next J
Next I
Dim M As Single

''' set read record
RNum = 2 + 8 * SNodeNum(CProj) + 6 * ElementXNum(CProj) + 6 * Element
YNum(CProj)
'''find area of steel in longitudinal x axis
For I = 1 To SRow(CProj)
For J = 1 To RowEle(I)
'''Read design moment
Get #20, RNum + 1, MCL
Get #20, RNum + 2, MCD
Get #20, RNum + 3, MCR
Get #20, RNum + 4, MML1
Get #20, RNum + 5, MMD1
Get #20, RNum + 6, MMR1
Get #20, RNum + 7, MML2
Get #20, RNum + 8, MMD2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FDESIGN.BAS - 11

```

Get #20, RNum + 9, MMR2
RNum = RNum + 9
'column strip
b = ((EleColRow(CProj, 0, I, J).L1 / 4) + (EleColRow(CProj, 0, I, J
).L2 / 4))
Find_AsU1 b, MCL, CLT, CLB
Find_AsU1 b, MCD, CMT, CMB
Find_AsU1 b, MCR, CRT, CRB
Put #20, WNum + 1, CLB
Put #20, WNum + 2, CMB
Put #20, WNum + 3, CRB
Put #20, WNum + 4, CLT
Put #20, WNum + 5, CMT
Put #20, WNum + 6, CRT
'Half middle strip 1
b1 = (EleColRow(CProj, 0, I, J).L1 / 4)
Find_AsU1 b1, MML1, MLT1, MLB1
Find_AsU1 b1, MMD1, MMT1, MMB1
Find_AsU1 b1, MMR1, MRT1, MRB1
Put #20, WNum + 7, MLB1
Put #20, WNum + 8, MMB1
Put #20, WNum + 9, MRB1
Put #20, WNum + 10, MLT1
Put #20, WNum + 11, MMT1
Put #20, WNum + 12, MRT1
'Half middle strip 2
b2 = (EleColRow(CProj, 0, I, J).L2 / 4)
Find_AsU1 b2, MML2, MLT2, MLB2
Find_AsU1 b2, MMD2, MMT2, MMB2
Find_AsU1 b2, MMR2, MRT2, MRB2
Put #20, WNum + 13, MLB2
Put #20, WNum + 14, MMB2
Put #20, WNum + 15, MRB2
Put #20, WNum + 16, MLT2
Put #20, WNum + 17, MMT2
Put #20, WNum + 18, MRT2
WNum = WNum + 18
Next J
Next I
RNum = 2 + 8 * SNodeNum(CProj) + ElementXNum(CProj) * 6 + ElementYNum
(CProj) * 6 + 9 * ElementXNum(CProj)
'''find area of steel in Transverse Along Y axis
For I = 1 To Scol(CProj)
For J = 1 To Colele(I)
'''Read Design moment
Get #20, RNum + 1, MCL
Get #20, RNum + 2, MCD
Get #20, RNum + 3, MCR
Get #20, RNum + 4, MML1
Get #20, RNum + 5, MMD1
Get #20, RNum + 6, MMR1
Get #20, RNum + 7, MML2
Get #20, RNum + 8, MMD2
Get #20, RNum + 9, MMR2
RNum = RNum + 9
'column strip
b = ((EleColRow(CProj, 1, I, J).L1 / 4) + (EleColRow(CProj, 1, I, J

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FDESIGN.BAS - 12

```

).L2 / 4))
Find_AsU1 b, MCL, CLT, CLB
Find_AsU1 b, MCD, CMT, CMB
Find_AsU1 b, MCR, CRT, CRB
Put #20, WNum + 1, CLB
Put #20, WNum + 2, CMB
Put #20, WNum + 3, CRB
Put #20, WNum + 4, CLT
Put #20, WNum + 5, CMT
Put #20, WNum + 6, CRT
'Half middle strip 1
b1 = (EleColRow(CProj, 1, I, J).L1 / 4)
Find_AsU1 b1, MML1, MLT1, MLB1
Find_AsU1 b1, MMD1, MMT1, MMB1
Find_AsU1 b1, MMR1, MRT1, MRB1
Put #20, WNum + 7, MLB1
Put #20, WNum + 8, MMB1
Put #20, WNum + 9, MRB1
Put #20, WNum + 10, MLT1
Put #20, WNum + 11, MMT1
Put #20, WNum + 12, MRT1
'Half middle strip 2
b2 = (EleColRow(CProj, 1, I, J).L2 / 4)
Find_AsU1 b2, MML2, MLT2, MLB2
Find_AsU1 b2, MMD2, MMT2, MMB2
Find_AsU1 b2, MMR2, MRT2, MRB2
Put #20, WNum + 13, MLB2
Put #20, WNum + 14, MMB2
Put #20, WNum + 15, MRB2
Put #20, WNum + 16, MLT2
Put #20, WNum + 17, MMT2
Put #20, WNum + 18, MRT2
WNum = WNum + 18
Next J
Next I
'For I = 1 To 980
' Get #20, I, M
' frmMain.grdOut.Row = 1: frmMain.grdOut.Col = I
' frmMain.grdOut.ColWidth(I) = 1000
' frmMain.grdOut.Text = Format$(M)
'Next I

Close #20
Initial_Menu 4
End Sub

Sub Swap (V1 As Single, V2 As Single)
Dim Temp As Single
Temp = V1
V1 = V2
V2 = Temp
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 1

```

Sub Cal_ElasticEndMoment (ColOrRow As Integer, ColRow As Integer) 'deadload transfer
  Dim Pattern As Integer
  Dim axis As Integer
  Const ErrorCheck = 1
  Dim ErrorValue As Single
  Dim I As Integer
  If ColOrRow Then
    axis = 0
  ElseIf Not ColOrRow Then
    axis = 1
  End If
  ReDim FEML(ElementNum(CProj)) As Single
  ReDim FEMR(ElementNum(CProj)) As Single
  ReDim DFL(ElementNum(CProj)) As Single
  ReDim DFR(ElementNum(CProj)) As Single
  ReDim COML(ElementNum(CProj)) As Single
  ReDim COMR(ElementNum(CProj)) As Single
  ReDim DML(ElementNum(CProj)) As Single
  ReDim DMR(ElementNum(CProj)) As Single
  ReDim MNode(NodeNum(CProj)) As Single
  ReDim MMaxL(ElementNum(CProj)) As Single
  ReDim MMaxR(ElementNum(CProj)) As Single
  For I = 1 To ElementNum(CProj)
    MMaxL(I) = 0
    MMaxR(I) = 0
  Next I
  Pattern = 0
  For I = 1 To ElementNum(CProj)
    R = Cal_FK(I, Not (ColOrRow))
    FEML(I) = Cal_FFEM(axis, I, True, Pattern)
    FEMR(I) = Cal_FFEM(axis, I, False, Pattern)
    DFL(I) = Cal_FDF(I, Not (ColOrRow), True)
    DFR(I) = Cal_FDF(I, Not (ColOrRow), False)
    COFL(CProj, I) = COFL(CProj, I)
    COFR(CProj, I) = COFR(CProj, I)
    ML(CProj, I) = FEML(I)
    MR(CProj, I) = FEMR(I)
    DML(I) = 0
    DMR(I) = 0
    COML(I) = 0
    COMR(I) = 0
  Next I
  'Distribution Moment
  Do
    ErrorValue = 0
    'Sum DM
    For I = 1 To NodeNum(CProj)
      MNode(I) = 0
      For J = 1 To ElementNum(CProj)
        If (Element(CProj, J).Start) = I Then
          MNode(I) = (MNode(I) + ML(CProj, J))
        ElseIf (Element(CProj, J).End = I) Then
          MNode(I) = (MNode(I) + MR(CProj, J))
        End If
      Next J
    Next I
  Loop While ErrorValue > 0

```

OUTPUT.BAS - 2

```

Next J
Next I
'DM
For I = 1 To NodeNum(CProj)
  For J = 1 To ElementNum(CProj)
    If (Element(CProj, J).Start = I) Then
      DML(J) = (((-1) * MNode(I)) * DFL(J))
    ElseIf (Element(CProj, J).End = I) Then
      DMR(J) = (((-1) * MNode(I)) * DFR(J))
    End If
  Next J
Next I
For I = 1 To ElementNum(CProj)
  If Abs(ErrorValue) < Abs(DML(I)) Then
    ErrorValue = DML(I)
  End If
  If Abs(ErrorValue) < Abs(DMR(I)) Then
    ErrorValue = DMR(I)
  End If
Next I
'Carry over moment
For I = 1 To ElementNum(CProj)
  COML(I) = DMR(I) * COFL(CProj, I)
  COMR(I) = DML(I) * COFR(CProj, I)
Next I
'Sum Moment
If (Abs(ErrorValue) < ErrorCheck) Then
  For I = 1 To ElementNum(CProj)
    ML(CProj, I) = ML(CProj, I) + DML(I)
    MR(CProj, I) = MR(CProj, I) + DMR(I)
  Next I
Else
  For I = 1 To ElementNum(CProj)
    ML(CProj, I) = ML(CProj, I) + DML(I) + COML(I)
    MR(CProj, I) = MR(CProj, I) + DMR(I) + COMR(I)
  Next I
End If
Loop Until (Abs(ErrorValue) < ErrorCheck)
For I = 1 To ElementNum(CProj)
  If Abs(MMaxL(I)) < Abs(ML(CProj, I)) Then
    MMaxL(I) = ML(CProj, I)
  End If
  If Abs(MMaxR(I)) < Abs(MR(CProj, I)) Then
    MMaxR(I) = MR(CProj, I)
  End If
  MD(CProj, I) = (1 / 8) * ((DeadCo * DeadLoad(CProj)) + (1.8 * LiveLoad(CProj))) * ((Element(CProj, I).L1 + Element(CProj, I).L2) / 2) * (Element(CProj, I).Length) ^ 2
Next I
For I = 1 To ElementNum(CProj)
  ML(CProj, I) = MMaxL(I)
  MR(CProj, I) = MMaxR(I)
  MD(CProj, I) = MD(CProj, I) - (ML(CProj, I) + MR(CProj, I)) / 2
Next I

''' Show End Moment at Programmed
'Load frmOutText(CProj)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 3

```

'frmOutText(CProj).Show
'frmOutText(CProj).Cls
'For I = 1 To ElementNum(CProj)
' frmOutText(CProj).Print "Fixed end moment Left element" + Format$(I)
+ " = " + Format$(FEML(I), "0.00")
' frmOutText(CProj).Print "Fixed end moment Right element" + Format$(I)
) + " = " + Format$(FEMR(I), "0.00")
' frmOutText(CProj).Print "DF Factor Left " + Format$(DFL(I), "0.000")
' frmOutText(CProj).Print "DF Factor Right " + Format$(DFR(I), "0.000"
)
' frmOutText(CProj).Print "End Moment Left " + Format$(ML(CProj, I), "
0.00")
' frmOutText(CProj).Print "End Moment Right " + Format$(MR(CProj, I),
"0.00")
'Next I
End Sub

```

```

Sub Cal_ServiceEndMoment (ColOrRow As Integer, ColRow As Integer) 'at serv
ice load all load with no factor
Dim Pattern As Integer
Dim axis As Integer
Const ErrorCheck = 1
Dim ErrorValue As Single
Dim I As Integer
If ColOrRow Then
axis = 0
ElseIf Not ColOrRow Then
axis = 1
End If
ReDim FEML(ElementNum(CProj)) As Single
ReDim FEMR(ElementNum(CProj)) As Single
ReDim DFL(ElementNum(CProj)) As Single
ReDim DFR(ElementNum(CProj)) As Single
ReDim COML(ElementNum(CProj)) As Single
ReDim COMR(ElementNum(CProj)) As Single
ReDim DML(ElementNum(CProj)) As Single
ReDim DMR(ElementNum(CProj)) As Single
ReDim MNode(NodeNum(CProj)) As Single
ReDim MMaxL(ElementNum(CProj)) As Single
ReDim MMaxR(ElementNum(CProj)) As Single
For I = 1 To ElementNum(CProj)
MMaxL(I) = 0
MMaxR(I) = 0
Next I
Pattern = 5
For I = 1 To ElementNum(CProj)
R = Cal_FK(I, Not (ColOrRow))
FEML(I) = Cal_FFEM(axis, I, True, Pattern)
FEMR(I) = Cal_FFEM(axis, I, False, Pattern)
DFL(I) = Cal_FDF(I, Not (ColOrRow), True)
DFR(I) = Cal_FDF(I, Not (ColOrRow), False)
COFL(CProj, I) = COFL(CProj, I)
COFR(CProj, I) = COFR(CProj, I)
ML(CProj, I) = FEML(I)
MR(CProj, I) = FEMR(I)
DML(I) = 0
DMR(I) = 0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 4

```

COML(I) = 0
COMR(I) = 0
Next I
'Distribution Moment
Do
ErrorValue = 0
'Sum DM
For I = 1 To NodeNum(CProj)
MNode(I) = 0
For J = 1 To ElementNum(CProj)
If (Element(CProj, J).Start) = I Then
MNode(I) = (MNode(I) + ML(CProj, J))
ElseIf (Element(CProj, J).End = I) Then
MNode(I) = (MNode(I) + MR(CProj, J))
End If
Next J
Next I
'DM
For I = 1 To NodeNum(CProj)
For J = 1 To ElementNum(CProj)
If (Element(CProj, J).Start = I) Then
DML(J) = ((-1) * MNode(I)) * DFL(J)
ElseIf (Element(CProj, J).End = I) Then
DMR(J) = ((-1) * MNode(I)) * DFR(J)
End If
Next J
Next I
For I = 1 To ElementNum(CProj)
If Abs(ErrorValue) < Abs(DML(I)) Then
ErrorValue = DML(I)
End If
If Abs(ErrorValue) < Abs(DMR(I)) Then
ErrorValue = DMR(I)
End If
Next I
'Carry over moment
For I = 1 To ElementNum(CProj)
COML(I) = DMR(I) * COFL(CProj, I)
COMR(I) = DML(I) * COFR(CProj, I)
Next I
'Sum Moment
If (Abs(ErrorValue) < ErrorCheck) Then
For I = 1 To ElementNum(CProj)
ML(CProj, I) = ML(CProj, I) + DML(I)
MR(CProj, I) = MR(CProj, I) + DMR(I)
Next I
Else
For I = 1 To ElementNum(CProj)
ML(CProj, I) = ML(CProj, I) + DML(I) + COML(I)
MR(CProj, I) = MR(CProj, I) + DMR(I) + COMR(I)
Next I
End If
Loop Until (Abs(ErrorValue) < ErrorCheck)
For I = 1 To ElementNum(CProj)
If Abs(MMaxL(I)) < Abs(ML(CProj, I)) Then
MMaxL(I) = ML(CProj, I)
End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 5

```

    If Abs(MMaxR(I)) < Abs(MR(CProj, I)) Then
        MMaxR(I) = MR(CProj, I)
    End If
    MD(CProj, I) = (1 / 8) * ((DeadCo * DeadLoad(CProj)) + (1.8 * LiveLoa
d(CProj))) * ((Element(CProj, I).L1 + Element(CProj, I).L2) / 2) * (Elemen
t(CProj, I).Length) ^ 2
    Next I
    For I = 1 To ElementNum(CProj)
        ML(CProj, I) = MMaxL(I)
        MR(CProj, I) = MMaxR(I)
        MD(CProj, I) = MD(CProj, I) - (ML(CProj, I) + MR(CProj, I)) / 2
    Next I
End Sub

```

```

Sub Cal UltimateEndMoment (ColOrRow As Integer, ColRow As Integer) 'Ultima
te all load with factor
    Dim Pattern As Integer
    Dim axis As Integer
    Const ErrorCheck = 1
    Dim ErrorValue As Single
    Dim I As Integer
    If ColOrRow Then
        axis = 0
    ElseIf Not ColOrRow Then
        axis = 1
    End If
    ReDim FEML(ElementNum(CProj)) As Single
    ReDim FEMR(ElementNum(CProj)) As Single
    ReDim DFL(ElementNum(CProj)) As Single
    ReDim DFR(ElementNum(CProj)) As Single
    ReDim COML(ElementNum(CProj)) As Single
    ReDim COMR(ElementNum(CProj)) As Single
    ReDim DML(ElementNum(CProj)) As Single
    ReDim DMR(ElementNum(CProj)) As Single
    ReDim MNode(NodeNum(CProj)) As Single
    ReDim MMaxL(ElementNum(CProj)) As Single
    ReDim MMaxR(ElementNum(CProj)) As Single
    For I = 1 To ElementNum(CProj)
        MMaxL(I) = 0
        MMaxR(I) = 0
    Next I
    Pattern = 4
    For I = 1 To ElementNum(CProj)
        R = Cal_FK(I, Not (ColOrRow))
        FEML(I) = Cal_FFEM(axis, I, True, Pattern)
        FEMR(I) = Cal_FFEM(axis, I, False, Pattern)
        DFL(I) = Cal_FDF(I, Not (ColOrRow), True)
        DFR(I) = Cal_FDF(I, Not (ColOrRow), False)
        COFL(CProj, I) = COFL(CProj, I)
        COFR(CProj, I) = COFR(CProj, I)
        ML(CProj, I) = FEML(I)
        MR(CProj, I) = FEMR(I)
        DML(I) = 0
        DMR(I) = 0
        COML(I) = 0
        COMR(I) = 0
    Next I

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 6

```

'Distribution Moment
Do
ErrorValue = 0
'Sum DM
For I = 1 To NodeNum(CProj)
MNode(I) = 0
For J = 1 To ElementNum(CProj)
If (Element(CProj, J).Start = I) Then
MNode(I) = (MNode(I) + ML(CProj, J))
ElseIf (Element(CProj, J).End = I) Then
MNode(I) = (MNode(I) + MR(CProj, J))
End If
Next J
Next I
'DM
For I = 1 To NodeNum(CProj)
For J = 1 To ElementNum(CProj)
If (Element(CProj, J).Start = I) Then
DML(J) = (((-1) * MNode(I) * DFL(J))
ElseIf (Element(CProj, J).End = I) Then
DMR(J) = (((-1) * MNode(I) * DFR(J))
End If
Next J
Next I
For I = 1 To ElementNum(CProj)
If Abs(ErrorValue) < Abs(DML(I)) Then
ErrorValue = DML(I)
End If
If Abs(ErrorValue) < Abs(DMR(I)) Then
ErrorValue = DMR(I)
End If
Next I
'Carry over moment
For I = 1 To ElementNum(CProj)
COML(I) = DMR(I) * COFL(CProj, I)
COMR(I) = DML(I) * COFR(CProj, I)
Next I
'Sum Moment
If (Abs(ErrorValue) < ErrorCheck) Then
For I = 1 To ElementNum(CProj)
ML(CProj, I) = ML(CProj, I) + DML(I)
MR(CProj, I) = MR(CProj, I) + DMR(I)
Next I
Else
For I = 1 To ElementNum(CProj)
ML(CProj, I) = ML(CProj, I) + DML(I) + COML(I)
MR(CProj, I) = MR(CProj, I) + DMR(I) + COMR(I)
Next I
End If
Loop Until (Abs(ErrorValue) < ErrorCheck)
For I = 1 To ElementNum(CProj)
If Abs(MMaxL(I)) < Abs(ML(CProj, I)) Then
MMaxL(I) = ML(CProj, I)
End If
If Abs(MMaxR(I)) < Abs(MR(CProj, I)) Then
MMaxR(I) = MR(CProj, I)
End If

```

.OUTPUT.BAS - 7

```

    MD(CProj, I) = (1 / 8) * ((DeadCo * DeadLoad(CProj)) + (1.8 * LiveLoa
d(CProj))) * ((Element(CProj, I).L1 + Element(CProj, I).L2) / 2) * (Elemen
t(CProj, I).Length) ^ 2
    Next I
    For I = 1 To ElementNum(CProj)
        ML(CProj, I) = MMaxL(I)
        MR(CProj, I) = MMaxR(I)
        'MD(CProj, I) = MD(CProj, I) - (ML(CProj, I) + MR(CProj, I)) / 2 'try
    Next I
End Sub

```

```

Function Check_Deflection (COR As Integer, LNo As Integer) As Single
    Dim L As Single
    Dim La As Single
    If COR Then
        L = EleColRow(CProj, 1, LNo, 1).Length / 360
        For I = 1 To ColEle(LNo)
            La = EleColRow(CProj, 1, LNo, I).Length / 360
            If L < La Then
                L = La
            End If
        Next I
    ElseIf Not (COR) Then
        L = EleColRow(CProj, 0, LNo, 1).Length / 360
        For I = 1 To ColEle(LNo)
            La = EleColRow(CProj, 0, LNo, I).Length / 360
            If L < La Then
                L = La
            End If
        Next I
    End If
    Check_Deflection = L
End Function

```

```

Function Check_MnMax (s As Integer, b As Single) As Single
    Dim a As Single
    Dim Mn As Single
    Dim ss As Single
    Dim ds As Single
    ds = (Thick(CProj) - .025) * 100 'cm
    ss = .004 * b * (Thick(CProj) / 2) * 10000 'cm^2
    a = ((s * GeneralData(CProj).SFs_ * .775) + (GeneralData(CProj).RFs_ *
ss)) / (.85 * b * 100 * GeneralData(CProj).SFc) 'cm
    Mn = (s * GeneralData(CProj).SFs_ * .775 * (100 * Thick(CProj) - 2 * (
Thick(CProj) * 100 - 100 * d(CProj)) - (a / 2))) + (ss * GeneralData(CProj)
).RFs_ * (ds - (a / 2))) 'kg-m
    Check_MnMax = (.9 * Mn) / 100
End Function

```

```

Sub Find_BL (COR As Integer, ColRow As Integer, Lmax As Single, Bbeam As S
ingle)
    Lmax = 0
    Bbeam = 0
    If COR Then
        For I = 1 To ColEle(ColRow)
            If Lmax < EleColRow(CProj, 1, ColRow, I).Length Then
                Lmax = EleColRow(CProj, 1, ColRow, I).Length
            End If
        Next I
    End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 8

```

End If
If Bbeam < EleColRow(CProj, 1, ColRow, I).L1 / 2 + EleColRow(CProj,
1, ColRow, I).L2 / 2 Then
    Bbeam = EleColRow(CProj, 1, ColRow, I).L1 / 2 + EleColRow(CProj, 1,
ColRow, I).L2 / 2
End If
Next I
ElseIf Not (COR) Then
For I = 1 To RowEle(ColRow)
    If Lmax < EleColRow(CProj, 0, ColRow, I).Length Then
        Lmax = EleColRow(CProj, 0, ColRow, I).Length
    End If
    If Bbeam < EleColRow(CProj, 0, ColRow, I).L1 / 2 + EleColRow(CProj,
0, ColRow, I).L2 / 2 Then
        Bbeam = EleColRow(CProj, 0, ColRow, I).L1 / 2 + EleColRow(CProj, 0,
ColRow, I).L2 / 2
    End If
Next I
End If
End Sub

```

```

Function Find_MaxDeflection (COR As Integer, LNo As Integer, EIndex As Sin
gle) As Single
Dim W As Single
Dim Def As Single
Dim DefT As Single
Dim L As Single
Dim b As Single
Dim Ini As Single
Def = 0
If COR Then
    For I = 1 To ColEle(LNo)
        b = (EleColRow(CProj, 1, LNo, I).L1 / 2) + (EleColRow(CProj, 1, LNo,
I).L2 / 2)
        L = EleColRow(CProj, 1, LNo, I).Length
        If EIndex = 1 Then 'after tension
            W = (DeadLoad(CProj) + SuperLoad(CProj)) * b
        ElseIf EIndex = 2 Then 'during use
            W = (LiveLoad(CProj) + SuperLoad(CProj)) * b
        End If
        Ini = (1 / 12) * b * Thick(CProj) ^ 3
        If I = 1 Or I = ColEle(LNo) Then
            DefT = (3.8 * W * L ^ 4) / (384 * GeneralData(CProj).SEc * Ini)
        Else
            DefT = (2.6 * W * L ^ 4) / (384 * GeneralData(CProj).SEc * Ini)
        End If
        If Def < DefT Then
            Def = DefT
        End If
    Next I
ElseIf Not COR Then
    For I = 1 To RowEle(LNo)
        b = (EleColRow(CProj, 0, LNo, I).L1 / 2) + (EleColRow(CProj, 0, LNo,
I).L2 / 2)
        L = EleColRow(CProj, 0, LNo, I).Length
        If EIndex = 1 Then 'after tension
            W = (DeadLoad(CProj) + SuperLoad(CProj)) * b

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 9

```

ElseIf EIndex = 2 Then 'during use
  W = (LiveLoad(CProj) + SuperLoad(CProj)) * b
End If
Ini = (1 / 12) * b * Thick(CProj) ^ 3
If I = 1 Or I = RowEle(I.No) Then
  DefT = (3.8 * W * L ^ 4) / (384 * GeneralData(CProj).SEC * Ini)
Else
  DefT = (2.6 * W * L ^ 4) / (384 * GeneralData(CProj).SEC * Ini)
End If
If Def < DefT Then
  Def = DefT
End If
Next I
End If
Find_MaxDeflection = Def / 100000'unit in meter
End Function

```

```

Function Find_MuMax (COR As Integer, LNo As Integer) As Single
Dim MMax As Single
If COR Then
  MMax = Abs(ML(CProj, 1))
  For I = 1 To ColeEle(LNo)
    If MMax < Abs(ML(CProj, I)) Then
      MMax = Abs(ML(CProj, I))
    End If
    If MMax < Abs(MR(CProj, I)) Then
      MMax = Abs(MR(CProj, I))
    End If
    If MMax < Abs(MD(CProj, I)) Then
      MMax = Abs(MD(CProj, I))
    End If
  Next I
ElseIf Not COR Then
  MMax = Abs(ML(CProj, 1))
  For I = 1 To RowEle(LNo)
    If MMax < Abs(ML(CProj, I)) Then
      MMax = Abs(ML(CProj, I))
    End If
    If MMax < Abs(MR(CProj, I)) Then
      MMax = Abs(MR(CProj, I))
    End If
    If MMax < Abs(MD(CProj, I)) Then
      MMax = Abs(MD(CProj, I))
    End If
  Next I
End If
Find_MuMax = Abs(MMax)
End Function

```

```

Sub Graph_BMD (AddSect As Single)
Dim SumL As Single
Dim MMax As Single
Dim MMin As Single
Dim StartPoint As Single
Dim EndPoint As Single

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 10

```

Dim Mx As Single
Dim I As Integer
Dim J As Single
Dim LineColor As Integer
Dim TwipsPerMeter As Integer
LineColor = 14
'find best dimension
SumL = 0
For I = 1 To ElementNum(CProj)
    SumL = SumL + Element(CProj, I).Length
Next I
TwipsPerMeter = (frmOutGraph(CProj).ScaleWidth - 1500) / SumL
'Find M Max
MMax = 0
MMin = 0
For I = 1 To ElementNum(CProj)
    For J = 0 To Element(CProj, I).Length Step AddSect
        Mx = Cal_BMoment(I, J, ML(CProj, I), MR(CProj, I))
        If Mx > MMax Then MMax = Mx
        If Mx < MMin Then MMin = Mx
    Next J
Next I
'find best amplitude
TwipsPerMoment = ((frmOutGraph(CProj).ScaleHeight / 2) - 1500) / (MMax - MMin)
frmOutGraph(CProj).PSet (frmOutGraph(CProj).lneMxX.X1, frmOutGraph(CProj).lneMxX.Y1), Colors(LineColor)
For I = 1 To ElementNum(CProj)
    For J = 0 To Element(CProj, I).Length Step AddSect
        Mx = Cal_BMoment(I, J, ML(CProj, I), MR(CProj, I)) * TwipsPerMoment
        frmOutGraph(CProj).Line - (frmOutGraph(CProj).lneMxX.X1 + ((Node(CProj, Element(CProj, I).Start).Co_X) + J) * TwipsPerMeter), frmOutGraph(CProj).lneMxX.Y1 - Mx), Colors(LineColor)
        frmOutGraph(CProj).Line (frmOutGraph(CProj).lneMxX.X1 + ((Node(CProj, Element(CProj, I).Start).Co_X) + J) * TwipsPerMeter), frmOutGraph(CProj).lneMxX.Y1 - (frmOutGraph(CProj).lneMxX.X1 + ((Node(CProj, Element(CProj, I).Start).Co_X) + J) * TwipsPerMeter), frmOutGraph(CProj).lneMxX.Y1 - Mx), Colors(LineColor)
    Next J
Next I
End Sub

