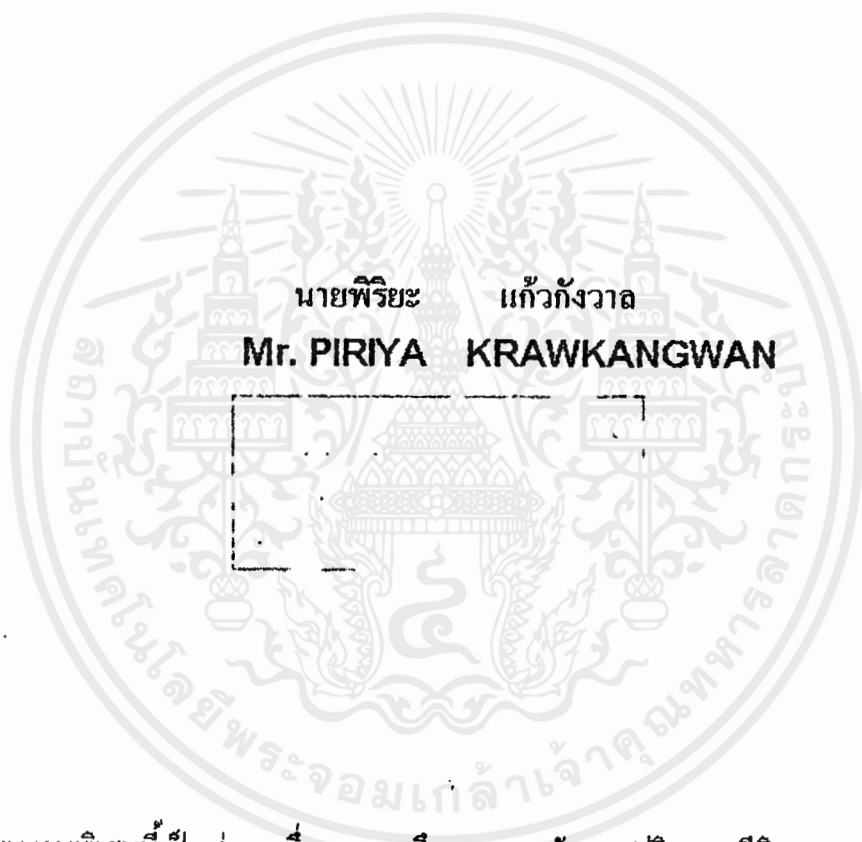




โปรแกรมคอมพิวเตอร์สำหรับการวางแผนงาน BAR CHART  
COMPUTER PROGRAMING FOR BAR CHART PLANNING



นายพิริยะ แก้วก้งวาด  
Mr. PIRIYA KRAWKANGWAN

โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาตรีวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการก่อสร้าง

ภาควิชาวิศวกรรมโยธา

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032568

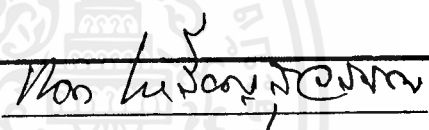
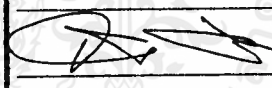
# COMPUTER PROGRAMING FOR BAR CHART PLANNING



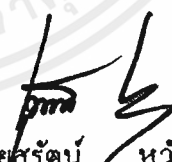
A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILMENT  
OF THE REQUIRMENTS FOR THE DEGREE  
BECHELOR OF CONSTRUCTION ENGINEERING  
DEPARTMENT OF CIVIL ENGINEERING  
KING MONGKUT 'S INSTITUE OF TECHNOLOGY LADKRABANG  
1991

ภาควิชาวิศวกรรมโยธา คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองโครงการพิเศษ

หัวข้อโครงการพิเศษ โปรแกรมคอมพิวเตอร์สำหรับการวางแผนงาน BAR CHART สำหรับงานก่อสร้าง  
นักศึกษา นายพิริยะ แก้วกั้งวาล รหัสประจำตัว 32.1215  
หลักสูตร วิศวกรรมศาสตรบัณฑิต สาขา วิศวกรรมก่อสร้าง  
ภาควิชา วิศวกรรมโยธา  
อาจารย์ที่ปรึกษา อ.วิบูลย์ วุฒินิยาม

คณะกรรมการสอบโครงการพิเศษ	ลายมือชื่อ
ผศ.ดร.แดง เจริญสุวรรณ	
อ. เกษม อมันตกุล	
อ. จักรพงษ์ พงษ์เพ็ง	

ภาควิชาโยธารับรองแล้ว

  
( นายสุรคน หวังเจริญ )

หัวหน้าภาควิชาวิศวกรรมโยธา

วันที่ เดือน พ.ศ. 2536

หัวข้อโครงการพิเศษ โปรแกรมคอมพิวเตอร์สำหรับการวางแผนงาน BAR CHART สำหรับงานก่อสร้าง  
 นักศึกษา นายพิริยะ แก้วกั้งवाल  
 อาจารย์ที่ปรึกษา อ.วิบูลย์ วุฒินุญ  
 ระดับการศึกษา วิศวกรรมศาสตรบัณฑิต สาขา วิศวกรรมก่อสร้าง  
 ภาควิชา วิศวกรรมโยธา คณะวิศวกรรมศาสตร์  
 พ.ศ. 2535