Sub Graph_Profile (COR As Integer, ColRow As Integer, AddSect As Single, T
PMX As Integer, TPP As Integer, InitX As Integer, Inity As Integer) '1 Line Column Or Row
Dim SumL As Single
Dim Thick As Single
Dim StartPoint As Single
Dim EndPoint As Single
Dim PF As Single
Dim I As Integer
Dim J As Single
Dim LineColor As Integer
Dim X As Single
Dim Y As Single
Dim X1 As Integer
Dim Y1 As Integer

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 11

```

Dim X2 As Integer
Dim Y2 As Integer
Dim e As Integer
Dim ElementNum As Integer

''' Init Value
LineColor = 37
SumL = 0
Trans_Element 'Design Column I
Trans_BEElement COR, ColRow 'Trans To Beam Element at ColRow Column Or
Row
'TPP = 2000 / 30
'If COR Then
' ElementNum = ColEle(ColRow)
'ElseIf Not COR Then
' ElementNum = RowEle(ColRow)
'End If
'For I = 1 To ElementNum
' SumL = SumL + Element(CProj, I).Length
'Next I
'SumL = 0
'TPM = frmOutGraph(CProj).Width \ SumL
SumL = 0
If COR Then
For I = 1 To ColEle(ColRow)
For J = 0 To Element(CProj, I).Length Step AddSect
'PF = Cal_Profile(I, J, .1, .1, emax(CProj))
X1 = InitX
Y1 = InitY - ((SumL + J) * TPM)
X2 = InitX - (PF * TPP)
Y2 = InitY - ((SumL + J) * TPM)
frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(LineColor)
Next J
SumL = SumL + Element(CProj, I).Length
Next I
ElseIf Not COR Then
For I = 1 To RowEle(ColRow)
For J = 0 To Element(CProj, I).Length Step AddSect
'PF = Cal_Profile(I, J, .1, .1, emax(CProj))
X1 = InitX + ((SumL + J) * TPM)
Y1 = InitY
X2 = InitX + ((SumL + J) * TPM)
Y2 = InitY - (PF * TPP)
frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(LineColor)
Next J
SumL = SumL + Element(CProj, I).Length
Next I
End If
Sub Graph_BMD (AddSect As Single)
' Dim SumL As Single
' Dim MMax As Single
' Dim MMin As Single
' Dim StartPoint As Single
' Dim EndPoint As Single
' Dim Mx As Single
' Dim I As Integer
' Dim J As Single

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 12

```

'Dim LineColor As Integer
'Dim TwipsPerMeter As Integer
' LineColor = 14
' 'find best dimension
' SumL = 0
'For I = 1 To ElementNum(CProj)
' SumL = SumL + Element(CProj, I).Length
' Next I
' TwipsPerMeter = (frmOutGraph(CProj).ScaleWidth - 1500) / SumL
' Find M Max
'MMax = 0
'MMin = 0
' For I = 1 To ElementNum(CProj)
'   For J = 0 To Element(CProj, I).Length Step AddSect
'     Mx = Cal_BMoment(I, J, ML(CProj, I), MR(CProj, I))
'     If Mx > MMax Then MMax = Mx
'     If Mx < MMin Then MMin = Mx
'   Next J
' Next I
''find best amplitude
' TwipsPerMoment = ((frmOutGraph(CProj).ScaleHeight / 2) - 1500) / (MMax
x - MMin)
' frmOutGraph(CProj).PSet (frmOutGraph(CProj).lneMxX.X1, frmOutGraph(CP
roj).lneMxX.Y1), Colors(LineColor)
' For I = 1 To ElementNum(CProj)
'   For J = 0 To Element(CProj, I).Length Step AddSect
'     Mx = Cal_BMoment(I, J, ML(CProj, I), MR(CProj, I)) * TwipsPerMoment
'     frmOutGraph(CProj).Line - (frmOutGraph(CProj).lneMxX.X1 + ((Node(CP
roj, Element(CProj, I).Start).Co_X) + J) * TwipsPerMeter), frmOutGraph(CPr
oj).lneMxX.Y1 - Mx), Colors(LineColor)
'     frmOutGraph(CProj).Line (frmOutGraph(CProj).lneMxX.X1 + ((Node(CPr
oj, Element(CProj, I).Start).Co_X) + J) * TwipsPerMeter), frmOutGraph(CPro
j).lneMxX.Y1) - (frmOutGraph(CProj).lneMxX.X1 + ((Node(CProj, Element(CProj
, I).Start).Co_X) + J) * TwipsPerMeter), frmOutGraph(CProj).lneMxX.Y1 - Mx
), Colors(LineColor)
'   Next J
' Next I
'End Sub

```

End Sub

```

Sub Graph_SFD (AddSect As Single)
Dim SumL As Single
Dim VMax As Single
Dim VMin As Single
Dim StartPoint As Single
Dim EndPoint As Single
Dim Vx As Single
Dim v As Single
Dim I As Integer
Dim J As Single
Dim LineColor As Integer
Dim TwipsPerMeter As Integer
LineColor = 34
SumL = 0
For I = 1 To ElementNum(CProj)
SumL = SumL + Element(CProj, I).Length

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 13

```

Next I
TwipsPerMeter = (frmOutGraph(CProj).ScaleWidth - 1500) / SumL
'Find V Max & Min
VMax = 0
VMin = 0
For I = 1 To ElementNum(CProj)
  For J = 0 To Element(CProj, I).Length Step AddSect
    Vx = Cal_BShear(I, J)
    If Vx > VMax Then VMax = Vx
    If Vx < VMin Then VMin = Vx
  Next J
Next I
'find best amplitude
TwipsPerShear = ((frmOutGraph(CProj).ScaleHeight / 2) - 1500) / (VMax
- VMin)
frmOutGraph(CProj).PSet (frmOutGraph(CProj).lneSxX.X1, frmOutGraph(CPr
oj).lneSxX.Y1), Colors(LineColor)
For I = 1 To ElementNum(CProj)
  For J = 0 To Element(CProj, I).Length Step AddSect
    Vx = Cal_BShear(I, J) * TwipsPerShear
    frmOutGraph(CProj).Line -(frmOutGraph(CProj).lneSxX.X1 + ((Node(CProj
, Element(CProj, I).Start).Co_X + J) * TwipsPerMeter), frmOutGraph(CProj)
.lneSxX.Y1 - Vx), Colors(LineColor)
    frmOutGraph(CProj).Line (frmOutGraph(CProj).lneSxX.X1 + ((Node(CProj
, Element(CProj, I).Start).Co_X + J) * TwipsPerMeter), frmOutGraph(CProj).
lneSxX.Y1)-(frmOutGraph(CProj).lneSxX.X1 + ((Node(CProj, Element(CProj, I)
.Start).Co_X + J) * TwipsPerMeter), frmOutGraph(CProj).lneSxX.Y1 - Vx), Co
lors(LineColor)
  Next J
Next I
End Sub

Sub List_Compute ()
End Sub

Sub List_DataInput ()
End Sub

Sub List_EndMoment ()
End Sub

Sub List_Loss (SType As Integer, PType As Integer, TimeApply As Integer, f
cpi As Single, fcpa As Single, fc As Single, fcds As Single, Volume As Sin
gle, Surface As Single)
'Import data is
'1. SType = StrandType
'2. fcpi
'3. fcpa
'4. fc
'5. fcds
'6. TimeApply = Time apply load
Dim Pxl As Single
Dim Al As Single '
Dim L As Single '

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 14

```

Dim Esl As Single
Dim Kes As Single
Dim Es As Single
Dim Eci As Single
Dim SH As Single
Dim Ksh As Single
Dim CR As Single
Dim RH As Single
Dim Kcir As Single
Dim fcir As Single
Dim fg As Single
Dim RE As Single
Dim Kre As Single
'Compute require data
Select Case TimeApply
Case 1: Ksh = .92
Case 3: Ksh = .85
Case 5: Ksh = .8
Case 7: Ksh = .77
Case 10: Ksh = .73
Case 20: Ksh = .64
Case 30: Ksh = .58
Case 60: Ksh = .45
End Select
Select Case PType
Case 0: 'Post-tensioned member
Kcr = 1.6
Kes = .5
Kcir = 1
Case 1: 'Pre-tensioned member
Kcr = 1.6
Kes = .5
Kcir = 1
End Select
Eci = 4730 * (fcir ^ (1 / 2)) 'fcir determine from dead load only
Ec = 4730 * (GeneralData(CProj).SFC_ ^ (1 / 2))
Es = GeneralData(CProj).SEs
fcir = (Kcir * fcpi) - fg
RH = 70
'Friction Loss
Pxl = (Al * UfrictionAngle(CProj) + L * KfrictionLength(CProj))
'Elastic Shortening of concrete
Select Case SType
Case 0: 'BOND
Esl = Kes * (Es / Eci) * fcir 'fcir = (Kcir*fcpi)-fg,Kcir = 1
Case 1: 'UNBOND
Esl = Kes * (Es / Eci) * fc 'fc = Average stress ,Kes = 0.5
End Select
'Shrinkage of concrete (SH)
SH = ((8.2) / 1000000) * Ksh * Es * (1 - .024 * (Volume / Surface)) *
(100 - 70) '100-RH ~ 30
'Creep of concrete (CR)
Select Case SType
Case 0: 'BOND
CR = Kcr * (Es / Ec) * fcpa 'fcpa = stress average between end & At' p
oint
Case 1: 'UNBOND

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 15

```
CR = Kcr * (Es / Ec) * (fcir - fcds) 'fcds = stress from Dead Load (D
.L.)
```

```
End Select
```

```
'Relaxation (RE)
```

```
RE = (Kre - (J * (SH + CR + Es1))) * c
```

```
End Sub
```

```
Sub List_NodeReaction ()
```

```
End Sub
```

```
Sub List_Profile ()
```

```
End Sub
```

```
Sub Print_Geometric ()
```

```
frmMain.cmdlgSystem.Flags = PD_DISABLEPRINTTOFILE
```

```
frmMain.cmdlgSystem.CancelError = True
```

```
On Error GoTo DlgErrorGeometric
```

```
frmMain.cmdlgSystem.Action = 5
```

```
Dim XMax As Integer
```

```
Dim XMin As Integer
```

```
Dim YMax As Integer
```

```
Dim YMin As Integer
```

```
Dim InitX As Integer
```

```
Dim InitY As Integer
```

```
Dim Col As Integer
```

```
Dim CStart As Integer
```

```
Dim CEnd As Integer
```

```
Dim TPMX As Single
```

```
Dim TPMY As Single
```

```
Dim TPP As Single
```

```
Dim TPM As Single
```

```
Dim AddSect As Single
```

```
Dim X1 As Single
```

```
Dim Y1 As Single
```

```
Dim X2 As Single
```

```
Dim Y2 As Single
```

```
Dim X As Single
```

```
Dim Y As Single
```

```
Dim I As Integer
```

```
Dim J As Integer
```

```
Dim Init1 As Integer
```

```
Dim Init2 As Integer
```

```
Dim PF As Single
```

```
Dim M As Single
```

```
Dim Wu As Single
```

```
Dim ColumnColor As Integer
```

```
Dim ElementColor As Integer
```

```
Dim ProfileColor As Integer
```

```
'''
```

```
XMax = Column(CProj, 1).Co_X
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 16

```

XMin = Column(CProj, 1).Co_X
YMax = Column(CProj, 1).Co_Y
YMin = Column(CProj, 1).Co_Y
For I = 1 To SNodeNum(CProj)
  If XMax < Column(CProj, I).Co_X Then
    XMax = Column(CProj, I).Co_X
  End If
  If XMin > Column(CProj, I).Co_X Then
    XMin = Column(CProj, I).Co_X
  End If
  If YMax < Column(CProj, I).Co_Y Then
    YMax = Column(CProj, I).Co_Y
  End If
  If YMin > Column(CProj, I).Co_Y Then
    YMin = Column(CProj, I).Co_Y
  End If
Next I

''' Init Value
frmOutGraph(CProj).Cls
Col = 50 'Column Width in Twips
AddSect = .1 'Additional Section
InitX = 1000 'Left Edge in Twips
InitY = 750 'Top Edge in Twips
ColumnColor = 34
ElementColor = 30
ProfileColor = 27
MomentColor = 17
Printer.PSet (3000, 250)
Printer.Print " SLAB TO DESIGN "
Printer.PSet (3000, 350)
Printer.Print "===== "

Wu = DeadCo * DeadLoad(CProj) + LiveCo * LiveLoad(CProj)

'''
Trans_UniformLoad (Wu)'Trans Load

'''
TPMX = (Printer.Width - (InitX * 2)) \ (XMax - XMin)'1m = TPMX Twips f
or X axis
TPMY = (Printer.Height - (InitY * 2)) \ (YMax - YMin)'1m = TPMY Twips
for Y axis
TPP = 2000 / .3 'profile 30 cm use 2000 Twips '1m =TPP Twips Fo
r Profile Value ,Maximum = Maximum Thickness/2
TPM = .01
If TPMX < TPMY Then
  TPMY = TPMX
ElseIf TPMX >= TPMY Then
  TPMX = TPMY
End If
''' Draw Column
For I = 1 To SNodeNum(CProj)
  X1 = ((Column(CProj, I).Co_X - MinX) * TPMX) - Col + InitX
  Y1 = ((Column(CProj, I).Co_Y - MinY) * TPMY) - Col + InitY
  X2 = ((Column(CProj, I).Co_X - MinX) * TPMX) + Col + InitX
  Y2 = ((Column(CProj, I).Co_Y - MinY) * TPMY) + Col + InitY

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 17

```
Printer.Line (X1, Y1)-(X2, Y2), Colors(ColumnColor), BF
Printer.Print Format$(I, "0") + "[" + Format$(Column(CProj, I).Co_X,
"#.00") + "," + Format$(Column(CProj, I).Co_Y, "#.00") + "]"
Next I
```

```
'''' Draw Element Conjunction in X axis
```

```
For I = 1 To SRow(CProj)
```

```
For J = 1 To RowEle(I)
```

```
  CStart = EleColRow(CProj, 0, I, J).Start
```

```
  CEnd = EleColRow(CProj, 0, I, J).End
```

```
  X1 = ((Column(CProj, CStart).Co_X - MinX) * TPMX) + InitX
```

```
  Y1 = ((Column(CProj, CStart).Co_Y - MinY) * TPMY) + InitY
```

```
  X2 = ((Column(CProj, CEnd).Co_X - MinX) * TPMX) + InitX
```

```
  Y2 = ((Column(CProj, CEnd).Co_Y - MinY) * TPMY) + InitY
```

```
  Printer.Line (X1, Y1)-(X2, Y2), Colors(ElementColor)
```

```
Next J
```

```
Next I
```

```
'''' Draw Element Conjunction in Y axis
```

```
For I = 1 To SCol(CProj)
```

```
For J = 1 To Colele(I)
```

```
  CStart = EleColRow(CProj, 1, I, J).Start
```

```
  CEnd = EleColRow(CProj, 1, I, J).End
```

```
  X1 = ((Column(CProj, CStart).Co_X - MinX) * TPMX) + InitX
```

```
  Y1 = ((Column(CProj, CStart).Co_Y - MinY) * TPMY) + InitY
```

```
  X2 = ((Column(CProj, CEnd).Co_X - MinX) * TPMX) + InitX
```

```
  Y2 = ((Column(CProj, CEnd).Co_Y - MinY) * TPMY) + InitY
```

```
  Printer.Line (X1, Y1)-(X2, Y2), Colors(ElementColor)
```

```
Next J
```

```
Next I
```

```
Printer.EndDoc
```

```
Exit Sub
```

```
DlgErrorGeometric: Exit Sub
```

```
End Sub
```

```
Sub Print_Layout ()
```

```
frmMain.cmdlgSystem.Flags = PD_DISABLEPRINTTOFILE
```

```
frmMain.cmdlgSystem.CancelError = True
```

```
On Error GoTo DlgErrorLayOut
```

```
frmMain.cmdlgSystem.Action = 5
```

```
Dim XMax As Integer
```

```
Dim XMin As Integer
```

```
Dim YMax As Integer
```

```
Dim YMin As Integer
```

```
Dim InitX As Integer
```

```
Dim InitY As Integer
```

```
Dim Col As Integer
```

```
Dim CStart As Integer
```

```
Dim CEnd As Integer
```

```
Dim TPMX As Single
```

```
Dim TPMY As Single
```

```
Dim TPP As Single
```

```
Dim TPM As Single
```

```
Dim AddSect As Single
```

```
Dim X1 As Single
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 18

```

Dim Y1 As Single
Dim X2 As Single
Dim Y2 As Single
Dim X As Single
Dim Y As Single
Dim Init1 As Integer
Dim Init2 As Integer
Dim PF As Single
Dim M As Single
Dim Wu As Single
Dim I As Integer
Dim J As Integer
Dim ColumnColor As Integer
Dim ElementColor As Integer
Dim ProfileColor As Integer

```

```

''''

```

```

XMax = Column(CProj, 1).Co_X
XMin = Column(CProj, 1).Co_X
YMax = Column(CProj, 1).Co_Y
YMin = Column(CProj, 1).Co_Y
For I = 1 To SNodeNum(CProj)
  If XMax < Column(CProj, I).Co_X Then
    XMax = Column(CProj, I).Co_X
  End If
  If XMin > Column(CProj, I).Co_X Then
    XMin = Column(CProj, I).Co_X
  End If
  If YMax < Column(CProj, I).Co_Y Then
    YMax = Column(CProj, I).Co_Y
  End If
  If YMin > Column(CProj, I).Co_Y Then
    YMin = Column(CProj, I).Co_Y
  End If
Next I

```

```

'''' Init Value

```

```

Printer.Cls
Col = 50 'Column Width in Twips
AddSect = .1 'Additional Section
InitX = 1000 'Left Edge in Twips
InitY = 2000 'Top Edge in Twips
ColumnColor = 34
ElementColor = 30
ProfileColor = 27
MomentColor = 17
Wu = DeadCo * DeadLoad(CProj) + LiveCo * LiveLoad(CProj)

```

```

''''

```

```

Trans_UniformLoad (Wu)'Trans Load

```

```

''''

```

```

TPMX = (Printer.Width - (InitX * 2)) \ (XMax - XMin)'1m = TPMX Twips f
or X axis
TPMY = (Printer.Height - (InitY * 2)) \ (YMax - YMin)'1m = TPMY Twips
for Y axis
TPP = 2000 / .3 'profile 30 cm use 2000 Twips '1m =TPP Twips Fo

```

OUTPUT.BAS - 19

```

r Profile Value ,Maximum = Maximum Thickness/2
  TPM = .01
  If TPMX < TPMY Then
    TPMY = TPMX
  ElseIf TPMX >= TPMY Then
    TPMX = TPMY
  End If

  '''' Draw Column
  For I = 1 To SNodeNum(CProj)
    X1 = ((Column(CProj, I).Co_X - MinX) * TPMX) - Col + InitX
    Y1 = ((Column(CProj, I).Co_Y - MinY) * TPMY) - Col + InitY
    X2 = ((Column(CProj, I).Co_X - MinX) * TPMX) + Col + InitX
    Y2 = ((Column(CProj, I).Co_Y - MinY) * TPMY) + Col + InitY
    Printer.Line (X1, Y1)-(X2, Y2), Colors(ColumnColor), BF
  Next I

  '''' Draw Element Conjunction in X axis
  For I = 1 To SRow(CProj)
    For J = 1 To RowEle(I)
      CStart = EleColRow(CProj, 0, I, J).Start
      CEnd = EleColRow(CProj, 0, I, J).End
      X1 = ((Column(CProj, CStart).Co_X - MinX) * TPMX) + InitX
      Y1 = ((Column(CProj, CStart).Co_Y - MinY) * TPMY) + InitY
      X2 = ((Column(CProj, CEnd).Co_X - MinX) * TPMX) + InitX
      Y2 = ((Column(CProj, CEnd).Co_Y - MinY) * TPMY) + InitY
      Printer.Line (X1, Y1)-(X2, Y2), Colors(ElementColor)
    Next J
  Next I

  '''' Draw Element Conjunction in Y axis
  For I = 1 To SCol(CProj)
    For J = 1 To ColeEle(I)
      CStart = EleColRow(CProj, 1, I, J).Start
      CEnd = EleColRow(CProj, 1, I, J).End
      X1 = ((Column(CProj, CStart).Co_X - MinX) * TPMX) + InitX
      Y1 = ((Column(CProj, CStart).Co_Y - MinY) * TPMY) + InitY
      X2 = ((Column(CProj, CEnd).Co_X - MinX) * TPMX) + InitX
      Y2 = ((Column(CProj, CEnd).Co_Y - MinY) * TPMY) + InitY
      Printer.Line (X1, Y1)-(X2, Y2), Colors(ElementColor)
    Next J
  Next I

  '''' Draw Profile in X axis
  Init1 = InitX
  Init2 = InitY
  For I = 1 To SRow(CProj)
    Trans_BElement False, I
    Cal_FEndMoment False, I
    For J = 1 To RowEle(I)
      InitX = ((Column(CProj; EleColRow(CProj, 0, I, J).Start).Co_X - MinX
) * TPMX) + Init1
      InitY = ((Column(CProj, EleColRow(CProj, 0, I, J).Start).Co_Y - MinY
) * TPMY) + Init2
      Printer.PSet (InitX, InitY), ProfileColor
      For X = 0 To EleColRow(CProj, 0, I, J).Length Step AddSect
        M = Cal_BMoment(J, X, ML(CProj, J), MR(CProj, J))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 20

```

PF = Cal_Profile(False, I, J, X, 1, 1, émax(CProj))
X1 = IniEX + (X * TPMX)
Y1 = InitY
X2 = InitX + (X * TPMX)
Y2 = InitY - (M * TPM)
Printer.Line -(X2, Y2), Colors(MomentColor)
Printer.Line (X1, Y1)-(X2, Y2), Colors(MomentColor)

X1 = InitX + (X * TPMX)
Y1 = InitY
X2 = InitX + (X * TPMX)
Y2 = InitY - (PF * TPP)
Printer.Line -(X2, Y2), Colors(ProfileColor)
Printer.Line (X1, Y1)-(X2, Y2), Colors(ProfileColor)
Next X
Next J
Next I

'''' Draw Profile in Y axis
For I = 1 To SCol(CProj)
  Trans BElement True, I
  Cal_FEndMoment True, I
  For J = 1 To Colele(I)
    InitX = (Column(CProj, EleColRow(CProj, 1, I, J).Start).Co_X - MinX)
* TPMX + Init1
    InitY = (Column(CProj, EleColRow(CProj, 1, I, J).Start).Co_Y - MinY)
* TPMY + Init2
    Printer.PSet (InitX, InitY), ProfileColor
    For Y = 0 To EleColRow(CProj, 1, I, J).Length Step AddSect
      M = Cal_BMoment(J, Y, ML(CProj, J), MR(CProj, J))
      PF = Cal_Profile(True, I, J, Y, 1, 1, émax(CProj))
      X1 = InitX
      Y1 = InitY + (Y * TPMY)
      X2 = InitX - (M * TPM)
      Y2 = InitY + (Y * TPMY)
      Printer.Line -(X2, Y2), Colors(MomentColor)
      Printer.Line (X1, Y1)-(X2, Y2), Colors(MomentColor)

      X1 = InitX
      Y1 = InitY + (Y * TPMY)
      X2 = InitX - (PF * TPP)
      Y2 = InitY + (Y * TPMY)
      Printer.Line -(X2, Y2), Colors(ProfileColor)
      Printer.Line (X1, Y1)-(X2, Y2), Colors(ProfileColor)
    Next Y
  Next J '820438
Next I
Printer.EndDoc
Exit Sub
DlgErrorLayOut: Exit Sub
End Sub

Sub Print Moment ()
frmMain.cmdlgSystem.Flags = PD_DISABLEPRINTTOFILE
frmMain.cmdlgSystem.CancelError = True
On Error GoTo DlgErrorMoment
frmMain.cmdlgSystem.Action = 5

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 21

```

Dim XMax As Integer
Dim XMin As Integer
Dim YMax As Integer
Dim YMin As Integer
Dim InitX As Integer
Dim InitY As Integer
Dim Col As Integer
Dim CStart As Integer
Dim CEnd As Integer
Dim TPMX As Single
Dim TPMY As Single
Dim TPP As Single
Dim TPM As Single
Dim AddSect As Single
Dim X1 As Single
Dim Y1 As Single
Dim X2 As Single
Dim Y2 As Single
Dim X As Single
Dim Y As Single
Dim I As Integer
Dim J As Integer

Dim Init1 As Integer
Dim Init2 As Integer
Dim PF As Single
Dim M As Single
Dim Wu As Single
Dim ColumnColor As Integer
Dim ElementColor As Integer
Dim ProfileColor As Integer

''''
XMax = Column(CProj, 1).Co_X
XMin = Column(CProj, 1).Co_X
YMax = Column(CProj, 1).Co_Y
YMin = Column(CProj, 1).Co_Y
For I = 1 To SNodeNum(CProj)
  If XMax < Column(CProj, I).Co_X Then
    XMax = Column(CProj, I).Co_X
  End If
  If XMin > Column(CProj, I).Co_X Then
    XMin = Column(CProj, I).Co_X
  End If
  If YMax < Column(CProj, I).Co_Y Then
    YMax = Column(CProj, I).Co_Y
  End If
  If YMin > Column(CProj, I).Co_Y Then
    YMin = Column(CProj, I).Co_Y
  End If
Next I

'''' Init Value
'Printer.Cls

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 22

```

Col = 50      'Column Width in Twips
AddSect = .1 'Additional Section
InitX = 1000 'Left Edge in Twips
InitY = 750  'Top Edge in Twips
ColumnColor = 34
ElementColor = 30
ProfileColor = 27
MomentColor = 17
Printer.PSet (3000, 250)
Printer.Print " MOMENT DIAGRAM  "
Printer.PSet (3000, 350)
Printer.Print "=====
```

```
Wu = DeadCo * DeadLoad(CProj) + LiveCo * LiveLoad(CProj)
```

```
****
```

```
Trans_UniformLoad (Wu)'Trans Load
```

```
****
```

```
TPMX = (Printer.Width - (InitX * 2)) \ (XMax - XMin)'1m = TPMX Twips f
or X axis
```

```
TPMY = (Printer.Height - (InitY * 2)) \ (YMax - YMin)'1m = TPMY Twips
for Y axis
```

```
TPP = 2000 / .3 'profile 30 cm use 2000 Twips          '1m =TPP Twips Fo
r Profile Value ,Maximum = Maximum Thickness/2
```

```
TPM = .01
```

```
If TPMX < TPMY Then
```

```
    TPMY = TPMX
```

```
ElseIf TPMX >= TPMY Then
```

```
    TPMX = TPMY
```

```
End If
```

```
**** Draw Column
```

```
For I = 1 To SNodeNum(CProj)
```

```
    X1 = ((Column(CProj, I).Co_X - MinX) * TPMX) - Col + InitX
```

```
    Y1 = ((Column(CProj, I).Co_Y - MinY) * TPMY) - Col + InitY
```

```
    X2 = ((Column(CProj, I).Co_X - MinX) * TPMX) + Col + InitX
```

```
    Y2 = ((Column(CProj, I).Co_Y - MinY) * TPMY) + Col + InitY
```

```
    Printer.Line (X1, Y1)-(X2, Y2), Colors(ColumnColor), BF
```

```
Next I
```

```
**** Draw Element Conjunction in X axis
```

```
For I = 1 To SRow(CProj)
```

```
    For J = 1 To RowEle(I)
```

```
        CStart = EleColRow(CProj, 0, I, J).Start
```

```
        CEnd = EleColRow(CProj, 0, I, J).End
```

```
        X1 = ((Column(CProj, CStart).Co_X - MinX) * TPMX) + InitX
```

```
        Y1 = ((Column(CProj, CStart).Co_Y - MinY) * TPMY) + InitY
```

```
        X2 = ((Column(CProj, CEnd).Co_X - MinX) * TPMX) + InitX
```

```
        Y2 = ((Column(CProj, CEnd).Co_Y - MinY) * TPMY) + InitY
```

```
        Printer.Line (X1, Y1)-(X2, Y2), Colors(ElementColor)
```

```
    Next J
```

```
Next I
```

```
**** Draw Element Conjunction in Y axis
```

```
For I = 1 To SCol(CProj)
```

OUTPUT.BAS - 23

```

For J = 1 To Colele(I)
  CStart = EleColRow(CProj, 1, I, J).Start
  CEnd = EleColRow(CProj, 1, I, J).End
  X1 = ((Column(CProj, CStart).Co_X - MinX) * TPMX) + InitX
  Y1 = ((Column(CProj, CStart).Co_Y - MinY) * TPMY) + InitY
  X2 = ((Column(CProj, CEnd).Co_X - MinX) * TPMX) + InitX
  Y2 = ((Column(CProj, CEnd).Co_Y - MinY) * TPMY) + InitY
  Printer.Line (X1, Y1)-(X2, Y2), Colors(ElementColor)
Next J
Next I

'''' Draw Moment in X axis
Init1 = InitX
Init2 = InitY
For I = 1 To SRow(CProj)
  Trans_BElement False, I
  Cal_FEndMoment False, I
  For J = 1 To RowEle(I)
    InitX = ((Column(CProj, EleColRow(CProj, 0, I, J).Start).Co_X - MinX
) * TPMX) + Init1
    InitY = ((Column(CProj, EleColRow(CProj, 0, I, J).Start).Co_Y - MinY
) * TPMY) + Init2
    Printer.PSet (InitX, InitY), ProfileColor
    If J = 1 Then
      TX1 = InitX
      TY1 = InitY + 150
      Printer.PSet (TX1, TY1)
      Printer.Print Format$(Cal_BShear(J, 0), "0.00")
    End If
    Printer.PSet (InitX, InitY), ProfileColor
    For X = 0 To EleColRow(CProj, 0, I, J).Length Step AddSect
      M = Cal_BMoment(J, X, ML(CProj, J), MR(CProj, J))
      X1 = InitX + (X * TPMX)
      Y1 = InitY
      X2 = InitX + (X * TPMX)
      Y2 = InitY - (M * TPM)
      Printer.Line -(X2, Y2), Colors(MomentColor)
      Printer.Line (X1, Y1)-(X2, Y2), Colors(MomentColor)
    Next X
    TX1 = InitX + EleColRow(CProj, 0, I, J).Length * TPMX
    TY1 = InitY + 150
    Printer.PSet (TX1, TY1)
    Printer.Print Format$(MR(CProj, J) * (-1), "0.00")
  Next J
Next I

'''' Draw Moment in Y axis
For I = 1 To SCol(CProj)
  Trans_BElement True, I
  Cal_FEndMoment True, I
  For J = 1 To Colele(I)
    InitX = (Column(CProj, EleColRow(CProj, 1, I, J).Start).Co_X - MinX
* TPMX) + Init1
    InitY = (Column(CProj, EleColRow(CProj, 1, I, J).Start).Co_Y - MinY
* TPMY) + Init2
    Printer.PSet (InitX, InitY), ProfileColor
    If J = 1 Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 24

```

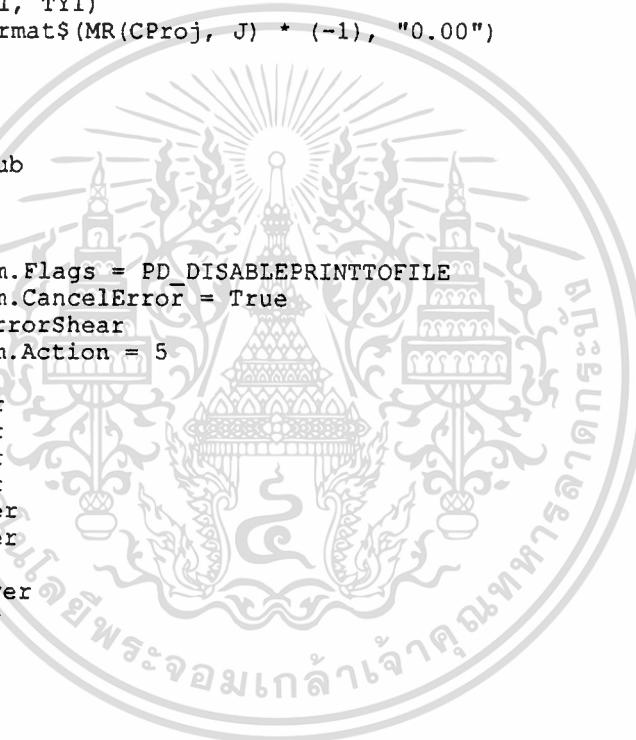
    TX1 = InitX
    TY1 = InitY
    Printer.PSet (TX1, TY1)
    Printer.Print Format$(Cal_BShear(J, 0), "0.00")
End If
Printer.PSet (InitX, InitY), ProfileColor
For Y = 0 To EleColRow(CProj, 1, I, J).Length Step AddSect
    M = Cal_BMoment(J, Y, ML(CProj, J), MR(CProj, J))
    X1 = InitX
    Y1 = InitY + (Y * TPMY)
    X2 = InitX - (M * TPM)
    Y2 = InitY + (Y * TPMY)
    Printer.Line -(X2, Y2), Colors(MomentColor)
    Printer.Line (X1, Y1)-(X2, Y2), Colors(MomentColor)
Next Y
TX1 = InitX
TY1 = InitY + EleColRow(CProj, 1, I, J).Length * TPMY
Printer.PSet (TX1, TY1)
Printer.Print Format$(MR(CProj, J) * (-1), "0.00")
Next J '820438
Next I
Printer.EndDoc
Exit Sub
DlgErrorMoment: Exit Sub
End Sub

Sub Print_Shear ()
    frmMain.cmdlgSystem.Flags = PD_DISABLEPRINTTOFILE
    frmMain.cmdlgSystem.CancelError = True
    On Error GoTo DlgErrorShear
    frmMain.cmdlgSystem.Action = 5

    Dim XMax As Integer
    Dim XMin As Integer
    Dim YMax As Integer
    Dim YMin As Integer
    Dim InitX As Integer
    Dim InitY As Integer
    Dim Col As Integer
    Dim CStart As Integer
    Dim CEnd As Integer
    Dim TPMX As Single
    Dim TPMY As Single
    Dim TPP As Single
    Dim TPM As Single
    Dim AddSect As Single
    Dim X1 As Single
    Dim Y1 As Single
    Dim X2 As Single
    Dim Y2 As Single
    Dim X As Single
    Dim Y As Single
    Dim I As Integer
    Dim J As Integer

    Dim Init1 As Integer
    Dim Init2 As Integer

```



OUTPUT.BAS - 25

```

Dim PF As Single
Dim v As Single
Dim Wu As Single
Dim ColumnColor As Integer
Dim ElementColor As Integer
Dim ProfileColor As Integer

```

```

****

```

```

XMax = Column(CProj, 1).Co_X
XMin = Column(CProj, 1).Co_X
YMax = Column(CProj, 1).Co_Y
YMin = Column(CProj, 1).Co_Y
For I = 1 To SNodeNum(CProj)
  If XMax < Column(CProj, I).Co_X Then
    XMax = Column(CProj, I).Co_X
  End If
  If XMin > Column(CProj, I).Co_X Then
    XMin = Column(CProj, I).Co_X
  End If
  If YMax < Column(CProj, I).Co_Y Then
    YMax = Column(CProj, I).Co_Y
  End If
  If YMin > Column(CProj, I).Co_Y Then
    YMin = Column(CProj, I).Co_Y
  End If
Next I

```

```

**** Init Value

```

```

'frmOutGraph(CProj).Cls
Col = 50 'Column Width in Twips
AddSect = .1 'Additional Section
InitX = 1000 'Left Edge in Twips
InitY = 750 'Top Edge in Twips
ColumnColor = 34
ElementColor = 30
ProfileColor = 27
MomentColor = 17
Printer.PSet (3000, 250)
Printer.Print " SHEAR DIAGRAM "
Printer.PSet (3000, 400)
Printer.Print "======"

```

```

Wu = DeadCo * DeadLoad(CProj) + LiveCo * LiveLoad(CProj)

```

```

****

```

```

Trans_UniformLoad (Wu)'Trans Load

```

```

****

```

```

TPMX = (Printer.Width - (InitX * 2)) \ (XMax - XMin)'1m = TPMX Twips f
or X axis
TPMY = (Printer.Height - (InitY * 2)) \ (YMax - YMin)'1m = TPMY Twips
for Y axis
TPP = 2000 / .3 'profile 30 cm use 2000 Twips '1m =TPP Twips Fo
r Profile Value ,Maximum = Maximum Thickness/2
TPM = .01

```

OUTPUT.BAS - 26

```

If TPMX < TPMY Then
  TPMY = TPMX
ElseIf TPMX >= TPMY Then
  TPMX = TPMY
End If

```

```

''' Draw Column

```

```

For I = 1 To SNodeNum(CProj)
  X1 = ((Column(CProj, I).Co_X - MinX) * TPMX) - Col + InitX
  Y1 = ((Column(CProj, I).Co_Y - MinY) * TPMY) - Col + InitY
  X2 = ((Column(CProj, I).Co_X - MinX) * TPMX) + Col + InitX
  Y2 = ((Column(CProj, I).Co_Y - MinY) * TPMY) + Col + InitY
  Printer.Line (X1, Y1)-(X2, Y2), Colors(ColumnColor), BF
Next I

```

```

''' Draw Element Conjunction in X axis

```

```

For I = 1 To SRow(CProj)
  For J = 1 To RowEle(I)
    CStart = EleColRow(CProj, 0, I, J).Start
    CEnd = EleColRow(CProj, 0, I, J).End
    X1 = ((Column(CProj, CStart).Co_X - MinX) * TPMX) + InitX
    Y1 = ((Column(CProj, CStart).Co_Y - MinY) * TPMY) + InitY
    X2 = ((Column(CProj, CEnd).Co_X - MinX) * TPMX) + InitX
    Y2 = ((Column(CProj, CEnd).Co_Y - MinY) * TPMY) + InitY
    Printer.Line (X1, Y1)-(X2, Y2), Colors(ElementColor)
  Next J
Next I

```

```

''' Draw Element Conjunction in Y axis

```

```

For I = 1 To SCol(CProj)
  For J = 1 To ColEle(I)
    CStart = EleColRow(CProj, 1, I, J).Start
    CEnd = EleColRow(CProj, 1, I, J).End
    X1 = ((Column(CProj, CStart).Co_X - MinX) * TPMX) + InitX
    Y1 = ((Column(CProj, CStart).Co_Y - MinY) * TPMY) + InitY
    X2 = ((Column(CProj, CEnd).Co_X - MinX) * TPMX) + InitX
    Y2 = ((Column(CProj, CEnd).Co_Y - MinY) * TPMY) + InitY
    Printer.Line (X1, Y1)-(X2, Y2), Colors(ElementColor)
  Next J
Next I

```

```

''' Draw Shear in X axis

```

```

Init1 = InitX
Init2 = InitY
For I = 1 To SRow(CProj)
  Trans_BElement False, I
  Cal_FEndMoment False, I
  For J = 1 To RowEle(I)
    InitX = ((Column(CProj, EleColRow(CProj, 0, I, J).Start).Co_X - MinX
* TPMX) + Init1
    InitY = ((Column(CProj, EleColRow(CProj, 0, I, J).Start).Co_Y - MinY
* TPMY) + Init2
    Printer.PSet (InitX, InitY), ProfileColor
    If J = 1 Then
      TX1 = InitX
      TY1 = InitY
    Printer.PSet (TX1, TY1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 27

```

Printer.Print Format$(Cal_BShear(J, 0), "0.00")
End If
Printer.PSet (InitX, InitY), ProfileColor
For X = 0 To EleColRow(CProj, 0, I, J).Length Step AddSect
v = Cal_BShear(J, X)
X1 = InitX + (X * TPMX)
Y1 = InitY
X2 = InitX + (X * TPMX)
Y2 = InitY - (v * TPM)
Printer.Line -(X2, Y2), Colors(MomentColor)
Printer.Line (X1, Y1)-(X2, Y2), Colors(MomentColor)
Next X
TX1 = InitX + EleColRow(CProj, 0, I, J).Length * TPMX
TY1 = InitY + 150
Printer.PSet (TX1, TY1)
Printer.Print Format$(Cal_BShear(J, EleColRow(CProj, 0, I, J).Length
), "0.00")
Next J
Next I

''' Draw Shear in Y axis
For I = 1 To SCol(CProj)
Trans_BElement True, I
Cal_FEndMoment True, I
For J = 1 To ColEle(I)
InitX = ((Column(CProj, EleColRow(CProj, 1, I, J).Start).Co_X - MinX
) * TPMX) + Init1
InitY = ((Column(CProj, EleColRow(CProj, 1, I, J).Start).Co_Y - MinY
) * TPMY) + Init2
Printer.PSet (InitX, InitY), ProfileColor
If J = 1 Then
TX1 = InitX
TY1 = InitY
Printer.PSet (TX1, TY1)
Printer.Print Format$(Cal_BShear(J, 0), "0.00")
End If

Printer.PSet (InitX, InitY), ProfileColor
For Y = 0 To EleColRow(CProj, 1, I, J).Length Step AddSect
v = Cal_BShear(J, Y)
PF = Cal_Profile(True, I, J, Y, 1, 1, emax(CProj))
X1 = InitX
Y1 = InitY + (Y * TPMY)
X2 = InitX - (v * TPM)
Y2 = InitY + (Y * TPMY)
Printer.Line -(X2, Y2), Colors(MomentColor)
Printer.Line (X1, Y1)-(X2, Y2), Colors(MomentColor)
Next Y
TX1 = InitX
TY1 = InitY + EleColRow(CProj, 1, I, J).Length * TPMY
Printer.PSet (TX1, TY1)

Printer.Print Format$(Cal_BShear(J, EleColRow(CProj, 1, I, J).Length
), "0.00")
Next J '820438
Next I
Printer.EndDoc

```

OUTPUT.BAS - 28

```

Exit Sub
DlgErrorShear: Exit Sub
End Sub

```

```

Sub Print_Text (Title As Integer, Text As String)
Dim CH As String * 1
Dim CR As String * 1
Dim LF As String * 1
Dim TB As String * 1
Dim TitleText As String
CR = Chr$(13)
LF = Chr$(10)
TB = Chr$(9)
Printer.FontName = "Courier"
Printer.FontBold = True
Printer.FontSize = 7
Text = " " + Text 'tab from front 2 space
TitleText = ""
Select Case Title
Case 0: 'Unnecessary title
TitleText = ""
Case 1: 'for Column
TitleText = TitleText + CR + LF
TitleText = TitleText + "COLUMN NO. CO_X(m) CO_Y(m) LENGTH(m)
" + CR + LF
TitleText = TitleText + CR + LF
Case 2: 'for row element
TitleText = TitleText + CR + LF
TitleText = TitleText + " ROW ELEMENT START-COL END-COL LENGTH(
TOP-BAY(m) BOTTOM-BAY(m)" + CR + LF
TitleText = TitleText + CR + LF
Case 3: 'for column element
TitleText = TitleText + CR + LF
TitleText = TitleText + "COLUMN ELEMENT START-COL END-COL LENGTH(
TOP-BAY(m) BOTTOM-BAY(m)" + CR + LF
TitleText = TitleText + CR + LF
Case 4: 'for shear check
TitleText = TitleText + CR + LF
TitleText = TitleText + "COLUMN" + s10 + s10 + s5 + "PUNCHING SHEAR"
+ s10 + s10 + s9 + "BEAM-ACTION-SHEAR" + CR + LF
TitleText = TitleText + s2 + "NO." + s9 + "COLUMN" + s3 + s10 + "DROP
-PANEL" + s1 + s10 + "COLUMN-CAPITAL" + CR + LF
TitleText = TitleText + s8 + "Vc(kg)" + s5 + "Vn(kg)" + s5 + "Vc(kg)"
+ s5 + "Vn(kg)" + s5 + "Vc(kg)" + s5 + "Vn(kg)" + s5 + "Vc(kg)" + s5 + "V
n(kg)" + CR + LF
TitleText = TitleText + CR + LF
Case 5: 'for drop panel & capital dimension
TitleText = TitleText + CR + LF
TitleText = TitleText + s2 + "COLUMN" + s2 + "DROP WIDTH(m)" + s2 + "
DROP LENGTH(m)" + s2 + "DROP THICK(m)" + s2 + "CAPITAL WIDTH(m)" + s2 + "C
APITAL LENGTH(m)" + CR + LF
TitleText = TitleText + CR + LF
Case 6: 'Check flexural stress in row
TitleText = TitleText + CR + LF
TitleText = TitleText + s2 + "ROW" + s2 + "ELEMENT" + s2 + "DISTANCE
FROM LEFT(m)" + s2 + "POSITION OF TENDON(cm)" + s2 + "CHECKING STRESS(kg/c
m^2)" + CR + LF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 29

```

TitleText = TitleText + "
                TOP" + s10 + "BOTTOM" + CR + LF
TitleText = TitleText + CR + LF
Case 7: 'Check flexeral stress in column
TitleText = TitleText + CR + LF
TitleText = TitleText + "COLUMN" + s1 + "ELEMENT" + s2 + "DISTANCE FR
OM TOP(m) " + s2 + "POSITION OF TENDON(cm)" + s2 + "CHECKING STRESS(kg/cm^
2)" + CR + LF
TitleText = TitleText + "
                TOP" + s10 + "BOTTOM" + CR + LF
TitleText = TitleText + CR + LF
Case 8: 'Reinforcement for flat plate slab in row
TitleText = TitleText + CR + LF
TitleText = TitleText + "ROW" + s3 + "ELEMENT" + s10 + s1 + "SECTION"
+ s10 + s3 + "DESIGN M(kg-m)" + s3 + "As(cm^2)" + CR + LF
TitleText = TitleText + CR + LF
Case 9: 'Reinforcement for flat plate slab in column
TitleText = TitleText + CR + LF
TitleText = TitleText + "COLUMN" + s1 + "ELEMENT" + s10 + s1 + "SECTI
ON" + s10 + s3 + "DESIGN M(kg-m)" + s3 + "As(cm^2)" + CR + LF
TitleText = TitleText + CR + LF
End Select

'Printer.CurrentY = 500
If Printer.CurrentY > (Printer.Height - 1500) Then
Printer.NewPage
Printer.CurrentY = 1000
Text = TitleText + Text
End If
Printer.Print Text;
End Sub

Sub Print_Try ()
frmMain.cmdlgSystem.Flags = PD_DISABLEPRINTTOFILE
frmMain.cmdlgSystem.CancelError = True
On Error GoTo DlgError
frmMain.cmdlgSystem.Action = 5

Dim XMax As Integer
Dim XMin As Integer
Dim YMax As Integer
Dim YMin As Integer
Dim InitX As Integer
Dim InitY As Integer
Dim Col As Integer
Dim CStart As Integer
Dim CEnd As Integer
Dim TPMX As Single
Dim TPMY As Single
Dim TPP As Single
Dim TPM As Single
Dim AddSect As Single
Dim X1 As Single
Dim Y1 As Single
Dim X2 As Single
Dim Y2 As Single
Dim X As Single

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 30

```

Dim Y As Single
Dim I As Integer
Dim J As Integer
'''end moment
Dim MWL As Single
Dim MWR As Single
Dim MSL As Single
Dim MSR As Single
Dim MUL As Single
Dim MUR As Single

Dim Init1 As Integer
Dim Init2 As Integer
Dim PF As Single
Dim M As Single
Dim Wu As Single
Dim ColumnColor As Integer
Dim ElementColor As Integer
Dim ProfileColor As Integer
'Dim Text As String
Dim CR As String * 1
Dim LF As String * 1
Dim TB As String * 1
CR = Chr$(13)
LF = Chr$(10)
TB = Chr$(9)
Const ucm = " cm"
Const um = " m"
Const ukg = " kg"
Const uksc = " ksc"
Const uksm = " kg/m^2"
Dim CNo As Integer
Dim RNum As Integer

'shear
'Column
Dim Vc0 As Single
Dim Vn0 As Single
'Drop
Dim Vc1 As Single
Dim Vn1 As Single
'Capital
Dim Vc2 As Single
Dim Vn2 As Single
'Diagnal
Dim Vc3 As Single
Dim Vn3 As Single

'''M design
Dim MDL As Single
Dim MDD As Single
Dim MDR As Single
'M design at column strip
Dim MCL As Single
Dim MCD As Single
Dim MCR As Single
'M design at middle strip

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

'OUTPUT.BAS - 31

```

Dim MML1 As Single
Dim MMD1 As Single
Dim MMR1 As Single
Dim MML2 As Single
Dim MMD2 As Single
Dim MMR2 As Single
'''Area of reinforce steel
'Half middle strip 1
Dim MLB1 As Single
Dim MMB1 As Single
Dim MRB1 As Single
Dim MLT1 As Single
Dim MMT1 As Single
Dim MRT1 As Single
'Half middle strip 2
Dim MLB2 As Single
Dim MMB2 As Single
Dim MRB2 As Single
Dim MLT2 As Single
Dim MMT2 As Single
Dim MRT2 As Single
'column strip
Dim CLB As Single
Dim CMB As Single
Dim CRB As Single
Dim CLT As Single
Dim CMT As Single
Dim CRT As Single

'''Reinforcement design
Dim RMDL As Single
Dim RMDD As Single
Dim RMDR As Single
'M design at column strip
Dim RMCL As Single
Dim RMCD As Single
Dim RMCR As Single
'M design at middle strip
Dim RMML1 As Single
Dim RMMD1 As Single
Dim RMMR1 As Single
Dim RMML2 As Single
Dim RMMD2 As Single
Dim RMMR2 As Single
'''Area of reinforce steel
'Half middle strip 1
Dim RMLB1 As Single
Dim RMMB1 As Single
Dim RMRB1 As Single
Dim RMLT1 As Single
Dim RMMT1 As Single
Dim RMRT1 As Single
'Half middle strip 2
Dim RMLB2 As Single
Dim RMMB2 As Single
Dim RMRB2 As Single
Dim RMLT2 As Single

```

OUTPUT.BAS - 32

```
Dim RMMT2 As Single
Dim RMRT2 As Single
'column strip
Dim RCLB As Single
Dim RCMB As Single
Dim RCRB As Single
Dim RCLT As Single
Dim RCMT As Single
Dim RCRT As Single
```

```
AddSect = .5
```

```
''' Init
```

```
frmList(CProj).Cls
```

```
Open DirApp + "temp.swp" For Random Access Read Write Lock Read Write
```

```
As #20 Len = 4
```

```
''' Information
```

```
Print_Text 0, "List of Flat Slab & Flat Plate Design" + CR + LF
```

```
Print_Text 0, "Data " + TB + " : " + SData(CProj).DateData + "
```

```
::: Time : " + SData(CProj).TimeData + CR + LF
```

```
Print_Text 0, "File Name " + TB + " : " + SData(CProj).FileName + CR
```

```
+ LF
Print_Text 0, "Project " + TB + " : " + SData(CProj).ProjectData +
```

```
CR + LF
Print_Text 0, "Engineer " + TB + " : " + SData(CProj).Engineer + CR
```

```
+ LF
Print_Text 0, "Address " + TB + " : " + SData(CProj).AddressData +
```

```
CR + LF
Print_Text 0, "Information" + TB + " : " + SData(CProj).InformData + C
```

```
R + LF
```

```
''' Data for Design
```

```
Print_Text 0, "Length Unit " + TB + " : Meter (m) ::: Load Unit :
```

```
Kilogram (Kg)" + CR + LF
Print_Text 0, "fc' (Slab) " + TB + " : " + Format$(GeneralData(CPro
```

```
j).SFc, ".00") + uksc + CR + LF
Print_Text 0, "fc' (Column) " + TB + " : " + Format$(GeneralData(CPro
```

```
j).CFc, ".00") + uksc + CR + LF
Print_Text 0, "fy (Strand) " + TB + " : " + Format$(GeneralData(CPro
```

```
j).SFs, ".00") + uksc + CR + LF
Select Case StrandIndex(CProj):
```

```
Case 0:
```

```
Case 1:
```

```
Case 2:
```

```
Case 3:
```

```
Case 4:
```

```
Case 5:
```

```
Case 6:
```

```
Case 7:
```

```
End Select
```

```
'Sum_Text Text(TextNo),TextNo, "Es (Strand) : " + Format$(Es, ".00"
```

```
)+ CR + LF
Print_Text 0, "Ec (Slab) : " + Format$(GeneralData(CProj).SEc, "
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 34

```

Print_Text 0, " (m) " + CR + LF
(m) " + CR + LF
For I = 1 To SRow(CProj)
  For J = 1 To RowEle(I)
    LCol = EleColRow(CProj, 0, I, J).Start
    RCol = EleColRow(CProj, 0, I, J).End
    Length = EleColRow(CProj, 0, I, J).Length
    LBay = EleColRow(CProj, 0, I, J).L1
    RBay = EleColRow(CProj, 0, I, J).L2
    Print_Text 1, Format$(I, "@@@@@@") + Format$(J, "@@@@@@@@@@@@@@@@")
    + Format$(LCol, "@@@@@@@@@@@@@@@@") + Format$(RCol, "@@@@@@@@@@@@@@@@")
    + Format$(Length, "@@@@@@@@@@@@.00") + Format$(LBay, "@@@@@@@@@@@@.00")
    + Format$(RBay, "@@@@@@@@@@@@.00") + CR + LF
  Next J
Next I
'Column
Print_Text 0, CR + LF
Print_Text 0, CR + LF
Print_Text 0, "Row Element Data" + CR + LF
Print_Text 0, CR + LF
Print_Text 0, "Column:Element No.:Start Column: End Column: Length
: Left Bay : Right Bay " + CR + LF
Print_Text 0, CR + LF
For I = 1 To Scol(CProj)
  For J = 1 To Colele(I)
    LCol = EleColRow(CProj, 1, I, J).Start
    RCol = EleColRow(CProj, 1, I, J).End
    Length = EleColRow(CProj, 1, I, J).Length
    LBay = EleColRow(CProj, 1, I, J).L1
    RBay = EleColRow(CProj, 1, I, J).L2
    Print_Text 2, Format$(I, "@@@@@@") + Format$(J, "@@@@@@@@@@@@@@@@")
    + Format$(LCol, "@@@@@@@@@@@@@@@@") + Format$(RCol, "@@@@@@@@@@@@@@@@")
    + Format$(Length, "@@@@@@@@@@@@.00") + Format$(LBay, "@@@@@@@@@@@@.00")
    + Format$(RBay, "@@@@@@@@@@@@.00") + CR + LF
  Next J
Next I
Print_Text 0, CR + LF
Print_Text 0, CR + LF
Print_Text 0, CR + LF
Dim SSystem As String * 50
Select Case GeneralData(CProj).SlabIndex
Case 0:
  SSystem = "Flat Plate"
Case 1:
  SSystem = "Flat Slab With Drop Panel"
Case 2:
  SSystem = "Flat Slab With Column Capital"
Case 3:
  SSystem = "Flat Slab With Column Drop Panel And Column Capital"
End Select
Print_Text 0, "SLAB SYSTEM : " + SSystem + CR + LF
Print_Text 0, CR + LF
Print_Text 0, "Load factor ::: " + CR + LF
Print_Text 0, "Wu = " + Format$(DeadCo, "0.0") + "DL" + Format$(LiveCo,
"0.0") + "LL" + CR + LF
Print_Text 0, "Unfactored Dead Load = " + Format$(DeadLoad(C
Proj), "0.00") + uksm + CR + LF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 35

```

Print_Text 0, "Unfactored Live Load          = " + Format$(LiveLoad(C
Proj), "0.00") + uksm + CR + LF
Print_Text 0, "Unfactored Supper imposed load = " + Format$(SuperLoad(
CProj), "0.00") + uksm + CR + LF
Print_Text 0, ""

Print_Text 0, "Column " + TB + "          Punching at Column  " + TB + TB
+ "          Punching at Drop Panel  " + TB + "          Punching at Capit
al          " + TB + "          Diagonal shear          " + CR + LF
Print_Text 0, " No.  " + Chr$(9) + " Vc(kg) " + Chr$(9) + Chr$(9) +
" Vn(kg) " + Chr$(9) + Chr$(9) + " Vc(kg) " + Chr$(9) + Chr$(9) + " Vn(
kg) " + Chr$(9) + Chr$(9) + " Vc(kg) " + Chr$(9) + Chr$(9) + " Vn(kg) "
+ Chr$(9) + Chr$(9) + " Vc(kg) " + Chr$(9) + Chr$(9) + " Vn(kg) " + Ch
r$(9) + CR + LF
Print_Text 0, CR + LF
RNum = 2
RIden = 2
Dim ColumnNo As Integer
ColumnNo = 0
For I = 1 To SRow(CProj)
  For J = 1 To RowEle(I)
    ColumnNo = ColumnNo + 1
    CNo = EleColRow(CProj, 0, I, J).Start
    If Column(CProj, CNo).SupportType <> 2 Then 'If Not Free,Mean That i
s Column support
      RIden = RNum + ((CNo - 1) * 8)
      Get #20, RIden + 1, Vc0
      Get #20, RIden + 2, Vn0
      Get #20, RIden + 3, Vc1
      Get #20, RIden + 4, Vn1
      Get #20, RIden + 5, Vc2
      Get #20, RIden + 6, Vn2
      Get #20, RIden + 7, Vc3
      Get #20, RIden + 8, Vn3
      Print_Text 3, Format$(ColumnNo, "###") + Chr$(9) + Format$(Vc0, "0
.0000E+00") + Chr$(9) + Format$(Vn0, "0.0000E+00") + Chr$(9) + Format$(Vc1
, "0.0000E+00") + Chr$(9) + Format$(Vn1, "0.0000E+00") + Chr$(9) + Format$(
Vc2, "0.0000E+00") + Chr$(9) + Format$(Vn2, "0.0000E+00") + Chr$(9) + For
mat$(Vc3, "0.0000E+00") + Chr$(9) + Format$(Vn3, "0.0000E+00") + CR + LF
    End If
  If J = RowEle(I) Then
    ColumnNo = ColumnNo + 1
    CNo = EleColRow(CProj, 0, I, J).End
    If Column(CProj, CNo).SupportType <> 2 Then
      RIden = RNum + ((CNo - 1) * 8)
      Get #20, RIden + 1, Vc0
      Get #20, RIden + 2, Vn0
      Get #20, RIden + 3, Vc1
      Get #20, RIden + 4, Vn1
      Get #20, RIden + 5, Vc2
      Get #20, RIden + 6, Vn2
      Get #20, RIden + 7, Vc3
      Get #20, RIden + 8, Vn3
      Print_Text 3, Format$(ColumnNo, "###") + Chr$(9) + Format$(Vc0, "
0.0000E+00") + Chr$(9) + Format$(Vn0, "0.0000E+00") + Chr$(9) + Format$(Vc
1, "0.0000E+00") + Chr$(9) + Format$(Vn1, "0.0000E+00") + Chr$(9) + Format
$(Vc2, "0.0000E+00") + Chr$(9) + Format$(Vn2, "0.0000E+00") + Chr$(9) + Fo

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 36

```

rmat$(Vc3, "0.0000E+00") + Chr$(9) + Format$(Vn3, "0.0000E+00") + CR + LF

    End If
    End If
    Next J
    Next I

'Drop panel & Capital
CNo = 0
ColumnNo = 0
Print_Text 0, CR + LF
Print_Text 0, "Dimension of drop & column capital (m)"
Print_Text 0, CR + LF
Print_Text 0, " No. " + TB + "Drop Width" + TB + "Drop Length" + TB +
"Drop Thick" + TB + "Capital Width" + TB + "Capital Length" + TB + CR + LF
Print_Text 0, CR + LF
For I = 1 To SRow(CProj)
    For J = 1 To RowEle(I)
        CNo = CNo + 1
        ColumnNo = EleColRow(CProj, 0, I, J).Start
        DropX = Column(CProj, ColumnNo).DropX
        DropY = Column(CProj, ColumnNo).DropY
        DropThick = Column(CProj, ColumnNo).DropThick
        CapitalX = Column(CProj, ColumnNo).CapitalX
        CapitalY = Column(CProj, ColumnNo).CapitalY
        Print_Text 4, Format$(CNo, "@@@@") + Chr$(9) + Format$(DropX, "0.000
0E+00") + Chr$(9) + Format$(DropY, "0.0000E+00") + Chr$(9) + Format$(DropT
hick, "0.0000E+00") + Chr$(9) + Format$(CapitalX, "0.0000E+00") + Chr$(9)
+ Format$(CapitalY, "0.0000E+00") + CR + LF
        If J = RowEle(I) Then
            CNo = CNo + 1
            ColumnNo = EleColRow(CProj, 0, I, RowEle(I)).End
            DropX = Column(CProj, ColumnNo).DropX
            DropY = Column(CProj, ColumnNo).DropY
            DropThick = Column(CProj, ColumnNo).DropThick
            CapitalX = Column(CProj, ColumnNo).CapitalX
            CapitalY = Column(CProj, ColumnNo).CapitalY
            Print_Text 4, Format$(CNo, "@@@@") + Chr$(9) + Format$(DropX, "0.00
00E+00") + Chr$(9) + Format$(DropY, "0.0000E+00") + Chr$(9) + Format$(Drop
Thick, "0.0000E+00") + Chr$(9) + Format$(CapitalX, "0.0000E+00") + Chr$(9)
+ Format$(CapitalY, "0.0000E+00") + CR + LF
        End If
    Next J
Next I
'declare
Dim Lmax As Single
Dim Bbeam As Single
Dim P As Single
Dim Top As Single
Dim Bottom As Single
Dim Text1 As String
Dim Text2 As String
Dim Text3 As String
Dim Text4 As String

Text1 = ""

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 37

```

'''check Cracking Moment & Deflection
Wu = DeadLoad(CProj) + LiveLoad(CProj) + SuperLoad(CProj)
'text flexeral stress in row
Lmax = 0
Bbeam = 0
Mu = 0
For I = 1 To SRow(CProj)
  Trans_UniformLoad Wu
  Trans_BElement False, I
  Cal_UltimateEndMoment False, I
  Mu = Find_MuMax(False, I)
  Find_BL True, I, Lmax, Bbeam
  P = .775 * GeneralData(CProj).SFs_ * Find_AsStrand(Lmax, Bbeam)
  Print_Text 0, CR + LF
  Print_Text 0, " ROW : " + Format$(I, "@@@") + CR + LF
  Print_Text 0, " No of tendon = " + Format$(Find_AsStrand(Lmax, Bbeam)
, "@@@") + TB + "Unit" + CR + LF
  Print_Text 0, " Tension per 1 unit tendon = " + Format$(GeneralData(C
Proj).SFs_, "0.00") + " Kg" + CR + LF
  Print_Text 0, " Mcr = " + Format$(Check_CrackMoment(P, Bbeam), "0.00"
) + " kg-m" + CR + LF
  Print_Text 0, " Mu = " + Format$(Mu, ".00") + " Kg-m" + CR + LF
  Print_Text 0, "CHECK DEFLECTION AT AFTER TENSIONED" + CR + LF
  Print_Text 0, " Deflection(Actual) = " + Format$(Find_MaxDeflectio
n(False, I, 1), "0.00E+00") + um + CR + LF
  Print_Text 0, " Deflection(Allowable) = " + Format$(Check_Deflection(
False, I), "0.00E+00") + um + CR + LF
  Print_Text 0, "CHECK DEFLECTION AT USED" + CR + LF
  Print_Text 0, " Deflection(Actual) = " + Format$(Find_MaxDeflectio
n(False, I, 2), "0.00E+00") + um + CR + LF
  Print_Text 0, " Deflection(Allowable) = " + Format$(Check_Deflection(
False, I), "0.00E+00") + um + CR + LF
  Print_Text 0, "=====
===== " + CR + LF
Next I

'Textiflexeral stress in Column
Lmax = 0
Bbeam = 0
Mu = 0
For I = 1 To SRow(CProj)
  Trans_UniformLoad Wu
  Trans_BElement True, I
  Cal_UltimateEndMoment True, I
  Mu = Find_MuMax(True, I)
  Find_BL True, I, Lmax, Bbeam
  P = .775 * GeneralData(CProj).SFs_ * Find_AsStrand(Lmax, Bbeam)
  Print_Text 0, CR + LF
  Print_Text 0, " COLUMN : " + Format$(I, "@@@") + CR + LF
  Print_Text 0, " No of tendon = " + Format$(Find_AsStrand(Lmax, Bbeam)
, "@@@") + TB + "Unit" + CR + LF
  Print_Text 0, " Tension per 1 unit tendon = " + Format$(GeneralData(C
Proj).SFs_, ".00") + " Kg" + CR + LF
  Print_Text 0, " Mcr = " + Format$(Check_CrackMoment(P, Bbeam), ".00")
+ " kg-m" + CR + LF
  Print_Text 0, " Mu = " + Format$(Mu, ".00") + " Kg-m" + CR + LF
  Print_Text 0, "CHECK DEFLECTION AT USED" + CR + LF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 38

```

Print_Text 0, " Deflection(Actual)      = " + Format$(Find_MaxDeflectio
n(True, I, 1), "0.00E+00") + um + CR + LF
Print_Text 0, " Deflection(Allowable) = " + Format$(Check_Deflection(
True, I), "0.00E+00") + um + CR + LF
Print_Text 0, "CHECK DEFLECTION AT AFTER TENSIONED" + CR + LF
Print_Text 0, " Deflection(Actual)      = " + Format$(Find_MaxDeflectio
n(True, I, 2), "0.00E+00") + um + CR + LF
Print_Text 0, " Deflection(Allowable) = " + Format$(Check_Deflection(
True, I), "0.00E+00") + um + CR + LF
Print_Text 0, "=====
===== " + CR + LF
Next I

''''at working load
Text2 = ""
Wu = DeadLoad(CProj)
'text flexeral stress in row
Lmax = 0
Bbeam = 0
Mu = 0
For I = 1 To SRow(CProj)
  Trans_UniformLoad Wu
  Trans_BElement False, I
  Cal_ElasticEndMoment False, I
  Find_BL True, I, Lmax, Bbeam
  P = .775 * GeneralData(CProj).SFs_ * Find_AsStrand(Lmax, Bbeam)
  Print_Text 0, "Row No. " + " Element No." + TB + "Distance From Lef
t" + TB + " Position of tendon" + TB + TB + TB + "Checking Stress" + CR +
LF
  Print_Text 0, " " + TB + TB + " " + TB + TB + TB + TB
+ TB + TB + TB + TB + "Top" + TB + "Bottom" + CR + LF
  Print_Text 0, CR + LF
  For J = 1 To RowEle(I)
    For X = 0 To EleColRow(CProj, 0, I, J).Length Step AddSect
      M = Cal_BMoment(J, X, ML(CProj, J), MR(CProj, J))
      PF = Cal_Profile(False, I, J, X, 1, 1, emax(CProj))
      Find_Stress P, M, PF, EleColRow(CProj, 0, I, J).L1 / 2 + EleColRow(
CProj, 0, I, J).L2 / 2, Top, Bottom
      Print_Text 5, Format$(I, "#####") + TB + TB + Format$(J, "#####") +
TB + TB + Format$(X, "0.00E+00") + TB + TB + Format$(PF * 100, "0.000E+00
") + TB + TB + Format$(Top / 10000, "0.000E+00") + TB + Format$(Bottom / 1
0000, "0.00E+00") + CR + LF
    Next X
    Print_Text 0, "=====
===== " + CR + LF
  Next J
Next I

'text2 flexeral stress in Column
Lmax = 0
Bbeam = 0
Mu = 0
For I = 1 To SRow(CProj)
  Trans_UniformLoad Wu
  Trans_BElement True, I
  Cal_ElasticEndMoment True, I
  Find_BL True, I, Lmax, Bbeam

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 39

```

P = .775 * GeneralData(CProj).SFs_ * Find_AsStrand(Lmax, Bbeam)
Print_Text 0, CR + LF
Print_Text 0, "Column No." + " Element No." + TB + "Distance From Left" + TB + " Position of tendon" + TB + TB + TB + "Checking Stress" + CR + LF
Print_Text 0, " " + TB + TB + " " + TB + TB + TB + TB
+ TB + TB + TB + "Top" + TB + TB + "Bottom" + CR + LF
For J = 1 To RowEle(I)
  For X = 0 To EleColRow(CProj, 0, I, J).Length Step AddSect
    M = Cal_BMoment(J, X, ML(CProj, J), MR(CProj, J))
    PF = Cal_Profile(False, I, J, X, 1, 1, emax(CProj))
    Find_Stress P, M, PF, EleColRow(CProj, 0, I, J).L1 / 2 + EleColRow(CProj, 0, I, J).L2 / 2, Top, Bottom
    Print_Text 6, Format$(I, "#####") + TB + TB + Format$(J, "#####") + TB + TB + Format$(X, "0.00E+00") + TB + TB + Format$(PF * 100, "0.000E+00") + TB + TB + Format$(Top / 10000, "0.000E+00") + TB + Format$(Bottom / 10000, "0.00E+00") + CR + LF
  Next X
  Print_Text 0, "=====" + CR + LF
  Print_Text 0, "=====" + CR + LF
Next J
Next I

AddSect = .5
'''check at transfer load
Text3 = ""
Wu = DeadLoad(CProj) + LiveLoad(CProj) + SuperLoad(CProj)
Text3 flexural stress in row
Lmax = 0
Bbeam = 0
Mu = 0
For I = 1 To SRow(CProj)
  Trans_UniformLoad Wu
  Trans_BElement False, I
  Cal_ServiceEndMoment False, I
  Find_BL True, I, Lmax, Bbeam
  P = .775 * GeneralData(CProj).SFs_ * Find_AsStrand(Lmax, Bbeam)
  Print_Text 0, CR + LF
  Print_Text 0, "Row No. " + " Element No." + TB + "Distance From Left" + TB + " Position of tendon" + TB + TB + TB + "Checking Stress" + CR + LF
  Print_Text 0, " " + TB + TB + " " + TB + TB + TB + TB
+ TB + TB + TB + "Top" + TB + TB + "Bottom" + CR + LF
  Print_Text 0, CR + LF
  For J = 1 To RowEle(I)
    For X = 0 To EleColRow(CProj, 0, I, J).Length Step AddSect
      M = Cal_BMoment(J, X, ML(CProj, J), MR(CProj, J))
      PF = Cal_Profile(False, I, J, X, 1, 1, emax(CProj))
      Find_Stress P, M, PF, EleColRow(CProj, 0, I, J).L1 / 2 + EleColRow(CProj, 0, I, J).L2 / 2, Top, Bottom
      Print_Text 0, Format$(I, "#####") + TB + TB + Format$(J, "#####") + TB + TB + Format$(X, "0.00E+00") + TB + TB + Format$(PF * 100, "0.000E+00") + TB + TB + Format$(Top / 10000, "0.000E+00") + TB + Format$(Bottom / 10000, "0.00E+00") + CR + LF
    Next X
    Print_Text 0, "=====" + CR + LF
  Next J
  Print_Text 0, "=====" + CR + LF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 40

```

Next J
Next I

'Text3 flexeral stress in Column
Lmax = 0
Bbeam = 0
Mu = 0
For I = 1 To SRow(CProj)
  Trans_UniformLoad Wu
  Trans_BEElement True, I
  Cal_ServiceEndMoment False, I
  Find_BL True, I, Lmax, Bbeam
  P = .775 * GeneralData(CProj).SFs_ * Find_AsStrand(Lmax, Bbeam)
  Print_Text 0, CR + LF
  Print_Text 0, "Column No." + " Element No." + TB + "Distance From Le
ft" + TB + " Position of tendon" + TB + TB + TB + "Checking Stress" + CR +
LF
  Print_Text 0, " " + TB + TB + " " + TB + TB + TB + TB
+ TB + TB + TB + "Top" + TB + TB + "Bottom" + CR + LF
  Print_Text 0, CR + LF
  For J = 1 To RowEle(I)
    For X = 0 To EleColRow(CProj, 0, I, J).Length Step AddSect
      M = Cal_BMoment(J, X, ML(CProj, J), MR(CProj, J))
      PF = Cal_Profile(False, I, J, X, 1, 1, emax(CProj))
      Find_Stress P, M, PF, EleColRow(CProj, 0, I, J).L1 / 2 + EleColRow(
CProj, 0, I, J).L2 / 2, Top, Bottom
      Print_Text 0, Format$(I, "#####") + TB + TB + Format$(J, "#####") +
TB + TB + Format$(X, "0.00E+00") + TB + TB + Format$(PF * 100, "0.000E+00
") + TB + TB + Format$(Top / 10000, "0.000E+00") + TB + Format$(Bottom / 1
0000, "0.00E+00") + CR + LF
    Next X
    Print_Text 0, "=====" + CR + LF
  Next J
Next I
Printer.EndDoc
Close #20
Exit Sub
DlgError: Exit Sub
End Sub

Sub Screen_Geometric ()
Dim XMax As Integer
Dim XMin As Integer
Dim YMax As Integer
Dim YMin As Integer
Dim InitX As Integer
Dim InitY As Integer
Dim Col As Integer
Dim CStart As Integer
Dim CEnd As Integer
Dim TPMX As Single
Dim TPMY As Single
Dim TPP As Single
Dim TPM As Single
Dim AddSect As Single
Dim X1 As Single

```

OUTPUT.BAS - 41

```
Dim Y1 As Single
Dim X2 As Single
Dim Y2 As Single
Dim X As Single
Dim Y As Single
Dim I As Integer
Dim J As Integer
```

```
Dim Init1 As Integer
Dim Init2 As Integer
Dim PF As Single
Dim M As Single
Dim Wu As Single
Dim ColumnColor As Integer
Dim ElementColor As Integer
Dim ProfileColor As Integer
```

```
''''
XMax = Column(CProj, 1).Co_X
XMin = Column(CProj, 1).Co_X
YMax = Column(CProj, 1).Co_Y
YMin = Column(CProj, 1).Co_Y
For I = 1 To SNodeNum(CProj)
  If XMax < Column(CProj, I).Co_X Then
    XMax = Column(CProj, I).Co_X
  End If
  If XMin > Column(CProj, I).Co_X Then
    XMin = Column(CProj, I).Co_X
  End If
  If YMax < Column(CProj, I).Co_Y Then
    YMax = Column(CProj, I).Co_Y
  End If
  If YMin > Column(CProj, I).Co_Y Then
    YMin = Column(CProj, I).Co_Y
  End If
Next I
```

```
'''' Init Value
frmOutGraph(CProj).Cls
Col = 50 'Column Width in Twips
AddSect = .1 'Additional Section
InitX = 1000 'Left Edge in Twips
InitY = 750 'Top Edge in Twips
ColumnColor = 34
ElementColor = 30
ProfileColor = 27
MomentColor = 17
frmOutGraph(CProj).PSet (3000, 250)
frmOutGraph(CProj).Print " SLAB TO DESIGN "
frmOutGraph(CProj).PSet (3000, 350)
frmOutGraph(CProj).Print "====="
```

```
Wu = DeadCo * DeadLoad(CProj) + LiveCo * LiveLoad(CProj)
```

```
''''
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 42

Trans_UniformLoad (Wu)'Trans Load

''''

TPMX = (frmOutGraph(CProj).Width - (InitX + 2)) \ (XMax - XMin)'1m = T

PMX Twips for X axis

TPMY = (frmOutGraph(CProj).Height - (InitY + 2)) \ (YMax - YMin)'1m =

TPMY Twips for Y axis

TPP = 2000 / .3 'profile 30 cm use 2000 Twips '1m =TPP Twips Fo

r Profile Value ,Maximum = Maximum Thickness/2

TPM = .01

If TPMX < TPMY Then

TPMY = TPMX

ElseIf TPMX >= TPMY Then

TPMX = TPMY

End If

'''' Draw Column

For I = 1 To SNodeNum(CProj)

X1 = ((Column(CProj, I).Co_X - MinX) * TPMX) - Col + InitX

Y1 = ((Column(CProj, I).Co_Y - MinY) * TPMY) - Col + InitY

X2 = ((Column(CProj, I).Co_X - MinX) * TPMX) + Col + InitX

Y2 = ((Column(CProj, I).Co_Y - MinY) * TPMY) + Col + InitY

frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(ColumnColor), BF

frmOutGraph(CProj).Print Format\$(I, "0") + "[" + Format\$(Column(CProj

, I).Co_X, "#.00") + ", " + Format\$(Column(CProj, I).Co_Y, "#.00") + "]"

Next I

'''' Draw Element Conjunction in X axis

For I = 1 To SRow(CProj)

For J = 1 To RowEle(I)

CStart = EleColRow(CProj, 0, I, J).Start

CEnd = EleColRow(CProj, 0, I, J).End

X1 = ((Column(CProj, CStart).Co_X - MinX) * TPMX) + InitX

Y1 = ((Column(CProj, CStart).Co_Y - MinY) * TPMY) + InitY

X2 = ((Column(CProj, CEnd).Co_X - MinX) * TPMX) + InitX

Y2 = ((Column(CProj, CEnd).Co_Y - MinY) * TPMY) + InitY

frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(ElementColor)

Next J

Next I

'''' Draw Element Conjunction in Y axis

For I = 1 To SCol(CProj)

For J = 1 To ColEle(I)

CStart = EleColRow(CProj, 1, I, J).Start

CEnd = EleColRow(CProj, 1, I, J).End

X1 = ((Column(CProj, CStart).Co_X - MinX) * TPMX) + InitX

Y1 = ((Column(CProj, CStart).Co_Y - MinY) * TPMY) + InitY

X2 = ((Column(CProj, CEnd).Co_X - MinX) * TPMX) + InitX

Y2 = ((Column(CProj, CEnd).Co_Y - MinY) * TPMY) + InitY

frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(ElementColor)

Next J

Next I

End Sub

Sub Screen_Layout ()

Dim XMax As Integer

Dim XMin As Integer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 43

```

Dim YMax As Integer
Dim YMin As Integer
Dim InitX As Integer
Dim InitY As Integer
Dim Col As Integer
Dim CStart As Integer
Dim CEnd As Integer
Dim TPMX As Single
Dim TPMY As Single
Dim TPP As Single
Dim TPM As Single
Dim AddSect As Single
Dim X1 As Single
Dim Y1 As Single
Dim X2 As Single
Dim Y2 As Single
Dim X As Single
Dim Y As Single
Dim I As Integer
Dim J As Integer

```

```

Dim Init1 As Integer
Dim Init2 As Integer
Dim PF As Single
Dim M As Single
Dim Wu As Single
Dim ColumnColor As Integer
Dim ElementColor As Integer
Dim ProfileColor As Integer

```

```

''''

```

```

XMax = Column(CProj, 1).Co_X
XMin = Column(CProj, 1).Co_X
YMax = Column(CProj, 1).Co_Y
YMin = Column(CProj, 1).Co_Y
For I = 1 To SNodeNum(CProj)
  If XMax < Column(CProj, I).Co_X Then
    XMax = Column(CProj, I).Co_X
  End If
  If XMin > Column(CProj, I).Co_X Then
    XMin = Column(CProj, I).Co_X
  End If
  If YMax < Column(CProj, I).Co_Y Then
    YMax = Column(CProj, I).Co_Y
  End If
  If YMin > Column(CProj, I).Co_Y Then
    YMin = Column(CProj, I).Co_Y
  End If
Next I

```

```

'''' Init Value
frmOutGraph(CProj).Cls
Col = 50 'Column Width in Twips
AddSect = .1 'Additional Section
InitX = 1000 'Left Edge in Twips
InitY = 750 'Top Edge in Twips
ColumnColor = 34

```



OUTPUT.BAS - 44

```

ElementColor = 30
ProfileColor = 27
MomentColor = 17
Wu = DeadCo * DeadLoad(CProj) + LiveCo * LiveLoad(CProj)

''''
Trans_UniformLoad (Wu)'Trans Load

''''
TPMX = (frmOutGraph(CProj).Width - (InitX * 2)) \ (XMax - XMin)'1m = T
PMX Twips for X axis
TPMY = (frmOutGraph(CProj).Height - (InitY * 2)) \ (YMax - YMin)'1m =
TPMY Twips for Y axis
TPP = 2000 / .3 'profile 30 cm use 2000 Twips '1m =TPP Twips Fo
r Profile Value ,Maximum = Maximum Thickness/2
TPM = .01
If TPMX < TPMY Then
    TPMY = TPMX
ElseIf TPMX >= TPMY Then
    TPMX = TPMY
End If

'''' Draw Column
For I = 1 To SNodeNum(CProj)
    X1 = ((Column(CProj, I).Co_X - MinX) * TPMX) - Col + InitX
    Y1 = ((Column(CProj, I).Co_Y - MinY) * TPMY) - Col + InitY
    X2 = ((Column(CProj, I).Co_X - MinX) * TPMX) + Col + InitX
    Y2 = ((Column(CProj, I).Co_Y - MinY) * TPMY) + Col + InitY
    frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(ColumnColor), BF
Next I

'''' Draw Element Conjunction in X axis
For I = 1 To SRow(CProj)
    For J = 1 To RowEle(I)
        CStart = EleColRow(CProj, 0, I, J).Start
        CEnd = EleColRow(CProj, 0, I, J).End
        X1 = ((Column(CProj, CStart).Co_X - MinX) * TPMX) + InitX
        Y1 = ((Column(CProj, CStart).Co_Y - MinY) * TPMY) + InitY
        X2 = ((Column(CProj, CEnd).Co_X - MinX) * TPMX) + InitX
        Y2 = ((Column(CProj, CEnd).Co_Y - MinY) * TPMY) + InitY
        frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(ElementColor)
    Next J
Next I

'''' Draw Element Conjunction in Y axis
For I = 1 To SCol(CProj)
    For J = 1 To ColEle(I)
        CStart = EleColRow(CProj, 1, I, J).Start
        CEnd = EleColRow(CProj, 1, I, J).End
        X1 = ((Column(CProj, CStart).Co_X - MinX) * TPMX) + InitX
        Y1 = ((Column(CProj, CStart).Co_Y - MinY) * TPMY) + InitY
        X2 = ((Column(CProj, CEnd).Co_X - MinX) * TPMX) + InitX
        Y2 = ((Column(CProj, CEnd).Co_Y - MinY) * TPMY) + InitY
        frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(ElementColor)
    Next J
Next I

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 45

```

''' Draw Profile in X axis
Init1 = InitX
Init2 = InitY
For I = 1 To SRow(CProj)
  Trans_BElement False, I
  Cal_FEndMoment False, I
  For J = 1 To RowEle(I)
    InitX = ((Column(CProj, EleColRow(CProj, 0, I, J).Start).Co_X - MinX
) * TPMX) + Init1
    InitY = ((Column(CProj, EleColRow(CProj, 0, I, J).Start).Co_Y - MinY
) * TPMY) + Init2
    frmOutGraph(CProj).PSet (InitX, InitY), ProfileColor
    For X = 0 To EleColRow(CProj, 0, I, J).Length Step AddSect
      M = Cal_BMoment(J, X, ML(CProj, J), MR(CProj, J))
      PF = Cal_Profile(False, I, J, X, 1, 1, emax(CProj))
      X1 = InitX + (X * TPMX)
      Y1 = InitY
      X2 = InitX + (X * TPMX)
      Y2 = InitY - (M * TPM)
      frmOutGraph(CProj).Line -(X2, Y2), Colors(MomentColor)
      frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(MomentColor)

      X1 = InitX + (X * TPMX)
      Y1 = InitY
      X2 = InitX + (X * TPMX)
      Y2 = InitY - (PF * TPP)
      'frmOutGraph(CProj).Line -(X2, Y2), Colors(ProfileColor)
      'frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(ProfileColor)
    Next X
  Next J
Next I

''' Draw Profile in Y axis
For I = 1 To SCol(CProj)
  Trans_BElement True, I
  Cal_FEndMoment True, I
  For J = 1 To ColEle(I)
    InitX = (Column(CProj, EleColRow(CProj, 1, I, J).Start).Co_X - MinX
) * TPMX + Init1
    InitY = (Column(CProj, EleColRow(CProj, 1, I, J).Start).Co_Y - MinY
) * TPMY + Init2
    frmOutGraph(CProj).PSet (InitX, InitY), ProfileColor
    For Y = 0 To EleColRow(CProj, 1, I, J).Length Step AddSect
      M = Cal_BMoment(J, Y, ML(CProj, J), MR(CProj, J))
      PF = Cal_Profile(True, I, J, Y, 1, 1, emax(CProj))
      X1 = InitX
      Y1 = InitY + (Y * TPMY)
      X2 = InitX - (M * TPM)
      Y2 = InitY + (Y * TPMY)
      frmOutGraph(CProj).Line -(X2, Y2), Colors(MomentColor)
      frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(MomentColor)

      X1 = InitX
      Y1 = InitY + (Y * TPMY)
      X2 = InitX - (PF * TPP)
      Y2 = InitY + (Y * TPMY)

```

OUTPUT.BAS - 46

```
'frmOutGraph(CProj).Line -(X2, Y2), Colors(ProfileColor)
'frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(ProfileColor)
Next Y
Next J '820438
Next I
End Sub
```

```
Sub Screen Moment ()
Dim XMax As Integer
Dim XMin As Integer
Dim YMax As Integer
Dim YMin As Integer
Dim InitX As Integer
Dim InitY As Integer
Dim Col As Integer
Dim CStart As Integer
Dim CEnd As Integer
Dim TPMX As Single
Dim TPMY As Single
Dim TPP As Single
Dim TPM As Single
Dim AddSect As Single
Dim X1 As Single
Dim Y1 As Single
Dim X2 As Single
Dim Y2 As Single
Dim X As Single
Dim Y As Single
Dim I As Integer
Dim J As Integer

Dim Init1 As Integer
Dim Init2 As Integer
Dim PF As Single
Dim M As Single
Dim Wu As Single
Dim ColumnColor As Integer
Dim ElementColor As Integer
Dim ProfileColor As Integer
```

```
''''
XMax = Column(CProj, 1).Co_X
XMin = Column(CProj, 1).Co_X
YMax = Column(CProj, 1).Co_Y
YMin = Column(CProj, 1).Co_Y
For I = 1 To SNodeNum(CProj)
If XMax < Column(CProj, I).Co_X Then
XMax = Column(CProj, I).Co_X
End If
If XMin > Column(CProj, I).Co_X Then
XMin = Column(CProj, I).Co_X
End If
If YMax < Column(CProj, I).Co_Y Then
YMax = Column(CProj, I).Co_Y
End If
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 47

```

If YMin > Column(CProj, I).Co_Y Then
  YMin = Column(CProj, I).Co_Y
End If
Next I

**** Init Value
frmOutGraph(CProj).Cls
Col = 50 'Column Width in Twips
AddSect = .1 'Additional Section
InitX = 1000 'Left Edge in Twips
InitY = 750 'Top Edge in Twips
ColumnColor = 34
ElementColor = 30
ProfileColor = 27
MomentColor = 17
frmOutGraph(CProj).PSet (3000, 250)
frmOutGraph(CProj).Print " MOMENT DIAGRAM "
frmOutGraph(CProj).PSet (3000, 350)
frmOutGraph(CProj).Print "===== "

Wu = DeadCo * DeadLoad(CProj) + LiveCo * LiveLoad(CProj)

****
Trans_UniformLoad (Wu)'Trans Load

****
TPMX = (frmOutGraph(CProj).Width - (InitX * 2)) \ (XMax - XMin)'lm = T
PMX Twips for X axis
TPMY = (frmOutGraph(CProj).Height - (InitY * 2)) \ (YMax - YMin)'lm =
TPMY Twips for Y axis
TPP = 2000 / .3 'profile 30 cm use 2000 Twips 'lm =TPP Twips Fo
r Profile Value ,Maximum = Maximum Thickness/2
TPM = .01
If TPMX < TPMY Then
  TPMY = TPMX
ElseIf TPMX >= TPMY Then
  TPMX = TPMY
End If

**** Draw Column
For I = 1 To SNodeNum(CProj)
  X1 = ((Column(CProj, I).Co_X - MinX) * TPMX) - Col + InitX
  Y1 = ((Column(CProj, I).Co_Y - MinY) * TPMY) - Col + InitY
  X2 = ((Column(CProj, I).Co_X - MinX) * TPMX) + Col + InitX
  Y2 = ((Column(CProj, I).Co_Y - MinY) * TPMY) + Col + InitY
  frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(ColumnColor), BF
Next I

**** Draw Element Conjunction in X axis
For I = 1 To SRow(CProj)
  For J = 1 To RowEle(I)
    CStart = EleColRow(CProj, 0, I, J).Start
    CEnd = EleColRow(CProj, 0, I, J).End
    X1 = ((Column(CProj, CStart).Co_X - MinX) * TPMX) + InitX
    Y1 = ((Column(CProj, CStart).Co_Y - MinY) * TPMY) + InitY
    X2 = ((Column(CProj, CEnd).Co_X - MinX) * TPMX) + InitX

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 48

```

    Y2 = ((Column(CProj, CEnd).Co_Y - MinY) * TPMY) + InitY
    frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(ElementColor)
Next J
Next I

'''' Draw Element Conjunction in Y axis
For I = 1 To SCol(CProj)
  For J = 1 To ColEle(I)
    CStart = EleColRow(CProj, 1, I, J).Start
    CEnd = EleColRow(CProj, 1, I, J).End
    X1 = ((Column(CProj, CStart).Co_X - MinX) * TPMX) + InitX
    Y1 = ((Column(CProj, CStart).Co_Y - MinY) * TPMY) + InitY
    X2 = ((Column(CProj, CEnd).Co_X - MinX) * TPMX) + InitX
    Y2 = ((Column(CProj, CEnd).Co_Y - MinY) * TPMY) + InitY
    frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(ElementColor)
  Next J
Next I

'''' Draw Moment in X axis
frmOutGraph(CProj).ForeColor = Colors(34)
Init1 = InitX
Init2 = InitY
For I = 1 To SRow(CProj)
  Trans_BElement False, I
  Cal_FEndMoment False, I
  For J = 1 To RowEle(I)
    InitX = ((Column(CProj, EleColRow(CProj, 0, I, J).Start).Co_X - MinX
) * TPMX) + Init1
    InitY = ((Column(CProj, EleColRow(CProj, 0, I, J).Start).Co_Y - MinY
) * TPMY) + Init2
    frmOutGraph(CProj).PSet (InitX, InitY), ProfileColor
    If J = 1 Then
      TX1 = InitX
      TY1 = InitY + 150
      frmOutGraph(CProj).PSet (TX1, TY1)
      frmOutGraph(CProj).Print Format$(Cal_BShear(J, 0), "0.00")
    End If
    frmOutGraph(CProj).PSet (InitX, InitY), ProfileColor
    For X = 0 To EleColRow(CProj, 0, I, J).Length Step AddSect
      M = Cal_BMoment(J, X, ML(CProj, J), MR(CProj, J))
      X1 = InitX + (X * TPMX)
      Y1 = InitY
      X2 = InitX + (X * TPMX)
      Y2 = InitY - (M * TPM)
      frmOutGraph(CProj).Line -(X2, Y2), Colors(MomentColor)
      frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(MomentColor)
    Next X
    TX1 = InitX + EleColRow(CProj, 0, I, J).Length * TPMX
    TY1 = InitY + 150
    frmOutGraph(CProj).PSet (TX1, TY1)
    frmOutGraph(CProj).Print Format$(MR(CProj, J) * (-1), "0.00")
  Next J
Next I

'''' Draw Moment in Y axis
frmOutGraph(CProj).ForeColor = Colors(39)
For I = 1 To SCol(CProj)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 49

```

Trans BElement True, I
Cal_FEndMoment True, I
For J = 1 To ColEle(I)
  InitX = (Column(CProj, EleColRow(CProj, 1, I, J).Start).Co_X - MinX)
* TPMX + Init1
  InitY = (Column(CProj, EleColRow(CProj, 1, I, J).Start).Co_Y - MinY)
* TPMY + Init2
  frmOutGraph(CProj).PSet (InitX, InitY), ProfileColor
  If J = 1 Then
    TX1 = InitX
    TY1 = InitY
    frmOutGraph(CProj).PSet (TX1, TY1)
    frmOutGraph(CProj).Print Format$(Cal_BShear(J, 0), "0.00")
  End If
  frmOutGraph(CProj).PSet (InitX, InitY), ProfileColor
  For Y = 0 To EleColRow(CProj, 1, I, J).Length Step AddSect
    M = Cal_BMoment(J, Y, ML(CProj, J), MR(CProj, J))
    X1 = InitX
    Y1 = InitY + (Y * TPMY)
    X2 = InitX - (M * TPM)
    Y2 = InitY + (Y * TPMY)
    frmOutGraph(CProj).Line -(X2, Y2), Colors(MomentColor)
    frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(MomentColor)
  Next Y
  TX1 = InitX
  TY1 = InitY + EleColRow(CProj, 1, I, J).Length * TPMY
  frmOutGraph(CProj).PSet (TX1, TY1)
  frmOutGraph(CProj).Print Format$(MR(CProj, J) * (-1), "0.00")
Next J '820438
Next I
frmOutGraph(CProj).ForeColor = Colors(34)
End Sub

Sub Screen_Profile ()
Dim XMax As Integer
Dim XMin As Integer
Dim YMax As Integer
Dim YMin As Integer
Dim InitX As Integer
Dim InitY As Integer
Dim Col As Integer
Dim CStart As Integer
Dim CEnd As Integer
Dim TPMX As Single
Dim TPMY As Single
Dim TPP As Single
Dim TPM As Single
Dim AddSect As Single
Dim X1 As Single
Dim Y1 As Single
Dim X2 As Single
Dim Y2 As Single
Dim X As Single
Dim Y As Single
Dim I As Integer
Dim J As Integer

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 50

```

Dim Init1 As Integer
Dim Init2 As Integer
Dim PF As Single
Dim M As Single
Dim Wu As Single
Dim ColumnColor As Integer
Dim ElementColor As Integer
Dim ProfileColor As Integer

''''
XMax = Column(CProj, 1).Co_X
XMin = Column(CProj, 1).Co_X
YMax = Column(CProj, 1).Co_Y
YMin = Column(CProj, 1).Co_Y
For I = 1 To SNodeNum(CProj)
  If XMax < Column(CProj, I).Co_X Then
    XMax = Column(CProj, I).Co_X
  End If
  If XMin > Column(CProj, I).Co_X Then
    XMin = Column(CProj, I).Co_X
  End If
  If YMax < Column(CProj, I).Co_Y Then
    YMax = Column(CProj, I).Co_Y
  End If
  If YMin > Column(CProj, I).Co_Y Then
    YMin = Column(CProj, I).Co_Y
  End If
Next I

'''' Init Value
frmOutGraph(CProj).Cls
Col = 50 'Column Width in Twips
AddSect = .5 'Additional Section
InitX = 1000 'Left Edge in Twips
InitY = 750 'Top Edge in Twips
ColumnColor = 34
ElementColor = 30
ProfileColor = 27
MomentColor = 17
Wu = DeadCo * DeadLoad(CProj) + LiveCo * LiveLoad(CProj)

''''
Trans_UniformLoad (Wu)'Trans. Load

''''
TPMX = (frmOutGraph(CProj).Width - (InitX * 2)) \ (XMax - XMin)'1m = T
PMX Twips for X axis
TPMY = (frmOutGraph(CProj).Height - (InitY * 2)) \ (YMax - YMin)'1m =
TPMY Twips for Y axis
TPP = 2000 / .3 'profile 30 cm use. 2000 Twips '1m =TPP Twips Fo
Profile Value ,Maximum = Maximum Thickness/2
TPM = .01
If TPMX < TPMY Then
  TPMY = TPMX
ElseIf TPMX >= TPMY Then
  TPMX = TPMY
End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 51

```

'Head
frmOutGraph(CProj).PSet (3000, 250)
frmOutGraph(CProj).Print "PROFILE OF TENDON"
frmOutGraph(CProj).PSet (3000, 350)
frmOutGraph(CProj).Print "======"
''' Draw Column
For I = 1 To SNodeNum(CProj)
  X1 = ((Column(CProj, I).Co_X - MinX) * TPMX) - Col + InitX
  Y1 = ((Column(CProj, I).Co_Y - MinY) * TPMY) - Col + InitY
  X2 = ((Column(CProj, I).Co_X - MinX) * TPMX) + Col + InitX
  Y2 = ((Column(CProj, I).Co_Y - MinY) * TPMY) + Col + InitY
  frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(ColumnColor), BF
Next I

''' Draw Element Conjunction in X axis
For I = 1 To SRow(CProj)
  For J = 1 To RowEle(I)
    CStart = EleColRow(CProj, 0, I, J).Start
    CEnd = EleColRow(CProj, 0, I, J).End
    X1 = ((Column(CProj, CStart).Co_X - MinX) * TPMX) + InitX
    Y1 = ((Column(CProj, CStart).Co_Y - MinY) * TPMY) + InitY
    X2 = ((Column(CProj, CEnd).Co_X - MinX) * TPMX) + InitX
    Y2 = ((Column(CProj, CEnd).Co_Y - MinY) * TPMY) + InitY
    frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(ElementColor)
  Next J
Next I

''' Draw Element Conjunction in Y axis
For I = 1 To SCol(CProj)
  For J = 1 To Colele(I)
    CStart = EleColRow(CProj, 1, I, J).Start
    CEnd = EleColRow(CProj, 1, I, J).End
    X1 = ((Column(CProj, CStart).Co_X - MinX) * TPMX) + InitX
    Y1 = ((Column(CProj, CStart).Co_Y - MinY) * TPMY) + InitY
    X2 = ((Column(CProj, CEnd).Co_X - MinX) * TPMX) + InitX
    Y2 = ((Column(CProj, CEnd).Co_Y - MinY) * TPMY) + InitY
    frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(ElementColor)
  Next J
Next I

Open DirApp + "profile.swp" For Random Access Read Write Lock Read Wri
te As #21 Len = 4
Dim WNum As Integer
Dim L As Single
WNum = 0
AddSect = .5

''' Draw Profile in X axis

Init1 = InitX
Init2 = InitY
For I = 1 To SRow(CProj)
  For J = 1 To RowEle(I)
    InitX = ((Column(CProj, EleColRow(CProj, 0, I, J).Start).Co_X - MinX)
) * TPMX) + Init1

```

OUTPUT.BAS - 52

```

      InitY = ((Column(CProj, EleColRow(CProj, 0, I, J).Start).Co_Y - MinY
) * TPMY) + Init2
      frmOutGraph(CProj).PSet (InitX, InitY), ProfileColor
      For X = 0 To EleColRow(CProj, 0, I, J).Length Step AddSect
        PF = Cal_Profile(False, I, J, X, 1, 1, emax(CProj))
        X1 = InitX + (X * TPMX)
        Y1 = InitY
        X2 = InitX + (X * TPMX)
        Y2 = InitY - (PF * TPP)
        frmOutGraph(CProj).Line -(X2, Y2), Colors(ProfileColor)
        frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(ProfileColor)
        'frmOutGraph(CProj).PSet (X1, Y1)
        'frmOutGraph(CProj).Print Format$(Y, "@@.0")
        'frmOutGraph(CProj).PSet (X2, Y2)
        'frmOutGraph(CProj).Print Format$(PF * 100, "00.0")
        'frmOutGraph(CProj).PSet (X2, Y2)

      Next X
    Next J
  Next I

  ''' Draw Profile in Y axis
  For I = 1 To SCol(CProj)
    For J = 1 To ColEle(I)
      InitX = (Column(CProj, EleColRow(CProj, 1, I, J).Start).Co_X - MinX)
* TPMX + Init1
      InitY = (Column(CProj, EleColRow(CProj, 1, I, J).Start).Co_Y - MinY)
* TPMY + Init2
      frmOutGraph(CProj).PSet (InitX, InitY), ProfileColor
      For Y = 0 To EleColRow(CProj, 1, I, J).Length Step AddSect
        PF = Cal_Profile(True, I, J, Y, 1, 1, emax(CProj))
        X1 = InitX
        Y1 = InitY + (Y * TPMY)
        X2 = InitX - (PF * TPP)
        Y2 = InitY + (Y * TPMY)
        frmOutGraph(CProj).Line -(X2, Y2), Colors(ProfileColor)
        frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(ProfileColor)
        'frmOutGraph(CProj).PSet (X1, Y1)
        'frmOutGraph(CProj).Print Format$(Y, "@@.0")
        'frmOutGraph(CProj).PSet (X2, Y2)
        'frmOutGraph(CProj).Print Format$(PF * 100, "00.0")
        'frmOutGraph(CProj).PSet (X2, Y2)
      Next Y
    Next J '820438
  Next I
  Close #21
End Sub

Sub Screen_Shear ()
  Dim XMax As Integer
  Dim XMin As Integer
  Dim YMax As Integer
  Dim YMin As Integer
  Dim InitX As Integer
  Dim InitY As Integer
  Dim Col As Integer

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 53

```

Dim CStart As Integer
Dim CEnd As Integer
Dim TPMX As Single
Dim TPMY As Single
Dim TPP As Single
Dim TPM As Single
Dim AddSect As Single
Dim X1 As Single
Dim Y1 As Single
Dim X2 As Single
Dim Y2 As Single
Dim X As Single
Dim Y As Single
Dim I As Integer
Dim J As Integer

```

```

Dim Init1 As Integer
Dim Init2 As Integer
Dim PF As Single
Dim v As Single
Dim Wu As Single
Dim ColumnColor As Integer
Dim ElementColor As Integer
Dim ProfileColor As Integer
Dim TX1 As Single
Dim TY1 As Single

```

```

''''
XMax = Column(CProj, 1).Co_X
XMin = Column(CProj, 1).Co_X
YMax = Column(CProj, 1).Co_Y
YMin = Column(CProj, 1).Co_Y
For I = 1 To SNodeNum(CProj)
  If XMax < Column(CProj, I).Co_X Then
    XMax = Column(CProj, I).Co_X
  End If
  If XMin > Column(CProj, I).Co_X Then
    XMin = Column(CProj, I).Co_X
  End If
  If YMax < Column(CProj, I).Co_Y Then
    YMax = Column(CProj, I).Co_Y
  End If
  If YMin > Column(CProj, I).Co_Y Then
    YMin = Column(CProj, I).Co_Y
  End If
Next I

```

```

'''' Init Value
frmOutGraph(CProj).Cls
Col = 50 'Column Width in Twips
AddSect = .1 'Additional Section
InitX = 1000 'Left Edge in Twips
InitY = 750 'Top Edge in Twips
ColumnColor = 34
ElementColor = 30
ProfileColor = 27

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 54

```

MomentColor = 17
frmOutGraph(CProj).PSet (3000, 250)
frmOutGraph(CProj).Print " SHEAR DIAGRAM "
frmOutGraph(CProj).PSet (3000, 350)
frmOutGraph(CProj).Print "===== "

Wu = DeadCo * DeadLoad(CProj) + LiveCo * LiveLoad(CProj)

****
Trans_UniformLoad (Wu)'Trans Load

****
TPMX = (frmOutGraph(CProj).Width - (InitX * 2)) \ (XMax - XMin)'1m = T
PMX Twips for X axis
TPMY = (frmOutGraph(CProj).Height - (InitY * 2)) \ (YMax - YMin)'1m =
TPMY Twips for Y axis
TPP = 2000 / .3 'profile 30 cm use 2000 Twips '1m =TFP Twips Fo
Profile Value ,Maximum = Maximum Thickness/2
TPM = .01
If TPMX < TPMY Then
    TPMY = TPMX
ElseIf TPMX >= TPMY Then
    TPMX = TPMY
End If

**** Draw Column
For I = 1 To SNodeNum(CProj)
    X1 = ((Column(CProj, I).Co_X - MinX) * TPMX) - Col + InitX
    Y1 = ((Column(CProj, I).Co_Y - MinY) * TPMY) - Col + InitY
    X2 = ((Column(CProj, I).Co_X - MinX) * TPMX) + Col + InitX
    Y2 = ((Column(CProj, I).Co_Y - MinY) * TPMY) + Col + InitY
    frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(ColumnColor), BF
Next I

**** Draw Element Conjunction in X axis
For I = 1 To SRow(CProj)
    For J = 1 To RowEle(I)
        CStart = EleColRow(CProj, 0, I, J).Start
        CEnd = EleColRow(CProj, 0, I, J).End
        X1 = ((Column(CProj, CStart).Co_X - MinX) * TPMX) + InitX
        Y1 = ((Column(CProj, CStart).Co_Y - MinY) * TPMY) + InitY
        X2 = ((Column(CProj, CEnd).Co_X - MinX) * TPMX) + InitX
        Y2 = ((Column(CProj, CEnd).Co_Y - MinY) * TPMY) + InitY
        frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(ElementColor)
    Next J
Next I

**** Draw Element Conjunction in Y axis
For I = 1 To SCol(CProj)
    For J = 1 To ColEle(I)
        CStart = EleColRow(CProj, 1, I, J).Start
        CEnd = EleColRow(CProj, 1, I, J).End
        X1 = ((Column(CProj, CStart).Co_X - MinX) * TPMX) + InitX
        Y1 = ((Column(CProj, CStart).Co_Y - MinY) * TPMY) + InitY
        X2 = ((Column(CProj, CEnd).Co_X - MinX) * TPMX) + InitX
        Y2 = ((Column(CProj, CEnd).Co_Y - MinY) * TPMY) + InitY
        frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(ElementColor)
    Next J
Next I

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 55

```

Next J
Next I

''' Draw Shear in X axis
frmOutGraph(CProj).ForeColor = Colors(34)
Init1 = InitX
Init2 = InitY
For I = 1 To SRow(CProj)
  Trans_BElement False, I
  Cal_FEndMoment False, I
  For J = 1 To RowEle(I)
    InitX = ((Column(CProj, EleColRow(CProj, 0, I, J).Start).Co_X - MinX
) * TPMX) + Init1
    InitY = ((Column(CProj, EleColRow(CProj, 0, I, J).Start).Co_Y - MinY
) * TPMY) + Init2
    frmOutGraph(CProj).PSet (InitX, InitY), ProfileColor
    If J = 1 Then
      TX1 = InitX
      TY1 = InitY + 150
      frmOutGraph(CProj).PSet (TX1, TY1)
      frmOutGraph(CProj).Print Format$(Cal_BShear(J, 0), "0.00")
    End If
    frmOutGraph(CProj).PSet (InitX, InitY), ProfileColor
    For X = 0 To EleColRow(CProj, 0, I, J).Length Step AddSect
      v = Cal_BShear(J, X)
      X1 = InitX + (X * TPMX)
      Y1 = InitY
      X2 = InitX + (X * TPMX)
      Y2 = InitY - (v * TPM)
      frmOutGraph(CProj).Line -(X2, Y2), Colors(MomentColor)
      frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(MomentColor)
    Next X
    TX1 = InitX + EleColRow(CProj, 0, I, J).Length * TPMX
    TY1 = InitY + 150
    frmOutGraph(CProj).PSet (TX1, TY1)
    frmOutGraph(CProj).Print Format$(Cal_BShear(J, EleColRow(CProj, 0, I
, J).Length), "0.00")
  Next J
Next I

frmOutGraph(CProj).ForeColor = Colors(39)
''' Draw Shear in Y axis
For I = 1 To SCol(CProj)
  Trans_BElement True, I
  Cal_FEndMoment True, I
  For J = 1 To Colele(I)
    InitX = (Column(CProj, EleColRow(CProj, 1, I, J).Start).Co_X - MinX)
) * TPMX + Init1
    InitY = (Column(CProj, EleColRow(CProj, 1, I, J).Start).Co_Y - MinY)
) * TPMY + Init2
    frmOutGraph(CProj).PSet (InitX, InitY), ProfileColor
    If J = 1 Then
      TX1 = InitX
      TY1 = InitY
      frmOutGraph(CProj).PSet (TX1, TY1)
      frmOutGraph(CProj).Print Format$(Cal_BShear(J, 0), "0.00")
    End If

```

OUTPUT.BAS - 56

```

frmOutGraph(CProj).PSet (InitX, InitY), ProfileColor
For Y = 0 To EleColRow(CProj, 1, I, J).Length Step AddSect
v = Cal_BShear(J, Y)
PF = Cal_Profile(True, I, J, Y, 1, 1, emax(CProj))
X1 = InitX
Y1 = InitY + (Y * TPMY)
X2 = InitX - (v * TPM)
Y2 = InitY + (Y * TPMY)
frmOutGraph(CProj).Line -(X2, Y2), Colors(MomentColor)
frmOutGraph(CProj).Line (X1, Y1)-(X2, Y2), Colors(MomentColor)
Next Y
TX1 = InitX
TY1 = InitY + EleColRow(CProj, 1, I, J).Length * TPMY
frmOutGraph(CProj).PSet (TX1, TY1)
frmOutGraph(CProj).Print Format$(Cal_BShear(J, EleColRow(CProj, 1, I
, J).Length), "0.00")
Next J '820438
Next I
frmOutGraph(CProj).ForeColor = Colors(34)
End Sub

Sub Screen_Text (Text As String)
frmList(CProj).FontName = "Courier"
frmList(CProj).FontSize = 7
TextToScreen = TextToScreen + " " + Text
If Len(TextToScreen) > 30000 Then
TextNum = TextNum + 1
frmList(CProj).txtListCompute(TextNum).Text = TextToScreen
TextToScreen = ""
End If
End Sub

Sub Screen_Try ()
Dim XMax As Integer
Dim XMin As Integer
Dim YMax As Integer
Dim YMin As Integer
Dim InitX As Integer
Dim InitY As Integer
Dim Col As Integer
Dim CStart As Integer
Dim CEnd As Integer
Dim TPMX As Single
Dim TPMY As Single
Dim TPP As Single
Dim TPM As Single
Dim AddSect As Single
Dim X1 As Single
Dim Y1 As Single
Dim X2 As Single
Dim Y2 As Single
Dim X As Single
Dim Y As Single
Dim I As Integer
Dim J As Integer
'''end moment
Dim MWL As Single

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 57

```

Dim MWR As Single
Dim MSL As Single
Dim MSR As Single
Dim MUL As Single
Dim MUR As Single

Dim Init1 As Integer
Dim Init2 As Integer
Dim PF As Single
Dim M As Single
Dim Wu As Single
Dim ColumnColor As Integer
Dim ElementColor As Integer
Dim ProfileColor As Integer
Dim CR As String * 1
Dim LF As String * 1
CR = Chr$(13)
LF = Chr$(10)
Const ucm = " cm"
Const um = " m"
Const ukg = " kg"
Const uksc = " ksc"
Const uksm = " kg/m^2"
Dim CNo As Integer
Dim RNum As Integer

'shear
'Column
Dim Vc0 As Single
Dim Vn0 As Single
'Drop
Dim Vc1 As Single
Dim Vn1 As Single
'Capital
Dim Vc2 As Single
Dim Vn2 As Single
'Diagnal
Dim Vc3 As Single
Dim Vn3 As Single

'''M design
Dim MDL As Single
Dim MDD As Single
Dim MDR As Single
'M design at column strip
Dim MCL As Single
Dim MCD As Single
Dim MCR As Single
'M design at middle strip
Dim MML1 As Single
Dim MMD1 As Single
Dim MMR1 As Single
Dim MML2 As Single
Dim MMD2 As Single
Dim MMR2 As Single
'''Area of reinforce steel
'Half middle strip 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 58

```
Dim MLB1 As Single
Dim MMB1 As Single
Dim MRB1 As Single
Dim MLT1 As Single
Dim MMT1 As Single
Dim MRT1 As Single
'Half middle strip 2
Dim MLB2 As Single
Dim MMB2 As Single
Dim MRB2 As Single
Dim MLT2 As Single
Dim MMT2 As Single
Dim MRT2 As Single
'column strip
Dim CLB As Single
Dim CMB As Single
Dim CRB As Single
Dim CLT As Single
Dim CMT As Single
Dim CRT As Single
```

```
'''Reinforcement design
Dim RMDL As Single
Dim RMDD As Single
Dim RMDR As Single
'M design at column strip
Dim RMCL As Single
Dim RMCD As Single
Dim RMCR As Single
'M design at middle strip
Dim RMML1 As Single
Dim RMD1 As Single
Dim RMR1 As Single
Dim RMML2 As Single
Dim RMD2 As Single
Dim RMR2 As Single
'''Area of reinforce steel
'Half middle strip 1
Dim RMLB1 As Single
Dim RMMB1 As Single
Dim RMRB1 As Single
Dim RMLT1 As Single
Dim RMMT1 As Single
Dim RMRT1 As Single
'Half middle strip 2
Dim RMLB2 As Single
Dim RMMB2 As Single
Dim RMRB2 As Single
Dim RMLT2 As Single
Dim RMMT2 As Single
Dim RMRT2 As Single
'column strip
Dim RCLB As Single
Dim RCMB As Single
Dim RCRB As Single
Dim RCLT As Single
Dim RCMT As Single
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
' OUTPUT.BAS - 59
```

```

Dim RCRT As Single

'Dim TextNo As Integer
'Dim Text0 As String
'ReDim Text(100) As String
'TextNo = 1

Dim TextNo As Integer
ReDim Text(100) As String
TextNo = 1

TB = Chr$(9)

AddSect = .5

''' Init
frmList(CProj).Cls
Text0 = ""

Open DirApp + "temp.pswp" For Random Access Read Write Lock Read Write
As #20 Len = 4
''' Information
Screen_Text CR + LF
Screen_Text "===== " + C
R + LF
Screen_Text "TWO WAY FLAT PLATE SLAB DESIGN IN PRESTRESS SYSTEM " + C
R + LF
Screen_Text " KING MONGKUT INSTITUTE OF TECHNOLOGY LADKRABANG " + C
R + LF
Screen_Text " CIVIL ENGINEERING DEPARTMENT " + C
R + LF
Screen_Text " BY SIAM KUNKATE & SOMPHONG KONGKRAPAN " + C
R + LF
Screen_Text " LISTING COMPUTATION CHECKING " + C
R + LF
Screen_Text "===== " + C
R + LF
Screen_Text CR + LF
Screen_Text CR + LF

Screen_Text " List of Flat Slab & Flat Plate Design" + CR + LF
Screen_Text "Data " + TB + " : " + SData(CProj).DateData + " ::
: Time : " + SData(CProj).TimeData + CR + LF
Screen_Text "File Name " + TB + " : " + SData(CProj).FileName + CR +
LF
Screen_Text "Project " + TB + " : " + SData(CProj).ProjectData + CR
+ LF
Screen_Text "Engineer " + TB + " : " + SData(CProj).Engineer + CR +
LF
Screen_Text "Address " + TB + " : " + SData(CProj).AddressData + CR
+ LF
Screen_Text "Information" + TB + " : " + SData(CProj).InformData + CR
+ LF

''' Data for Design
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 60

```

Screen_Text "Length Unit " + TB + " : Meter(m) ::: Load Unit : Ki
logram(Kg)" + CR + LF
Screen_Text "fc' (Slab) " + TB + " : " + Format$(GeneralData(CProj)
.SFc_ , ".00") + uksc + CR + LF
Screen_Text "fc' (Column) " + TB + " : " + Format$(GeneralData(CProj)
.CFc_ , ".00") + uksc + CR + LF
Screen_Text "fy (Strand) " + TB + " : " + Format$(GeneralData(CProj)
.SFs_ , ".00") + uksc + CR + LF
Select Case StrandIndex(CProj):
Case 0:
Case 1:
Case 2:
Case 3:
Case 4:
Case 5:
Case 6:
Case 7:
End Select
Screen_Text "Es (Strand) : " + Format$(Es, ".00") + CR + LF
Screen_Text "Ec (Slab) : " + Format$(GeneralData(CProj).SEc, ".00
") + uksc + CR + LF
Screen_Text "Ec (Column) : " + Format$(GeneralData(CProj).CEc, ".00
") + uksc + CR + LF
Screen_Text "Slab Thickness : " + Format$(100 * Thick(CProj), ".00") +
ucm + CR + LF
Screen_Text "Slab Covering : " + Format$((Thick(CProj) - d(CProj)) *
100, ".00") + ucm + CR + LF
Screen_Text "Drape : " + Format$((Thick(CProj) - 2 * (Thick(C
Proj) - d(CProj))) * 100, ".00") + ucm + CR + LF

''' Column Data
frmList(CProj).FontSize = 7
frmList(CProj).FontBold = True
Screen_Text "Column Data" + CR + LF
Screen_Text CR + LF
Screen_Text "Column No." + TB + "Co_X(m)" + TB + TB + "Co_Y(m)" + TB
+ "SupportType" + TB + " Length(m) " + CR + LF
Screen_Text CR + LF
CNo = 0
For I = 1 To SRow(CProj)
For J = 1 To RowEle(I)
CNo = CNo + 1
Co_X = Column(CProj, EleColRow(CProj, 0, I, J).Start).Co_X
Co_Y = Column(CProj, EleColRow(CProj, 0, I, J).Start).Co_Y
Support = Column(CProj, EleColRow(CProj, 0, I, J).Start).SupportType
Length = Column(CProj, EleColRow(CProj, 0, I, J).Start).Length
Screen_Text Format$(CNo, "@@@") + TB + TB + Format$(Co_X, "0.00E+00"
) + TB + TB + Format$(Co_Y, "0.00E+00") + TB + TB + Format$(Support, "@@")
+ TB + Format$(Length, "0.00E+00") + CR + LF
If J = RowEle(I) Then
CNo = CNo + 1
Co_X = Column(CProj, EleColRow(CProj, 0, I, RowEle(I)).End).Co_X
Co_Y = Column(CProj, EleColRow(CProj, 0, I, RowEle(I)).End).Co_Y
Support = Column(CProj, EleColRow(CProj, 0, I, RowEle(I)).End).Supp
ortType
Length = Column(CProj, EleColRow(CProj, 0, I, RowEle(I)).End).Lengt

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 61

```

h
    Screen_Text Format$(CNo, "@@@") + TB + TB + Format$(Co_X, "0.00E+00
") + TB + TB + Format$(Co_Y, "0.00E+00") + TB + TB + Format$(Support, "@@"
) + TB + Format$(Length, "0.00E+00") + CR + LF
    End If
    Next J
Next I

''' Element Data
Dim LCol As Integer
Dim RCol As Integer
Dim LBay As Single
Dim RBay As Single
'Row
Screen_Text CR + LF
Screen_Text CR + LF
Screen_Text "Row Element Data" + CR + LF
Screen_Text CR + LF
Screen_Text " Row :Element No.:Start Column: End Column: Length
: Top Bay : Bottom Bay " + CR + LF
Screen_Text "
(m)
(m) " + CR + LF
For I = 1 To SRow(CProj)
    For J = 1 To RowEle(I)
        LCol = EleColRow(CProj, 0, I, J).Start
        RCol = EleColRow(CProj, 0, I, J).End
        Length = EleColRow(CProj, 0, I, J).Length
        LBay = EleColRow(CProj, 0, I, J).L1
        RBay = EleColRow(CProj, 0, I, J).L2
        Screen_Text Format$(I, "#####") + Format$(J, "#####
") + Format$(LCol, "#####") + Format$(RCol, "#####
@@") + Format$(Length, "#####.00") + Format$(LBay, "#####.00")
+ Format$(RBay, "#####.00") + CR + LF
    Next J
Next I
'Column
Screen_Text CR + LF
Screen_Text CR + LF
Screen_Text "Row Element Data" + CR + LF
Screen_Text CR + LF
Screen_Text "Column:Element No.:Start Column: End Column: Length
: Left Bay : Right Bay " + CR + LF
Screen_Text "
(m)
(m) " + CR + LF
For I = 1 To SCol(CProj)
    For J = 1 To Colele(I)
        LCol = EleColRow(CProj, 1, I, J).Start
        RCol = EleColRow(CProj, 1, I, J).End
        Length = EleColRow(CProj, 1, I, J).Length
        LBay = EleColRow(CProj, 1, I, J).L1
        RBay = EleColRow(CProj, 1, I, J).L2
        Screen_Text Format$(I, "#####") + Format$(J, "#####
") + Format$(LCol, "#####") + Format$(RCol, "#####
@@") + Format$(Length, "#####.00") + Format$(LBay, "#####.00")
+ Format$(RBay, "#####.00") + CR + LF
    Next J
Next I

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 62

```

Screen_Text CR + LF
Screen_Text CR + LF
Screen_Text CR + LF
Dim SSystem As String * 50
Select Case GeneralData(CProj).SlabIndex
Case 0:
  SSystem = "Flat Plate"
Case 1:
  SSystem = "Flat Slab With Drop Panel"
Case 2:
  SSystem = "Flat Slab With Column Capital"
Case 3:
  SSystem = "Flat Slab With Column Drop Panel And Column Capital"
End Select
Screen_Text "SLAB SYSTEM : " + SSystem + CR + LF
Screen_Text CR + LF
Screen_Text "Load factor ::: " + CR + LF
Screen_Text "Wu = " + Format$(DeadCo, "0.0") + "DL" + Format$(LiveCo,
"0.0") + "LL" + CR + LF
Screen_Text "Unfactored Dead Load          = " + Format$(DeadLoad(CPr
oj), "0.00") + uksm + CR + LF
Screen_Text "Unfactored Live Load          = " + Format$(LiveLoad(CPr
oj), "0.00") + uksm + CR + LF
Screen_Text "Unfactored Supper imposed load = " + Format$(SuperLoad(CP
roj), "0.00") + uksm + CR + LF
Screen_Text ""

Screen_Text "Column " + Chr$(9) + "          Punching at Column " + Chr$
(9) + Chr$(9) + "          Punching at Drop Panel " + Chr$(9) + "
Punching at Capital " + Chr$(9) + "          Diagonal shear
" + CR + LF
Screen_Text " No. " + Chr$(9) + " Vc(kg) " + Chr$(9) + Chr$(9) + "
Vn(kg) " + Chr$(9) + Chr$(9) + " Vc(kg) " + Chr$(9) + Chr$(9) + " Vn(kg)
" + Chr$(9) + Chr$(9) + " Vc(kg) " + Chr$(9) + Chr$(9) + " Vn(kg) " +
Chr$(9) + Chr$(9) + " Vc(kg) " + Chr$(9) + Chr$(9) + " Vn(kg) " + Chr$
(9) + CR + LF
Screen_Text CR + LF
RNum = 2
RIden = 2
Dim ColumnNo As Integer
ColumnNo = 0
For I = 1 To SRow(CProj)
  For J = 1 To RowEle(I)
    ColumnNo = ColumnNo + 1
    CNo = EleColRow(CProj, 0, I, J).Start
    If Column(CProj, CNo).SupportType <> 2 Then 'If Not Free, Mean That i
s Column support
      RIden = RNum + ((CNo - 1) * 8)
      Get #20, RIden + 1, Vc0
      Get #20, RIden + 2, Vn0
      Get #20, RIden + 3, Vc1
      Get #20, RIden + 4, Vn1
      Get #20, RIden + 5, Vc2
      Get #20, RIden + 6, Vn2
      Get #20, RIden + 7, Vc3
      Get #20, RIden + 8, Vn3
      Screen_Text Format$(ColumnNo, "@@@@") + Chr$(9) + Format$(Vc0, "0.0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 63

```

000E+00") + Chr$(9) + Format$(Vn0, "0.0000E+00") + Chr$(9) + Format$(Vc1,
"0.0000E+00") + Chr$(9) + Format$(Vn1, "0.0000E+00") + Chr$(9) + Format$(V
c2, "0.0000E+00") + Chr$(9) + Format$(Vn2, "0.0000E+00") + Chr$(9) + Forma
t$(Vc3, "0.0000E+00") + Chr$(9) + Format$(Vn3, "0.0000E+00") + CR + LF
End If
If J = RowEle(I) Then
ColumnNo = ColumnNo + 1
CNo = EleColRow(CProj, 0, I, J).End
If Column(CProj, CNo).SupportType <> 2 Then
Riden = RNum + ((CNo - 1) * 8)
Get #20, Riden + 1, Vc0
Get #20, Riden + 2, Vn0
Get #20, Riden + 3, Vc1
Get #20, Riden + 4, Vn1
Get #20, Riden + 5, Vc2
Get #20, Riden + 6, Vn2
Get #20, Riden + 7, Vc3
Get #20, Riden + 8, Vn3
Screen_Text Format$(ColumnNo, "###") + Chr$(9) + Format$(Vc0, "0.
0000E+00") + Chr$(9) + Format$(Vn0, "0.0000E+00") + Chr$(9) + Format$(Vc1,
"0.0000E+00") + Chr$(9) + Format$(Vn1, "0.0000E+00") + Chr$(9) + Format$(
Vc2, "0.0000E+00") + Chr$(9) + Format$(Vn2, "0.0000E+00") + Chr$(9) + Form
at$(Vc3, "0.0000E+00") + Chr$(9) + Format$(Vn3, "0.0000E+00") + CR + LF
End If
End If
Next J
Next I
'Drop panel & Capital
CNo = 0
ColumnNo = 0
Screen_Text CR + LF
Screen_Text "Dimension of drop & column capital"
Screen_Text CR + LF
Screen_Text " No. " + Chr$(9) + "Drop Width" + Chr$(9) + "Drop Length"
+ Chr$(9) + "Drop Thick" + Chr$(9) + "Capital Width" + Chr$(9) + "Capital
Length" + Chr$(9) + CR + LF
Screen_Text CR + LF
For I = 1 To SRow(CProj)
For J = 1 To RowEle(I)
CNo = CNo + 1
ColumnNo = EleColRow(CProj, 0, I, J).Start
DropX = Column(CProj, ColumnNo).DropX
DropY = Column(CProj, ColumnNo).DropY
DropThick = Column(CProj, ColumnNo).DropThick
CapitalX = Column(CProj, ColumnNo).CapitalX
CapitalY = Column(CProj, ColumnNo).CapitalY
Screen_Text Format$(CNo, "###") + Chr$(9) + Format$(DropX, "0.0000E
+00") + Chr$(9) + Format$(DropY, "0.0000E+00") + Chr$(9) + Format$(DropThi
ck, "0.0000E+00") + Chr$(9) + Format$(CapitalX, "0.0000E+00") + Chr$(9) +
Format$(CapitalY, "0.0000E+00") + CR + LF
If J = RowEle(I) Then
CNo = CNo + 1
ColumnNo = EleColRow(CProj, 0, I, RowEle(I)).End
DropX = Column(CProj, ColumnNo).DropX
DropY = Column(CProj, ColumnNo).DropY
DropThick = Column(CProj, ColumnNo).DropThick
CapitalX = Column(CProj, ColumnNo).CapitalX

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 64

```

      CapitalY = Column(CProj, ColumnNo).CapitalY
      Screen_Text Format$(CNo, "###") + Chr$(9) + Format$(DropX, "0.0000
E+00") + Chr$(9) + Format$(DropY, "0.0000E+00") + Chr$(9) + Format$(DropTh
ick, "0.0000E+00") + Chr$(9) + Format$(CapitalX, "0.0000E+00") + Chr$(9) +
Format$(CapitalY, "0.0000E+00") + CR + LF
      End If
    Next J
  Next I
  'declare
  Dim Lmax As Single
  Dim Bbeam As Single
  Dim P As Single
  Dim Top As Single
  Dim Bottom As Single
  TB = Chr$(9)

  Text1 = ""
  Screen_Text CR + LF
  Screen_Text "===== " + C
R + LF
  Screen_Text "CHECK CRACKING MOMENT & ULTIMATE MOMENT & DEFLECTION" + C
R + LF
  Screen_Text "===== " + C
R + LF
  Screen_Text CR + LF
  Screen_Text CR + LF
  '""check Cracking Moment & Deflection
  Wu = DeadCo * DeadLoad(CProj) + LiveCo * LiveLoad(CProj) + DeadCo * Su
perLoad(CProj)
  'text flexeral stress in row
  Lmax = 0
  Bbeam = 0
  Mu = 0
  For I = 1 To SRow(CProj)
    Trans_UniformLoad Wu
    Trans_BElement False, I
    Cal_UltimateEndMoment False, I
    Mu = Find_MuMax(False, I)
    Find_BL False, I, Lmax, Bbeam
    P = .775 * GeneralData(CProj).SFs_ * Find_AsStrand(Lmax, Bbeam)
    Screen_Text CR + LF
  Screen_Text " ROW : " + Format$(I, "###") + CR + LF
  Screen_Text " No of tendon = " + Format$(Find_AsStrand(Lmax, Bbeam),
"###") + TB + "Unit" + CR + LF
  Screen_Text " Tension per 1 unit tendon = " + Format$(.775 * GeneralD
ata(CProj).SFs_, "0.00") + " Kg" + CR + LF
  Screen_Text " Mcr = " + Format$(Check_CrackMoment(P, Bbeam), "0.00")
+ " kg-m" + CR + LF
  Screen_Text " Mn = " + Format$(Check_MnMax(Find_AsStrand(Lmax, Bbeam
), Bbeam), "0.00") + " kg-m" + CR + LF
  Screen_Text " Mu = " + Format$(Mu, ".00") + " Kg-m" + CR + LF
  Screen_Text "CHECK DEFLECTION AT AFTER TENSIONED" + CR + LF
  Screen_Text " Deflection(Actual) = " + Format$(Find_MaxDeflection(
False, I, 1), "0.00E+00") + um + CR + LF
  Screen_Text " Deflection(Allowable) = " + Format$(Check_Deflection(Fa
lse, I), "0.00E+00") + um + CR + LF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 66

```

Wu = DeadLoad(CProj)
'text flexeral stress in row
Lmax = 0
Bbeam = 0
Mu = 0
For I = 1 To SRow(CProj)
  Trans_UniformLoad Wu
  Trans_BEElement False, I
  Cal_ElasticEndMoment False, I
  Find_BL False, I, Lmax, Bbeam
  P = .775 * GeneralData(CProj).SFs_ * Find_AsStrand(Lmax, Bbeam)
  Screen_Text "Row No. " + " Element No." + TB + "Distance From Left"
+ TB + " Position of tendon" + TB + TB + TB + "Checking Stress" + CR + LF
  Screen_Text " " + TB + TB + " " + TB + TB + TB + TB +
TB + TB + TB + "Top" + TB + TB + "Bottom" + CR + LF
  Screen_Text CR + LF
  For J = 1 To RowEle(I)
    For X = 0 To EleColRow(CProj, 0, I, J).Length Step AddSect
      M = Cal_BMoment(J, X, ML(CProj, J), MR(CProj, J))
      PF = Cal_Profile(False, I, J, X, 1, 1, emax(CProj))
      Find_Stress P, M, PF, EleColRow(CProj, 0, I, J).L1 / 2 + EleColRow(
CProj, 0, I, J).L2 / 2, Top, Bottom
      Screen_Text Format$(I, "#####") + TB + TB + Format$(J, "#####") + T
B + TB + Format$(X, "0.00E+00") + TB + TB + Format$(PF * 100, "0.000E+00")
+ TB + TB + Format$(Top / 10000, "0.000E+00") + TB + Format$(Bottom / 100
00, "0.00E+00") + CR + LF
      Next X
      Screen_Text "======" + CR + LF
=====
    Next J
  Next I

'text2 flexeral stress in Column
Lmax = 0
Bbeam = 0
Mu = 0
For I = 1 To SRow(CProj)
  Trans_UniformLoad Wu
  Trans_BEElement True, I
  Cal_ElasticEndMoment True, I
  Find_BL True, I, Lmax, Bbeam
  P = .775 * GeneralData(CProj).SFs_ * Find_AsStrand(Lmax, Bbeam)
  Screen_Text CR + LF
  Screen_Text "Column No." + " Element No." + TB + "Distance From Left"
+ TB + " Position of tendon" + TB + TB + TB + "Checking Stress" + CR + L
F
  Screen_Text " " + TB + TB + " " + TB + TB + TB + TB +
TB + TB + TB + "Top" + TB + TB + "Bottom" + CR + LF
  For J = 1 To RowEle(I)
    For X = 0 To EleColRow(CProj, 0, I, J).Length Step AddSect
      M = Cal_BMoment(J, X, ML(CProj, J), MR(CProj, J))
      PF = Cal_Profile(False, I, J, X, 1, 1, emax(CProj))
      Find_Stress P, M, PF, EleColRow(CProj, 0, I, J).L1 / 2 + EleColRow(
CProj, 0, I, J).L2 / 2, Top, Bottom
      Screen_Text Format$(I, "#####") + TB + TB + Format$(J, "#####") + T
B + TB + Format$(X, "0.00E+00") + TB + TB + Format$(PF * 100, "0.000E+00")
+ TB + TB + Format$(Top / 10000, "0.000E+00") + TB + Format$(Bottom / 100

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS ~ 67

```

00, "0.00E+00") + CR + LF
Next X
Screen_Text "=====
===== " + CR + LF
Next J
Next I

AddSect = .5
'''check at SURVICE load
Text3 = ""
Screen_Text CR + LF
Screen_Text "===== " +
CR + LF
Screen_Text "CHECK STRESS AT TOP & BOTTOM OF SLAB AT TRANSFER LOAD" +
CR + LF
Screen_Text "===== " +
CR + LF
Screen_Text CR + LF
Screen_Text CR + LF

Wu = DeadLoad(CProj) + LiveLoad(CProj) + SuperLoad(CProj)
Text3 flexeral stress in row
Lmax = 0
Bbeam = 0
Mu = 0
For I = 1 To SRow(CProj)
Trans_UniformLoad Wu
Trans_BEElement False, I
Cal_ServiceEndMoment False, I
Find_BL True, I, Lmax, Bbeam
P = .775 * GeneralData(CProj).SFs_ * Find_AsStrand(Lmax, Bbeam)
Screen_Text CR + LF
Screen_Text "Row No. " + " Element No." + TB + "Distance From Lef
t" + TB + " Position of tendon" + TB + TB + TB + "Checking Stress" + CR +
LF
Screen_Text " " + TB + TB + " " + TB + TB + TB + TB +
TB + TB + TB + "Top" + TB + TB + "Bottom" + CR + LF
Screen_Text CR + LF
For J = 1 To RowEle(I)
For X = 0 To EleColRow(CProj, 0, I, J).Length Step AddSect
M = Cal_BMoment(J, X, ML(CProj, J), MR(CProj, J))
PF = Cal_Profile(False, I, J, X, 1, 1, emax(CProj))
Find_Stress P, M, PF, EleColRow(CProj, 0, I, J).L1 / 2 + EleColRow(
CProj, 0, I, J).L2 / 2, Top, Bottom
Screen_Text Format$(I, "#####") + TB + TB + Format$(J, "#####") + T
B + TB + Format$(X, "0.00E+00") + TB + TB + Format$(PF * 100, "0.000E+00")
+ TB + TB + Format$(Top / 10000, "0.000E+00") + TB + Format$(Bottom / 100
00, "0.00E+00") + CR + LF
Next X
Screen_Text "=====
===== " + CR + LF
Next J
Next I

Text3 flexeral stress in Column
Lmax = 0
Bbeam = 0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 68

```

Mu = 0
For I = 1 To SRow(CProj)
  Trans_UniformLoad Wu
  Trans_BEElement True, I
  Cal_ServiceEndMoment False, I
  Find_BL True, I, Lmax, Bbeam
  P = .775 * GeneralData(CProj).SFs_ * Find_AsStrand(Lmax, Bbeam)
  Screen_Text CR + LF
  Screen_Text "Column No." + " Element No." + TB + "Distance From Left
" + TB + " Position of tendon" + TB + TB + TB + "Checking Stress" + CR + L
F
  Screen_Text "          " + TB + TB + "          " + TB + TB + TB + TB +
TB + TB + TB + "Top" + TB + TB + "Bottom" + CR + LF
  Screen_Text CR + LF
  For J = 1 To RowEle(I)
    For X = 0 To EleColRow(CProj, 0, I, J).Length Step AddSect
      M = Cal_BMoment(J, X, ML(CProj, J), MR(CProj, J))
      PF = Cal_Profile(False, I, J, X, 1, 1, emax(CProj))
      Find_Stress P, M, PF, EleColRow(CProj, 0, I, J).L1 / 2 + EleColRow(
CProj, 0, I, J).L2 / 2, Top, Bottom
      Screen_Text Format$(I, "#####") + TB + TB + Format$(J, "#####") + T
B + TB + Format$(X, "0.00E+00") + TB + TB + Format$(PF * 100, "0.000E+00")
+ TB + TB + Format$(Top / 10000, "0.000E+00") + TB + Format$(Bottom / 100
00, "0.00E+00") + CR + LF
      Next X
      Screen_Text "-----" + CR + LF
    Next J
  Next I

For I = 1 To 20
  frmList(CProj).txtListCompute(I).Top = 0
  frmList(CProj).txtListCompute(I).Left = 0
  frmList(CProj).txtListCompute(I).Width = frmList(CProj).ScaleWidth
  frmList(CProj).txtListCompute(I).Height = frmList(CProj).ScaleHeight
  frmList(CProj).txtListCompute(I).Visible = False
Next I
frmList(CProj).txtListCompute(TextNum + 1).Text = TextToScreen
TextToScreen = ""
TextModule = TextNum + 1
TextNum = 0
CurrentPage = 1
frmList(CProj).txtListCompute(1).Visible = True
frmMain.cmdPrevious.Visible = True
frmMain.cmdNext.Visible = True
If TextModule = 1 Then
  frmMain.cmdPrevious.Enabled = False
  frmMain.cmdNext.Enabled = False
ElseIf TextModule > 1 Then
  frmMain.cmdPrevious.Enabled = False
  frmMain.cmdNext.Enabled = True
End If
Close #20
End Sub

Sub Sum_Text (Text0 As String, TextNo As Integer, TextAdd As String)
  On Error GoTo ErrorText

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT.BAS - 69

```
Text0 = Text0 + " " + TextAdd
If Len(Text0) > 30000 Then
  TextNo = TextNo + 1
End If

Exit Sub
ErrorText: For I = 1 To TextNo
  frmList(CProj).txtListCompute(I).Text = Text0
Next I
Exit Sub
End Sub
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SYSTEMSL.BAS - 1

Option Explicit

Function Cal_Thick (L As Single, Index As Integer) As Single

Dim T As Single
Dim tb As Single
Dim Lb As Single 'beam

Select Case Index

Case 0: 'Flat Plate

T = L / 42

Case 1: 'Panel

T = L / 45

Case 2: 'Capital

T = L / 42

Case 3: 'Both

T = L / 42

Case 4: 'Band Beam

T = L / 52

tb = Lb / 28

End Select

Dim x As Integer

T = T * 100

x = Int(T)

If T - x > 0 Then

T = x + 1 '»N'çÖé'

ElseIf T - x = 0 Then

T = x

End If

T = T / 100

Cal_Thick = T

End Function



OUTPUTF.BAS - 1

```

Sub Print_FlatListCompute ()
    frmMain.cmdlgSystem.Flags = PD_DISABLEPRINTTOFILE
    frmMain.cmdlgSystem.CancelError = True
    On Error GoTo DlgErrorPrintFlatListCompute
    frmMain.cmdlgSystem.Action = 5

    Dim XMax As Integer
    Dim XMin As Integer
    Dim YMax As Integer
    Dim YMin As Integer
    Dim InitX As Integer
    Dim InitY As Integer
    Dim Col As Integer
    Dim CStart As Integer
    Dim CEnd As Integer
    Dim TPMX As Single
    Dim TPMY As Single
    Dim TPP As Single
    Dim TPM As Single
    Dim AddSect As Single
    Dim X1 As Single
    Dim Y1 As Single
    Dim X2 As Single
    Dim Y2 As Single
    Dim X As Single
    Dim Y As Single
    Dim I As Integer
    Dim J As Integer

    Dim Init1 As Integer
    Dim Init2 As Integer
    Dim PF As Single
    Dim M As Single
    Dim Wu As Single
    Dim ColumnColor As Integer
    Dim ElementColor As Integer
    Dim ProfileColor As Integer
    Dim CR As String * 1
    Dim LF As String * 1
    CR = Chr$(13)
    LF = Chr$(10)
    Const ucm = " cm"
    Const um = " m"
    Const ukg = " kg"
    Const uksc = " ksc"
    Const uksm = " kg/m^2"
    Dim CNo As Integer
    Dim RNum As Integer

    'shear
    'Column
    Dim Vc0 As Single
    Dim Vn0 As Single
    'Drop
    Dim Vc1 As Single

```

'OUTPUTF.BAS - 2

```

Dim Vn1 As Single
'Capital

Dim Vc2 As Single
Dim Vn2 As Single
'Diagnal
Dim Vc3 As Single
Dim Vn3 As Single

''''M design
Dim MDL As Single
Dim MDD As Single
Dim MDR As Single
'M design at column strip
Dim MCL As Single
Dim MCD As Single
Dim MCR As Single
'M design at middle strip
Dim MML1 As Single
Dim MMD1 As Single
Dim MMR1 As Single
Dim MML2 As Single
Dim MMD2 As Single
Dim MMR2 As Single
''''Area of reinforce steel
'Half middle strip 1
Dim MLB1 As Single
Dim MMB1 As Single
Dim MRB1 As Single
Dim MLT1 As Single
Dim MMT1 As Single
Dim MRT1 As Single
'Half middle strip 2
Dim MLB2 As Single
Dim MMB2 As Single
Dim MRB2 As Single
Dim MLT2 As Single
Dim MMT2 As Single
Dim MRT2 As Single
'column strip
Dim CLB As Single
Dim CMB As Single
Dim CRB As Single
Dim CLT As Single
Dim CMT As Single
Dim CRT As Single

''''Reinforcement design
Dim RMDL As Single
Dim RMDD As Single
Dim RMDR As Single
'M design at column strip
Dim RMCL As Single
Dim RMCD As Single
Dim RMCR As Single
'M design at middle strip
Dim RMML1 As Single

```



OUTPUTF.BAS - 3

```

Dim RMD1 As Single
Dim RMR1 As Single
Dim RML2 As Single
Dim RMD2 As Single
Dim RMR2 As Single
'''Area of reinforce steel
'Half middle strip 1
Dim RMLB1 As Single
Dim RMB1 As Single
Dim RMRB1 As Single
Dim RMLT1 As Single
Dim RMT1 As Single
Dim RMRT1 As Single
'Half middle strip 2
Dim RMLB2 As Single
Dim RMB2 As Single
Dim RMRB2 As Single
Dim RMLT2 As Single
Dim RMT2 As Single
Dim RMRT2 As Single
'column strip
Dim RCLB As Single
Dim RCMB As Single
Dim RCRB As Single
Dim RCLT As Single
Dim RCMT As Single
Dim RCRT As Single

```

```

Dim TextNo As Integer
ReDim Text(100) As String
TextNo = 1

```

```

Open DirApp + "temp.swp" For Random Access Read Write Lock Read Write
As #20 Len = 4

```

```

''' Init
frmList(CProj).Cls

```

```

''' Information

```

```

Print_Text 0, CR + LF
Print_Text 0, "===== " +
CR + LF
Print_Text 0, "          TWO WAY FLAT PLATE SLAB DESIGN          " +
CR + LF
Print_Text 0, " KING MONGKUT INSTITUTE OF TECHNOLOGY LADKRABANG " +
CR + LF
Print_Text 0, "          CIVIL ENGINEERING DEPARTMENT          " +
CR + LF
Print_Text 0, "          BY SIAM KUNKATE & PATCHARAWALAI MAITA          " +
CR + LF
Print_Text 0, "          LISTING COMPUTATION CHECKING          " +
CR + LF
Print_Text 0, "===== " +
CR + LF
Print_Text 0, CR + LF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUTF.BAS - 4

```

Print_Text 0, CR + LF

Print_Text 0, "Date" : " + Format$(Now, "dddd mm/dd/yyyy ") +
CR + LF
Print_Text 0, "Time" : " + Format$(Now, "hh:mm:ss AM/PM") + C
R + LF
Print_Text 0, "File Name" : " + SData(CProj).FileName + CR + LF
Print_Text 0, "Project" : " + SData(CProj).ProjectData + CR + LF
Print_Text 0, "Engineer" : " + SData(CProj).Engineer + CR + LF
Print_Text 0, "Address" : " + SData(CProj).AddressData + CR + LF
Print_Text 0, "Information" : " + SData(CProj).InformData + CR + LF

''' Data for Design
Print_Text 0, "Length Unit" : Meter(m)" + CR + LF
Print_Text 0, "Load Unit" : Kilogram(Kg)" + CR + LF
Print_Text 0, "fc'" : " + Format$(GeneralData(CProj).SFc_, "
.00") + uksc + CR + LF
Print_Text 0, "fy (Reinforce)" : " + Format$(GeneralData(CProj).RFs_, "
.00") + uksc + CR + LF
Print_Text 0, "Ec (Slab)" : " + Format$(GeneralData(CProj).SEc, ".
00") + uksc + CR + LF
Print_Text 0, "Ec (Column)" : " + Format$(GeneralData(CProj).CEc, ".
00") + uksc + CR + LF

''' Column Data
frmList(CProj).FontBold = True
Print_Text 0, CR + LF
Print_Text 0, CR + LF
Print_Text 0, "COLUMN DATA" + CR + LF
Print_Text 0, "======" + CR + LF
Print_Text 0, CR + LF
Print_Text 0, "COLUMN NO. CO_X(m) CO_Y(m) LENGTH(m) " + CR
+ LF
Print_Text 0, CR + LF
CNo = 0
For I = 1 To SRow(CProj)
For J = 1 To RowEle(I)
CNo = CNo + 1
Co_X = Column(CProj, EleColRow(CProj, 0, I, J).Start).Co_X
Co_Y = Column(CProj, EleColRow(CProj, 0, I, J).Start).Co_Y
Support = Column(CProj, EleColRow(CProj, 0, I, J).Start).SupportType
Length = Column(CProj, EleColRow(CProj, 0, I, J).Start).Length
Print_Text 1, Format$(CNo, "#####") + s5 + Format$(Co_X, "0.00E+0"
) + s5 + Format$(Co_Y, "0.00E+0") + s5 + Format$(Length, "0.00E+0") + CR +
LF
If J = RowEle(I) Then
CNo = CNo + 1
Co_X = Column(CProj, EleColRow(CProj, 0, I, RowEle(I)).End).Co_X
Co_Y = Column(CProj, EleColRow(CProj, 0, I, RowEle(I)).End).Co_Y
Support = Column(CProj, EleColRow(CProj, 0, I, RowEle(I)).End).Supp
ortType
Length = Column(CProj, EleColRow(CProj, 0, I, RowEle(I)).End).Lengt
h
Print_Text 1, Format$(CNo, "#####") + s5 + Format$(Co_X, "0.00E+0
") + s5 + Format$(Co_Y, "0.00E+0") + s5 + Format$(Length, "0.00E+0") + CR
+ LF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUTF.BAS - 5

```

End If
Next J
Next I
Print_Text 0, "===== " + CR + L
F
''' Element Data
Dim LCol As Integer
Dim RCol As Integer
Dim LBay As Single
Dim RBay As Single
'Row
Print_Text 0, CR + LF
Print_Text 0, CR + LF
Print_Text 0, "DESIGN STRIP IN ROW" + CR + LF
Print_Text 0, "===== " + CR + LF
Print_Text 0, CR + LF
Print_Text 0, " ROW ELEMENT START-COL END-COL LENGTH(m) TOP-BA
Y(m) BOTTOM-BAY(m)" + CR + LF
Print_Text 0, CR + LF
For I= 1 To SRow(CProj)
For J = 1 To RowEle(I)
LCol = EleColRow(CProj, 0, I, J).Start
RCol = EleColRow(CProj, 0, I, J).End
Length = EleColRow(CProj, 0, I, J).Length
LBay = EleColRow(CProj, 0, I, J).L1
RBay = EleColRow(CProj, 0, I, J).L2
Print_Text 2, Format$(I, "###") + Format$(J, "#####") + Format$(
LCol, "#####") + Format$(RCol, "#####") + s5 + Format$(Length, "
0.00E+0") + s5 + Format$(LBay, "0.00E+0") + s5 + s2 + Format$(RBay, "0.00E
+0") + CR + LF
Next J
Next I

Print_Text 0, "=====
----- " + CR + LF
'Column
Print_Text 0, CR + LF
Print_Text 0, CR + LF
Print_Text 0, "DESIGN STRIP IN COLUMN" + CR + LF
Print_Text 0, "===== " + CR + LF
Print_Text 0, CR + LF
Print_Text 0, " COLUMN ELEMENT START-COL END-COL LENGTH(m) TOP-BA
Y(m) BOTTOM-BAY(m)" + CR + LF
Print_Text 0, CR + LF
For I= 1 To SCol(CProj)
For J = 1 To ColEle(I)
LCol = EleColRow(CProj, 1, I, J).Start
RCol = EleColRow(CProj, 1, I, J).End
Length = EleColRow(CProj, 1, I, J).Length
LBay = EleColRow(CProj, 1, I, J).L1
RBay = EleColRow(CProj, 1, I, J).L2
Print_Text 3, Format$(I, "###") + Format$(J, "#####") + Format$(
LCol, "#####") + Format$(RCol, "#####") + s5 + Format$(Length, "
0.00E+0") + s5 + Format$(LBay, "0.00E+0") + s5 + s2 + Format$(RBay, "0.00E
+0") + CR + LF
Next J
Next I

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUTF.BAS - 6

```

Print_Text 0, "=====
===== " + CR + LF
Print_Text 0, CR + LF
Print_Text 0, CR + LF
Print_Text 0, CR + LF
Dim SSystem As String * 50
Select Case GeneralData(CProj).SlabIndex
Case 0:
    SSystem = "Flat Plate"
Case 1:
    SSystem = "Flat Slab With Drop Panel"
Case 2:
    SSystem = "Flat Slab With Column Capital"
Case 3:
    SSystem = "Flat Slab With Column Drop Panel And Column Capital"
End Select
Print_Text 0, "SLAB SYSTEM : " + SSystem + CR + LF
Print_Text 0, "Overall Slab Thickness = " + Format$(100 * Thick(CProj)
"0.00") + ucm + CR + LF
Print_Text 0, "Cover of concrete = " + Format$(100 * (Thick(CProj)
- d(CProj)), "0.00") + ucm + CR + LF
Print_Text 0, CR + LF
Print_Text 0, "Load factor Wu = " + Format$(DeadCo, "0.0") + "DL
+ " + Format$(LiveCo, "0.0") + "LL" + CR + LF
Print_Text 0, "Unfactored Dead Load = " + Format$(DeadLoad(CProj), "0.
00") + uksm + CR + LF
Print_Text 0, "Unfactored Live Load = " + Format$(LiveLoad(CProj), "0.
00") + uksm + CR + LF

Print_Text 0, CR + LF
Print_Text 0, CR + LF
Print_Text 0, "SHEAR CHECKING AT COLUMN,DROP PANEL,COLUMN CAPITAL AND
BEAM ACTION SHEAR" + CR + LF
Print_Text 0, "=====
===== " + CR + LF
Print_Text 0, CR + LF
Print_Text 0, "COLUMN" + s10 + s10 + s5 + "PUNCHING SHEAR" + s10 + s10
+ s9 + "BEAM-ACTION-SHEAR" + CR + LF
Print_Text 0, s2 + "NO." + s9 + "COLUMN" + s3 + s10 + "DROP-PANEL" + s
1 + s10 + "COLUMN-CAPITAL" + CR + LF
Print_Text 0, s8 + "Vc(kg)" + s5 + "Vn(kg)" + s5 + "Vc(kg)" + s5 + "Vn
(kg)" + s5 + "Vc(kg)" + s5 + "Vn(kg)" + s5 + "Vc(kg)" + s5 + "Vn(kg)" + CR
+ LF
Print_Text 0, CR + LF
RNum = 2
RIden = 2
Dim ColumnNo As Integer
ColumnNo = 0
For I = 1 To SRow(CProj)
    For J = 1 To RowEle(I)
        ColumnNo = ColumnNo + 1
        CNo = EleColRow(CProj, 0, I, J).Start
        If Column(CProj, CNo).SupportType <> 2 Then 'If Not Free,Mean That i
s Column support
            RIden = RNum + ((CNo - 1) * 8)
            Get #20, RIden + 1, Vc0
            Get #20, RIden + 2, Vn0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUTF.BAS - 7

```

Get #20, RIden + 3, Vc1
Get #20, RIden + 4, Vn1
Get #20, RIden + 5, Vc2
Get #20, RIden + 6, Vn2
Get #20, RIden + 7, Vc3
Get #20, RIden + 8, Vn3
Print Text 4, Format$(ColumnNo, "###") + s3 + Format$(Vc0, "0.00E+
00") + s3 + Format$(Vn0, "0.00E+00") + s3 + Format$(Vc1, "0.00E+00") + s3
+ Format$(Vn1, "0.00E+00") + s3 + Format$(Vc2, "0.00E+00") + s3 + Format$(
Vn2, "0.00E+00") + s3 + Format$(Vc3, "0.00E+00") + s3 + Format$(Vn3, "0.00
E+00") + CR + LF
End If
If J = RowEle(I) Then
ColumnNo = ColumnNo + 1
CNo = EleColRow(CProj, 0, I, J).End
If Column(CProj, CNo).SupportType <> 2 Then
RIden = RNum + ((CNo - 1) * 8)
Get #20, RIden + 1, Vc0
Get #20, RIden + 2, Vn0
Get #20, RIden + 3, Vc1
Get #20, RIden + 4, Vn1
Get #20, RIden + 5, Vc2
Get #20, RIden + 6, Vn2
Get #20, RIden + 7, Vc3
Get #20, RIden + 8, Vn3
Print Text 4, Format$(ColumnNo, "###") + s3 + Format$(Vc0, "0.00E
+00") + s3 + Format$(Vn0, "0.00E+00") + s3 + Format$(Vc1, "0.00E+00") + s3
+ Format$(Vn1, "0.00E+00") + s3 + Format$(Vc2, "0.00E+00") + s3 + Format$(
Vn2, "0.00E+00") + s3 + Format$(Vc3, "0.00E+00") + s3 + Format$(Vn3, "0.0
0E+00") + CR + LF
End If
End If
Next J
Next I
Print_Text 0, "-----" + CR + LF
=====
Print_Text 0, CR + LF
Print_Text 0, CR + LF
Print_Text 0, "-----"
==" + CR + LF
Print_Text 0, "REINFORCEMENT OF SLAB IN COLUMN STRIP & HALF MIDDLE STR
IP" + CR + LF
Print_Text 0, "    DESIGN MOMENT AT CRITICAL SECTIONS FROM SUPPORTS
" + CR + LF
Print_Text 0, "-----"
==" + CR + LF
Print_Text 0, CR + LF
Print_Text 0, CR + LF
Print_Text 0, CR + LF
Print_Text 0, "REINFORCEMENT FOR DESIGN STRIP IN ROW" + CR + LF
Print_Text 0, "-----" + CR + LF
Print_Text 0, CR + LF
Print_Text 0, "ROW" + s3 + "ELEMENT" + s10 + s1 + "SECTION" + s10 + s3
+ "DESIGN M(kg-m)" + s3 + "As(cm^2)" + CR + LF
Print_Text 0, CR + LF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUTF.BAS - 8

```

Print Text 0, CR + LF
''' List Design moment
Dim RNum1 As Integer
Dim RNum2 As Integer
RNum1 = 2 + 8 * SNodeNum(CProj) + 6 * ElementXNum(CProj) + 6 * ElementYNum(CProj)
RNum2 = 2 + 8 * SNodeNum(CProj) + 6 * ElementXNum(CProj) + 6 * ElementYNum(CProj) + 9 * ElementXNum(CProj) + 9 * ElementYNum(CProj)
For I = 1 To SRow(CProj)
For J = 1 To RowEle(I)
Get #20, RNum1 + 1, MCL
Get #20, RNum1 + 2, MCD
Get #20, RNum1 + 3, MCR
Get #20, RNum1 + 4, MML1
Get #20, RNum1 + 5, MMD1
Get #20, RNum1 + 6, MMR1
Get #20, RNum1 + 7, MML2
Get #20, RNum1 + 8, MMD2
Get #20, RNum1 + 9, MMR2
RNum1 = RNum1 + 9
'Column strip
Get #20, RNum2 + 1, CLB
Get #20, RNum2 + 2, CMB
Get #20, RNum2 + 3, CRB
Get #20, RNum2 + 4, CLT
Get #20, RNum2 + 5, CMT
Get #20, RNum2 + 6, CRT
'Half middle strip1
Get #20, RNum2 + 7, MLB1
Get #20, RNum2 + 8, MMB1
Get #20, RNum2 + 9, MRB1
Get #20, RNum2 + 10, MLT1
Get #20, RNum2 + 11, MMT1
Get #20, RNum2 + 12, MRT1
'Half middle strip 2
Get #20, RNum2 + 13, MLB2
Get #20, RNum2 + 14, MMB2
Get #20, RNum2 + 15, MRB2
Get #20, RNum2 + 16, MLT2
Get #20, RNum2 + 17, MMT2
Get #20, RNum2 + 18, MRT2
RNum2 = RNum2 + 18
Print Text 8, Format$(I, "@@") + Format$(J, "@@@@@@") + "      Lef
Column Strip Top " + s3 + Format$(MCL, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(CLT, "0.00E+00") + CR + LF
Print Text 8, Format$(I, "@@") + Format$(J, "@@@@@@") + "
Bottom" + s3 + Format$(MCL, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(CLB, "0.00E+00") + CR + LF
Print Text 8, Format$(I, "@@") + Format$(J, "@@@@@@") + "
Top Bay Top " + s3 + Format$(MML1, " 0.0000E+00;-0.0000E+00")
s5 + Format$(MLT1, "0.00E+00") + CR + LF
Print Text 8, Format$(I, "@@") + Format$(J, "@@@@@@") + "
Bottom" + s3 + Format$(MML1, " 0.0000E+00;-0.0000E+00")
s5 + Format$(MLB1, "0.00E+00") + CR + LF
Print Text 8, Format$(I, "@@") + Format$(J, "@@@@@@") + "
Bottom Bay Top " + s3 + Format$(MML2, " 0.0000E+00;-0.0000E+00")

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUTF.BAS - 9

```

+ s5 + Format$(MLT2, "0.00E+00") + CR + LF
  Print_Text 8, Format$(I, "@@" + Format$(J, "#####") + "
    Bottom" + s3 + Format$(MML2, " 0.0000E+00;-0.0000E+00")
+ s5 + Format$(MLB2, "0.00E+00") + CR + LF
  Print_Text 8, Format$(I, "@@" + Format$(J, "#####") + "      Mid
dle Column Strip Top " + s3 + Format$(MCD, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(CMT, "0.00E+00") + CR + LF
  Print_Text 8, Format$(I, "@@" + Format$(J, "#####") + "
    Bottom" + s3 + Format$(MCD, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(CMB, "0.00E+00") + CR + LF
  Print_Text 8, Format$(I, "@@" + Format$(J, "#####") + "
Top Bay Top " + s3 + Format$(MMD1, " 0.0000E+00;-0.0000E+00")
+ s5 + Format$(MMT1, "0.00E+00") + CR + LF
  Print_Text 8, Format$(I, "@@" + Format$(J, "#####") + "
    Bottom" + s3 + Format$(MMD1, " 0.0000E+00;-0.0000E+00")
+ s5 + Format$(MMB1, "0.00E+00") + CR + LF
  Print_Text 8, Format$(I, "@@" + Format$(J, "#####") + "
Bottom Bay Top " + s3 + Format$(MMD2, " 0.0000E+00;-0.0000E+00")
+ s5 + Format$(MMT2, "0.00E+00") + CR + LF
  Print_Text 8, Format$(I, "@@" + Format$(J, "#####") + "
    Bottom" + s3 + Format$(MMD2, " 0.0000E+00;-0.0000E+00")
+ s5 + Format$(MMB2, "0.00E+00") + CR + LF
  Print_Text 8, Format$(I, "@@" + Format$(J, "#####") + "      Rig
ht Column Strip Top " + s3 + Format$(MCR, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(CRT, "0.00E+00") + CR + LF
  Print_Text 0, Format$(I, "@@" + Format$(J, "#####") + "
    Bottom" + s3 + Format$(MCR, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(CRB, "0.00E+00") + CR + LF
  Print_Text 8, Format$(I, "@@" + Format$(J, "#####") + "
Top Bay Top " + s3 + Format$(MMR1, " 0.0000E+00;-0.0000E+00")
+ s5 + Format$(MRT1, "0.00E+00") + CR + LF
  Print_Text 8, Format$(I, "@@" + Format$(J, "#####") + "
    Bottom" + s3 + Format$(MMR1, " 0.0000E+00;-0.0000E+00")
+ s5 + Format$(MRB1, "0.00E+00") + CR + LF
  Print_Text 8, Format$(I, "@@" + Format$(J, "#####") + "
Bottom Bay Top " + s3 + Format$(MMR2, " 0.0000E+00;-0.0000E+00")
+ s5 + Format$(MRT2, "0.00E+00") + CR + LF
  Print_Text 8, Format$(I, "@@" + Format$(J, "#####") + "
    Bottom" + s3 + Format$(MMR2, " 0.0000E+00;-0.0000E+00")
+ s5 + Format$(MRB2, "0.00E+00") + CR + LF
  Print_Text 8, "=====
===== " + CR + LF
  Next J
Next I
Print_Text 0, CR + LF
Print_Text 0, CR + LF
Print_Text 0, "REINFORCEMENT FOR DESIGN STRIP IN COLUMN" + CR + LF
Print_Text 0, "===== " + CR + LF
Print_Text 0, CR + LF
Print_Text 0, "COLUMN" + s1 + "ELEMENT" + s10 + s1 + "SECTION" + s10
+ s3 + "DESIGN M(kg-m)" + s3 + "As(cm^2)" + CR + LF
Print_Text 0, CR + LF
RNum1 = 2 + 8 * SNodeNum(CProj) + 6 * ElementXNum(CProj) + 6 * Elemen
tYNum(CProj) + 9 * ElementXNum(CProj)
RNum2 = 2 + 8 * SNodeNum(CProj) + 6 * ElementXNum(CProj) + 6 * Elemen
tYNum(CProj) + 9 * ElementXNum(CProj) + 9 * ElementYNum(CProj) + 18 * Elem
entXNum(CProj)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUTF.BAS - 10

```

For I = 1 To SCol(CProj)
  For J = 1 To ColEle(I)
    Get #20, RNum1 + 1, MCL
    Get #20, RNum1 + 2, MCD
    Get #20, RNum1 + 3, MCR
    Get #20, RNum1 + 4, MML1
    Get #20, RNum1 + 5, MMD1
    Get #20, RNum1 + 6, MMR1
    Get #20, RNum1 + 7, MML2
    Get #20, RNum1 + 8, MMD2
    Get #20, RNum1 + 9, MMR2
    RNum1 = RNum1 + 9
    'Column strip
    Get #20, RNum2 + 1, CLB
    Get #20, RNum2 + 2, CMB
    Get #20, RNum2 + 3, CRB
    Get #20, RNum2 + 4, CLT
    Get #20, RNum2 + 5, CMT
    Get #20, RNum2 + 6, CRT
    'Half middle strip1
    Get #20, RNum2 + 7, MLB1
    Get #20, RNum2 + 8, MMB1
    Get #20, RNum2 + 9, MRB1
    Get #20, RNum2 + 10, MLT1
    Get #20, RNum2 + 11, MMT1
    Get #20, RNum2 + 12, MRT1
    'Half middle strip 2
    Get #20, RNum2 + 13, MLB2
    Get #20, RNum2 + 14, MMB2
    Get #20, RNum2 + 15, MRB2
    Get #20, RNum2 + 16, MLT2
    Get #20, RNum2 + 17, MMT2
    Get #20, RNum2 + 18, MRT2
    RNum2 = RNum2 + 18
    Print Text 9, Format$(I, "@@") + Format$(J, "#####") + "      Top
Column Strip Top " + s3 + Format$(MCL, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(CLT, "0.00E+00") + CR + LF
    Print Text 9, Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MCL, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(CLB, "0.00E+00") + CR + LF
    Print Text 9, Format$(I, "@@") + Format$(J, "#####") + "
Left Bay Top " + s3 + Format$(MML1, " 0.0000E+00;-0.0000E+00")
+ s5 + Format$(MLT1, "0.00E+00") + CR + LF
    Print Text 9, Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MML1, " 0.0000E+00;-0.0000E+00")
+ s5 + Format$(MLB1, "0.00E+00") + CR + LF
    Print Text 9, Format$(I, "@@") + Format$(J, "#####") + "
Right Bay Top " + s3 + Format$(MML2, " 0.0000E+00;-0.0000E+00")
+ s5 + Format$(MLT2, "0.00E+00") + CR + LF
    Print Text 9, Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MML2, " 0.0000E+00;-0.0000E+00")
+ s5 + Format$(MLB2, "0.00E+00") + CR + LF
    Print Text 9, Format$(I, "@@") + Format$(J, "#####") + "      Mid
dle Column Strip Top " + s3 + Format$(MCD, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(CMT, "0.00E+00") + CR + LF
    Print Text 9, Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MCD, " 0.0000E+00;-0.0000E+00") +

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าการณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUTF.BAS - 11

```

s5 + Format$(CMB, "0.00E+00") + CR + LF
  Print_Text 9, Format$(I, "@@") + Format$(J, "#####") + "
Left Bay Top " + s3 + Format$(MMD1, " 0.0000E+00;-0.0000E+00")
+ s5 + Format$(MMT1, "0.00E+00") + CR + LF
  Print_Text 9, Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MMD1, " 0.0000E+00;-0.0000E+00")
+ s5 + Format$(MMB1, "0.00E+00") + CR + LF
  Print_Text 9, Format$(I, "@@") + Format$(J, "#####") + "
Right Bay Top " + s3 + Format$(MMD2, " 0.0000E+00;-0.0000E+00")
+ s5 + Format$(MMT2, "0.00E+00") + CR + LF
  Print_Text 9, Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MMD2, " 0.0000E+00;-0.0000E+00")
+ s5 + Format$(MMB2, "0.00E+00") + CR + LF
  Print_Text 9, Format$(I, "@@") + Format$(J, "#####") + " Bot
tom Column Strip Top " + s3 + Format$(MCR, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(CRT, "0.00E+00") + CR + LF
  Print_Text 9, Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MCR, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(CRB, "0.00E+00") + CR + LF
  Print_Text 9, Format$(I, "@@") + Format$(J, "#####") + "
Left Bay Top " + s3 + Format$(MMR1, " 0.0000E+00;-0.0000E+00")
+ s5 + Format$(MRT1, "0.00E+00") + CR + LF
  Print_Text 9, Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MMR1, " 0.0000E+00;-0.0000E+00")
+ s5 + Format$(MRB1, "0.00E+00") + CR + LF
  Print_Text 9, Format$(I, "@@") + Format$(J, "#####") + "
Right Bay Top " + s3 + Format$(MMR2, " 0.0000E+00;-0.0000E+00")
+ s5 + Format$(MRT2, "0.00E+00") + CR + LF
  Print_Text 9, Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MMR2, " 0.0000E+00;-0.0000E+00")
+ s5 + Format$(MRB2, "0.00E+00") + CR + LF
  Print_Text 9, "=====
===== " + CR + LF
  Next J
Next I
If GeneralData(CProj).SlabIndex <> 0 Then
CNo = 0
ColumnNo = 0
Print_Text 0, CR + LF
Print_Text 0, "=====
===== " + CR + LF
Print_Text 0, "
DIMENSION OF DROP PANEL & COLUMN CAPITAL
" + CR + LF
Print_Text 0, "
" + CR + LF
Print_Text 0, "=====
===== " + CR + LF
Print_Text 0, CR + LF
Print_Text 0, CR + LF
Print_Text 0, s2 + "COLUMN" + s2 + "DROP WIDTH(m)" + s2 + "DROP LENGTH
(m)" + s2 + "DROP THICK(m)" + s2 + "CAPITAL WIDTH(m)" + s2 + "CAPITAL LENG
TH(m)" + CR + LF
Print_Text 0, CR + LF
For I = 1 To SRow(CProj)
For J = 1 To RowEle(I)
CNo = CNo + 1
ColumnNo = EleColRow(CProj, 0, I, J).Start

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUTF.BAS - 12

```

DropX = Column(CProj, ColumnNo).DropX
DropY = Column(CProj, ColumnNo).DropY
DropThick = Column(CProj, ColumnNo).DropThick
CapitalX = Column(CProj, ColumnNo).CapitalX
CapitalY = Column(CProj, ColumnNo).CapitalY
Print_Text 5, Format$(CNo, "#####") + s6 + Format$(DropX, "0.00E+00
") + s7 + Format$(DropY, "0.00E+00") + s8 + Format$(DropThick, "0.00E+00")
+ s9 + Format$(CapitalX, "0.00E+00") + s10 + Format$(CapitalY, "0.00E+00"
) + CR + LF
If J = RowEle(I) Then
  CNo = CNo + 1
  ColumnNo = EleColRow(CProj, 0, I, RowEle(I)).End
  DropX = Column(CProj, ColumnNo).DropX
  DropY = Column(CProj, ColumnNo).DropY
  DropThick = Column(CProj, ColumnNo).DropThick
  CapitalX = Column(CProj, ColumnNo).CapitalX
  CapitalY = Column(CProj, ColumnNo).CapitalY
  Print_Text 5, Format$(CNo, "#####") + s6 + Format$(DropX, "0.00E+0
0") + s7 + Format$(DropY, "0.00E+00") + s8 + Format$(DropThick, "0.00E+00"
) + s9 + Format$(CapitalX, "0.00E+00") + s10 + Format$(CapitalY, "0.00E+00
") + CR + LF
End If
Next J
Next I
Print_Text 0, "-----" + CR + LF
ElseIf GeneralData(CProj).SlabIndex = 0 Then
End If
Printer.EndDoc
Close #20
Exit Sub
DlgErrorPrintFlatListCompute: Exit Sub
End Sub

Sub Screen_FlatListCompute ()
Dim XMax As Integer
Dim XMin As Integer
Dim YMax As Integer
Dim YMin As Integer
Dim InitX As Integer
Dim InitY As Integer
Dim Col As Integer
Dim CStart As Integer
Dim CEnd As Integer
Dim TPMX As Single
Dim TPMY As Single
Dim TPP As Single
Dim TPM As Single
Dim AddSect As Single
Dim X1 As Single
Dim Y1 As Single
Dim X2 As Single
Dim Y2 As Single
Dim X As Single
Dim Y As Single
Dim I As Integer

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUTF.BAS - 13

```

Dim J As Integer

Dim Init1 As Integer
Dim Init2 As Integer
Dim PF As Single
Dim M As Single
Dim Wu As Single
Dim ColumnColor As Integer
Dim ElementColor As Integer
Dim ProfileColor As Integer
Dim CR As String * 1
Dim LF As String * 1
CR = Chr$(13)
LF = Chr$(10)
Const ucm = " cm"
Const um = " m"
Const ukg = " kg"
Const uksc = " ksc"
Const uksm = " kg/m^2"
Dim CNo As Integer
Dim RNum As Integer

'shear
'Column
Dim Vc0 As Single
Dim Vn0 As Single
'Drop
Dim Vc1 As Single
Dim Vn1 As Single
'Capital

Dim Vc2 As Single
Dim Vn2 As Single
'Diagonal
Dim Vc3 As Single
Dim Vn3 As Single

'''M design
Dim MDL As Single
Dim MDD As Single
Dim MDR As Single
'M design at column strip
Dim MCL As Single
Dim MCD As Single
Dim MCR As Single
'M design at middle strip
Dim MML1 As Single
Dim MMD1 As Single
Dim MMR1 As Single
Dim MML2 As Single
Dim MMD2 As Single
Dim MMR2 As Single
'''Area of reinforce steel
'Half middle strip 1
Dim MLB1 As Single
Dim MMB1 As Single
Dim MRB1 As Single

```



OUTPUTF.BAS - 14

```

Dim MLT1 As Single
Dim MMT1 As Single
Dim MRT1 As Single
'Half middle strip 2
Dim MLB2 As Single
Dim MMB2 As Single
Dim MRB2 As Single
Dim MLT2 As Single
Dim MMT2 As Single
Dim MRT2 As Single
'column strip
Dim CLB As Single
Dim CMB As Single
Dim CRB As Single
Dim CLT As Single
Dim CMT As Single
Dim CRT As Single

'''Reinforcement design
Dim RMDL As Single
Dim RMDD As Single
Dim RMDR As Single
'M design at column strip
Dim RMCL As Single
Dim RMCD As Single
Dim RMCR As Single
'M design at middle strip
Dim RMLL1 As Single
Dim RMD1 As Single
Dim RMMR1 As Single
Dim RMLL2 As Single
Dim RMD2 As Single
Dim RMMR2 As Single
'''Area of reinforce steel
'Half middle strip 1
Dim RMLB1 As Single
Dim RMMB1 As Single
Dim RMRB1 As Single
Dim RMLT1 As Single
Dim RMMT1 As Single
Dim RMRT1 As Single
'Half middle strip 2
Dim RMLB2 As Single
Dim RMMB2 As Single
Dim RMRB2 As Single
Dim RMLT2 As Single
Dim RMMT2 As Single
Dim RMRT2 As Single
'column strip
Dim RCLB As Single
Dim RCMB As Single
Dim RCRB As Single
Dim RCLT As Single
Dim RCMT As Single
Dim RCRT As Single

Dim TextNo As Integer

```



OUTPUTF.BAS - 15

```
ReDim Text(100) As String
TextNo = 1
```

```
Open DirApp + "temp.swp" For Random Access Read Write Lock Read Write
As #20 Len = 4
```

```
''' Init
frmList(CProj).Cls
```

```
''' Information
```

```
Screen_Text CR + LF
```

```
Screen_Text "===== " + C
```

```
R + LF
Screen_Text "          TWO WAY FLAT PLATE SLAB DESIGN          " + C
```

```
R + LF
Screen_Text " KING MONGKUT INSTITUTE OF TECHNOLOGY LADKRABANG " + C
```

```
R + LF
Screen_Text "          CIVIL ENGINEERING DEPARTMENT          " + C
```

```
R + LF
Screen_Text "          BY SIAM KUNKATE & PATCHARAWALAI MAITA          " + C
```

```
R + LF
Screen_Text "          LISTING COMPUTATION CHECKING          " + C
```

```
R + LF
Screen_Text "===== " + C
```

```
R + LF
Screen_Text CR + LF
Screen_Text CR + LF
```

```
Screen_Text "Date : " + Format$(Now, "dddd mm'dd/yyyy ") + C
```

```
R + LF
Screen_Text "Time : " + Format$(Now, "hh:mm:ss AM/PM") + CR
```

```
+ LF
Screen_Text "File Name : " + SData(CProj).FileName + CR + LF
```

```
Screen_Text "Project : " + SData(CProj).ProjectData + CR + LF
```

```
Screen_Text "Engineer : " + SData(CProj).Engineer + CR + LF
```

```
Screen_Text "Address : " + SData(CProj).AddressData + CR + LF
```

```
Screen_Text "Information : " + SData(CProj).InformData + CR + LF
```

```
''' Data for Design
```

```
Screen_Text "Length Unit : Meter(m) " + CR + LF
```

```
Screen_Text "Load Unit : Kilogram(Kg) " + CR + LF
```

```
Screen_Text "fc' : " + Format$(GeneralData(CProj).SFC, ".0
```

```
0") + uksc + CR + LF
Screen_Text "fy (Reinforce) : " + Format$(GeneralData(CProj).RFs, ".0
```

```
0") + uksc + CR + LF
Screen_Text "Ec (Slab) : " + Format$(GeneralData(CProj).SEc, ".00
```

```
") + uksc + CR + LF
Screen_Text "Ec (Column) : " + Format$(GeneralData(CProj).CEc, ".00
```

```
0") + uksc + CR + LF
```

```
Screen_Text "Ec (Column) : " + Format$(GeneralData(CProj).CEc, ".00
```

```
0") + uksc + CR + LF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

) + uksc + CR + LF

```

'''' Column Data
frmList(CProj).FontSize = 7
frmList(CProj).FontBold = True
Screen_Text CR + LF
Screen_Text CR + LF

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUTF.BAS - 16

```

Screen_Text "COLUMN DATA" + CR + LF
Screen_Text "======" + CR + LF
Screen_Text CR + LF
Screen_Text "COLUMN NO.  CO_X(m)      CO_Y(m)      LENGTH(m)      " + CR +
LF
Screen_Text CR + LF
CNo = 0
For I = 1 To SRow(CProj)
  For J = 1 To RowEle(I)
    CNo = CNo + 1
    Co_X = Column(CProj, EleColRow(CProj, 0, I, J).Start).Co_X
    Co_Y = Column(CProj, EleColRow(CProj, 0, I, J).Start).Co_Y
    Support = Column(CProj, EleColRow(CProj, 0, I, J).Start).SupportType
    Length = Column(CProj, EleColRow(CProj, 0, I, J).Start).Length
    Screen_Text Format$(CNo, "@@@@@@") + s5 + Format$(Co_X, "0.00E+0")
+ s5 + Format$(Co_Y, "0.00E+0") + s5 + Format$(Length, "0.00E+0") + CR + L
F
    If J = RowEle(I) Then
      CNo = CNo + 1
      Co_X = Column(CProj, EleColRow(CProj, 0, I, RowEle(I)).End).Co_X
      Co_Y = Column(CProj, EleColRow(CProj, 0, I, RowEle(I)).End).Co_Y
      Support = Column(CProj, EleColRow(CProj, 0, I, RowEle(I)).End).Supp
ortType
      Length = Column(CProj, EleColRow(CProj, 0, I, RowEle(I)).End).Lengt
h
      Screen_Text Format$(CNo, "@@@@@@") + s5 + Format$(Co_X, "0.00E+0")
+ s5 + Format$(Co_Y, "0.00E+0") + s5 + Format$(Length, "0.00E+0") + CR +
LF
    End If
  Next J
Next I
Screen_Text "======" + CR + LF
''' Element Data
Dim LCol As Integer
Dim RCol As Integer
Dim LBay As Single
Dim RBay As Single
'Row
Screen_Text CR + LF
Screen_Text CR + LF
Screen_Text "DESIGN STRIP IN ROW" + CR + LF
Screen_Text "======" + CR + LF
Screen_Text CR + LF
Screen_Text " ROW      ELEMENT      START-COL  END-COL  LENGTH(m)      TOP-BAY(
m) BOTTOM-BAY(m)" + CR + LF
Screen_Text CR + LF
For I = 1 To SRow(CProj)
  For J = 1 To RowEle(I)
    LCol = EleColRow(CProj, 0, I, J).Start
    RCol = EleColRow(CProj, 0, I, J).End

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Length = EleColRow(CProj, 0, I, J).Length
LBay = EleColRow(CProj, 0, I, J).L1
RBay = EleColRow(CProj, 0, I, J).L2
Screen Text Format$(I, "###") + Format$(J, "#####") + Format$(LC
ol, "#####") + Format$(RCol, "#####") + s5 + Format$(Length, "0.
00E+0") + s5 + Format$(LBay, "0.00E+0") + s5 + s2 + Format$(RBay, "0.00E+0
") + CR + LF

```



```

Next J
Next I

Screen_Text "=====
===== " + CR + LF
'Column
Screen_Text CR + LF
Screen_Text CR + LF
Screen_Text "DESIGN STRIP IN COLUMN" + CR + LF
Screen_Text "===== " + CR + LF
Screen_Text CR + LF
Screen_Text "COLUMN ELEMENT    START-COL  END-COL  LENGTH(m)  TOP-BAY(
m) BOTTOM-BAY(m)" + CR + LF
Screen_Text CR + LF
For I = 1 To SCOL(CProj)
  For J = 1 To Colele(I)
    LCol = EleColRow(CProj, 1, I, J).Start
    RCol = EleColRow(CProj, 1, I, J).End
    Length = EleColRow(CProj, 1, I, J).Length
    LBay = EleColRow(CProj, 1, I, J).L1
    RBay = EleColRow(CProj, 1, I, J).L2
    Screen_Text Format$(I, "#####") + Format$(J, "#####") + Format$(LCo
ol, "#####") + Format$(RCol, "#####") + s5 + Format$(Length, "0.
00E+0") + s5 + Format$(LBay, "0.00E+0") + s5 + s2 + Format$(RBay, "0.00E+0
") + CR + LF
  Next J
Next I
Screen_Text "=====
===== " + CR + LF
Screen_Text CR + LF
Screen_Text CR + LF
Screen_Text CR + LF
Dim SSystem As String * 50
Select Case GeneralData(CProj).SlabIndex
Case 0:
  SSystem = "Flat Plate"
Case 1:
  SSystem = "Flat Slab With Drop Panel"
Case 2:
  SSystem = "Flat Slab With Column Capital"
Case 3:
  SSystem = "Flat Slab With Column Drop Panel And Column Capital"
End Select
Screen_Text "SLAB SYSTEM    : " + SSystem + CR + LF
Screen_Text "Overall Slab Thickness = " + Format$(100 * Thick(CProj),
"0.00") + ucm + CR + LF
Screen_Text "Cover of concrete    = " + Format$(100 * (Thick(CProj)
- d(CProj)), "0.00") + ucm + CR + LF
Screen_Text CR + LF
Screen_Text "Load factor          Wu = " + Format$(DeadCo, "0.0") + "DL+"
+ Format$(LiveCo, "0.0") + "LL" + CR + LF
Screen_Text "Unfactored Dead Load = " + Format$(DeadLoad(CProj), "0.00
") + uksm + CR + LF
Screen_Text "Unfactored Live Load = " + Format$(LiveLoad(CProj), "0.00
") + uksm + CR + LF

Screen_Text CR + LF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

End If
End If
Next J
Next I
Screen_Text "=====
===== " + CR + LF

Screen_Text CR + LF
Screen_Text CR + LF
Screen_Text "=====
" + CR + LF
Screen_Text "REINFORCEMENT OF SLAB IN COLUMN STRIP & HALF MIDDLE STRIP
" + CR + LF
Screen_Text "    DESIGN MOMENT AT CRITICAL SECTIONS FROM SUPPORTS
" + CR + LF
Screen_Text "=====
" + CR + LF
Screen_Text CR + LF
Screen_Text CR + LF

Screen_Text CR + LF
Screen_Text "REINFORCEMENT FOR DESIGN STRIP IN ROW" + CR + LF
Screen_Text "===== " + CR + LF
Screen_Text CR + LF
Screen_Text "ROW" + s3 + "ELEMENT" + s10 + s1 + "SECTION" + s10 + s3 +
"DESIGN M(kg-m)" + s3 + "As(cm^2)" + CR + LF
Screen_Text CR + LF

Screen_Text CR + LF
''' List Design moment
Dim RNum1 As Integer
Dim RNum2 As Integer
RNum1 = 2 + 8 * SNodeNum(CProj) + 6 * ElementXNum(CProj) + 6 * Elemen
tYNum(CProj)
RNum2 = 2 + 8 * SNodeNum(CProj) + 6 * ElementXNum(CProj) + 6 * Elemen
tYNum(CProj) + 9 * ElementXNum(CProj) + 9 * ElementYNum(CProj)
For I = 1 To SRow(CProj)
For J = 1 To RowEle(I)
Get #20, RNum1 + 1, MCL
Get #20, RNum1 + 2, MCD
Get #20, RNum1 + 3, MCR
Get #20, RNum1 + 4, MML1
Get #20, RNum1 + 5, MMD1
Get #20, RNum1 + 6, MMR1
Get #20, RNum1 + 7, MML2
Get #20, RNum1 + 8, MMD2
Get #20, RNum1 + 9, MMR2
RNum1 = RNum1 + 9
'Column strip
Get #20, RNum2 + 1, CLB
Get #20, RNum2 + 2, CMB
Get #20, RNum2 + 3, CRB
Get #20, RNum2 + 4, CLT
Get #20, RNum2 + 5, CMT
Get #20, RNum2 + 6, CRT
'Half middle strip1
Get #20, RNum2 + 7, MLB1

```

```

Get #20, RNum2 + 8, MMB1
Get #20, RNum2 + 9, MRB1
Get #20, RNum2 + 10, MLT1
Get #20, RNum2 + 11, MMT1
Get #20, RNum2 + 12, MRT1
'Half middle strip 2
Get #20, RNum2 + 13, MLB2
Get #20, RNum2 + 14, MMB2
Get #20, RNum2 + 15, MRB2
Get #20, RNum2 + 16, MLT2
Get #20, RNum2 + 17, MMT2
Get #20, RNum2 + 18, MRT2
RNum2 = RNum2 + 18
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "      Left
Column Strip Top " + s3 + Format$(MCL, " 0.0000E+00;-0.0000E+00") + s
5 + Format$(CLT, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MCL, " 0.0000E+00;-0.0000E+00") + s
5 + Format$(CLB, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Top Bay Top " + s3 + Format$(MML1, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(MLT1, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MML1, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(MLB1, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Bottom Bay Top " + s3 + Format$(MML2, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(MLT2, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MML2, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(MLB2, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "      Middl
e Column Strip Top " + s3 + Format$(MCD, " 0.0000E+00;-0.0000E+00") + s
5 + Format$(CMT, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MCD, " 0.0000E+00;-0.0000E+00") + s
5 + Format$(CMB, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Top Bay Top " + s3 + Format$(MMD1, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(MMT1, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MMD1, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(MMB1, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Bottom Bay Top " + s3 + Format$(MMD2, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(MMT2, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MMD2, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(MMB2, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "      Right
Column Strip Top " + s3 + Format$(MCR, " 0.0000E+00;-0.0000E+00") + s
5 + Format$(CRT, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MCR, " 0.0000E+00;-0.0000E+00") + s
5 + Format$(CRB, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Top Bay Top " + s3 + Format$(MMR1, " 0.0000E+00;-0.0000E+00") +

```



```

Get #20, RNum2 + 16, MLT2
Get #20, RNum2 + 17, MMT2
Get #20, RNum2 + 18, MRT2
RNum2 = RNum2 + 18
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "      Top
Column Strip Top      " + s3 + Format$(MCL, " 0.0000E+00;-0.0000E+00") + s
5 + Format$(CLT, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MCL, " 0.0000E+00;-0.0000E+00") + s
5 + Format$(CLB, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Left Bay Top      " + s3 + Format$(MML1, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(MLT1, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MML1, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(MLB1, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Right Bay Top     " + s3 + Format$(MML2, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(MLT2, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MML2, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(MLB2, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "      Middl
e Column Strip Top  " + s3 + Format$(MCD, " 0.0000E+00;-0.0000E+00") + s
5 + Format$(CMT, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MCD, " 0.0000E+00;-0.0000E+00") + s
5 + Format$(CMB, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Left Bay Top      " + s3 + Format$(MMD1, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(MMT1, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MMD1, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(MMB1, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Right Bay Top     " + s3 + Format$(MMD2, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(MMT2, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MMD2, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(MMB2, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "      Botto
m Column Strip Top  " + s3 + Format$(MCR, " 0.0000E+00;-0.0000E+00") + s
5 + Format$(CRT, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MCR, " 0.0000E+00;-0.0000E+00") + s
5 + Format$(CRB, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Left Bay Top      " + s3 + Format$(MMR1, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(MRT1, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MMR1, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(MRB1, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Right Bay Top     " + s3 + Format$(MMR2, " 0.0000E+00;-0.0000E+00") +
s5 + Format$(MRT2, "0.00E+00") + CR + LF
Screen_Text Format$(I, "@@") + Format$(J, "#####") + "
Bottom" + s3 + Format$(MMR2, " 0.0000E+00;-0.0000E+00") +

```

```

s5 + Format$(MRB2, "0.00E+00") + CR + LF
  Screen_Text "=====
===== " + CR + LF
  Next J
  Next I
  If GeneralData(CProj).SlabIndex <> 0 Then
  CNo = 0
  ColumnNo = 0
  Screen_Text CR + LF
  Screen_Text "=====
===== " + CR + LF
  Screen_Text "
  " + CR + LF
  Screen_Text "
  " + CR + LF
  Screen_Text "=====
===== " + CR + LF
  Screen_Text CR + LF
  Screen_Text CR + LF
  Screen_Text 's2 + "COLUMN" + s2 + "DROP WIDTH(m)." + s2 + "DROP LENGTH(m
)" + s2 + "DROP THICK(m)" + s2 + "CAPITAL WIDTH(m)" + s2 + "CAPITAL LENGTH
(m)" + CR + LF
  Screen_Text CR + LF
  For I = 1 To SRow(CProj)
  For J = 1 To RowEle(I)
  CNo = CNo + 1
  ColumnNo = EleColRow(CProj, 0, I, J).Start
  DropX = Column(CProj, ColumnNo).DropX
  DropY = Column(CProj, ColumnNo).DropY
  DropThick = Column(CProj, ColumnNo).DropThick
  CapitalX = Column(CProj, ColumnNo).CapitalX
  CapitalY = Column(CProj, ColumnNo).CapitalY
  Screen_Text Format$(CNo, "#####") + s6 + Format$(DropX, "0.00E+00")
+ s7 + Format$(DropY, "0.00E+00") + s8 + Format$(DropThick, "0.00E+00") +
+ s9 + Format$(CapitalX, "0.00E+00") + s10 + Format$(CapitalY, "0.00E+00")
+ CR + LF
  If J = RowEle(I) Then
  CNo = CNo + 1
  ColumnNo = EleColRow(CProj, 0, I, RowEle(I)).End
  DropX = Column(CProj, ColumnNo).DropX
  DropY = Column(CProj, ColumnNo).DropY
  DropThick = Column(CProj, ColumnNo).DropThick
  CapitalX = Column(CProj, ColumnNo).CapitalX
  CapitalY = Column(CProj, ColumnNo).CapitalY
  Screen_Text Format$(CNo, "#####") + s6 + Format$(DropX, "0.00E+00"
) + s7 + Format$(DropY, "0.00E+00") + s8 + Format$(DropThick, "0.00E+00")
+ s9 + Format$(CapitalX, "0.00E+00") + s10 + Format$(CapitalY, "0.00E+00")
+ CR + LF
  End If
  Next J

```

```
Next I
Screen_Text "=====
===== " + CR + LF
ElseIf GeneralData(CProj).SlabIndex = 0 Then

End If
For I = 1 To 20
```



```
frmList(CProj).txtListCompute(I).Top = 0
frmList(CProj).txtListCompute(I).Left = 0
frmList(CProj).txtListCompute(I).Width = frmList(CProj).ScaleWidth
frmList(CProj).txtListCompute(I).Height = frmList(CProj).ScaleHeight
frmList(CProj).txtListCompute(I).Visible = False
Next I
frmList(CProj).txtListCompute(TextNum + 1).Text = TextToScreen
TextToScreen = ""
TextModule = TextNum + 1
TextNum = 0
CurrentPage = 1
frmList(CProj).txtListCompute(1).Visible = True
frmMain.cmdPrevious.Visible = True
frmMain.cmdNext.Visible = True
If TextModule = 1 Then
    frmMain.cmdPrevious.Enabled = False
    frmMain.cmdNext.Enabled = False
ElseIf TextModule > 1 Then
    frmMain.cmdPrevious.Enabled = False
    frmMain.cmdNext.Enabled = True
End If
Close #20
End Sub
```



บรรณานุกรม

1. “Requirements for Reinforced Concrete with Design Applications”,
Note on ACI 318-77 Building Code, Portland Cement Association,
Second Edition, part 21
2. Chu - Kai Wang and Charles G. Salmon , “Reinforced Concrete Design” ,
Fifth Edition, chapter 3 page 58-61 , chapter 16
3. Paul F. Rice and Edward S.Hoffman , “ Structural Desing Guide To The
ACI Building Code ” , Third Edition , chapter 5,6
4. GAME G. Magregor , “ Reinforced Concrete. ”, chapter 16,17
5. Arthur H. Nilson, George Winter , “Design Of Concrete Structures” ,
McGraw-Hill , Eleventh Edition , page 472-485
6. Kenneth Leet , “Reinforced Concrete Disign” , Second Edition,
Chapter 11 , page 509
7. ศิริวัฒน์ ไชยชนะ, “คอนกรีตเสริมเหล็ก ทฤษฎีกำลังประลัย”, ภาควิชาวิศวกรรม
โยธาสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง , พิมพ์ครั้งที่ 2 ,
พฤษภาคม 2531, บทที่ 10
8. อำนวย พานิชกุล, สมนึก กุลประภา, วินิต ช่อวิเชียร “การวิเคราะห์โครงสร้าง”
พิมพ์ครั้งที่ 6, พ.ศ. 2535 , หน้า 198
9. “มาตรฐานสำหรับอาคารคอนกรีตเสริมเหล็ก” , คณะกรรมการสาขาวิศวกรรมโยธา
2533-34, ตุลาคม 2534, พิมพ์ครั้งที่ 2 , พฤษภาคม 2536
10. สนั่น เจริญเผ่า และ วินิต ช่อวิเชียร , “คอนกรีตเสริมเหล็ก” , ฉบับแก้ไขเพิ่มเติม
พิมพ์ครั้งที่ 8, มิถุนายน 2537, บทที่ 10, 11