### บทคัดย่อ

ในปัจจุบันนี้การทำงานใด ๆ ก็ตามมักจะอยู่ภายใต้เงื่อนไขต่าง ๆ มากมาย เช่น เวลา งบประมาณ ทรัพยากร ต่าง ๆ ที่ต้องนำมาใช้ เช่น คน เครื่องจักร ฯลฯ จนบางครั้งสมองของคนเราไม่สามารถจะรวบรวมข้อมูลต่าง ๆ เหล่านี้ได้หมด เพื่อเป็นการแก้ปัญหาจึงมีการสร้างแผนงานต่าง ๆ ขึ้นหลายแบบ เช่น BAR CHART , Network Diagram, CPM ,PERT เพื่อช่วยให้เห็นภาพรวมคร่าว ๆ ได้ บางแบบก็สามารถติดตามผล ได้อย่างละเอียดถี่ถ้วนขึ้น วิธีการของ BAR CHART เป็นวิธีการที่ง่ายที่สุดและยังมีประสิทธิภาพในการ แสดงความสัมพันธ์ของเวลากับงานที่เกี่ยวข้อง ซึ่งต้องการควบคุมและติดตามผลอย่างใกล้ชิด แต่นอกจาก ความสัมพันธ์ระหว่างงานกับเวลาแล้ว ปัญหาด้านค่าใช้จ่ายก็เป็นอีกปัญหาที่มีความสำคัญมาก จึงได้มีการขยายขอบเขตของการวางแผนงานโดยการปรับปรุง BAR CHART ให้มีข้อมูลด้านค่าใช้จ่ายเพิ่มเข้าไปในแผนงาน เพื่อช่วยให้การควบคุมและการติดตามผลงานของโครงการให้มีประสิทธิภาพมากยิ่งขึ้น

การจัดทำปริญญานิพนธ์นี้เป็นโครงการที่มุ่งหวังที่จะนำเครื่องคอมพิวเตอร์มาช่วยในการวางแผนงาน และการควบคุมงาน เนื่องจากคอมพิวเตอร์มีความสามารถในการเก็บข้อมูลและเปลี่ยนแปลงแก้ไข ได้อย่างมีประสิทธิภาพ ทั้งยังมีความรวดเร็วในการนำเสนอ เหมาะสมกับการนำคอมพิวเตอร์มาใช้ในด้านนี้

Project Title      COMPUTER PROGRAMING FOR BAR CHART PLANNING  
 Student            Mr.PIRIYA            KRAWKANGWAN  
 Project Advisor   Mr.VIBOL            VOTTIYAN  
 Level of Study    Bachelor of Engineering in Construction Engineering  
 Department       Civil Engineering Faculty of Engineering King Mongkut  
                      's Institute of Technology Ladkrabang  
 Year                1991

## ABSTRACT

The increasing development and application of quantitative management methods are requiring construction managers to manipulate, summarize, and interpret larger and larger volumes of numerical information. For this problem, many management procedures such as Bar chart, CPM, PERT and etc. were made for planning and scheduling to compute and condense most forms of quantitative information. These are rote procedures and are essentially clerical in nature. Cost aspect is one of the most aspects of project that must be concerned about. Also, BAR CHART method was an extension from original idea to add cost elements to the network to plan for cost aspect of the project just like time aspect.

Despite the simplicity of the data reduction procedures, however, appreciable time and expense can be involved and computation errors are always possible. To an ever larger extent, the project manager is relying on computer and associated items of data processing equipment to provide him with management information. Project managers must react quickly to changing conditions and their decisions should be made promptly on the basis of adequate, accurate, and current information. The computer can help to make this possible as well as the evaluation of alternative courses of action.

### กิติกรรมประกาศ

การจัดทำปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยความกรุณาของ อาจารย์ วิบูลย์ วุฒินุญ อาจารย์ จักรพงษ์ พงษ์เพ็ง และ คุณมณี อยู่ประพัฒน์ ที่ได้ให้คำปรึกษาและตรวจแก้ไขข้อบกพร่องของปริญญานิพนธ์ฉบับนี้ และขอขอบคุณคณาจารย์ภาควิชาโยธาทุกท่าน ที่ประสิทธิประสาทวิชาความรู้ต่าง ๆ ให้แก่ข้าพเจ้า

นอกจากนี้ ข้าพเจ้าขอขอบคุณเพื่อน ๆ และน้อง ๆ ที่ช่วยเหลือและให้กำลังใจแก่ข้าพเจ้าด้วยดีตลอดมา

ท้ายนี้ บุคคลที่มีพระคุณกับข้าพเจ้าที่มีอาจใช้หมดสิ้น คือ บิดา มารดา ของข้าพเจ้าที่ให้โอกาสข้าพเจ้าได้ศึกษาเล่าเรียนจนกระทั่งทุกวันนี้

พิริยะ

แก้วก้ววล

นักศึกษาผู้ประกาศ

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ .....	II
กิตติกรรมประกาศ .....	III
สารบัญ.....	IV
<b>บทที่ 1 บทนำ</b>	
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของโครงการพิเศษ .....	1
1.3 ทฤษฎี หรือ แนวคิดที่ใช้ในโครงการพิเศษ .....	2
1.4 ขอบเขตของโครงการพิเศษ .....	2
1.5 ขั้นตอนในการดำเนินงาน.....	2
1.6 ผลที่คาดว่าจะได้รับจากโครงการพิเศษ.....	3
<b>บทที่ 2 การวางแผนงาน</b>	
2.1 บทนำ.....	4
2.2 ความสำคัญของการวางแผนงาน.....	4
2.3 ประโยชน์และข้อดีข้อเสียของการวางแผนงานด้วยระบบ BAR CHART.....	4
2.4 ลักษณะโดยทั่วไปของ BAR CHART.....	5
2.5 การแบ่งเวลาและคำนวณเวลา แต่ละงานในโครงการ.....	7
2.6 เงื่อนไขที่ผู้ที่จะทำการวางแผนงานพึงมี.....	8
2.7 หลักการวางแผนงาน.....	8
2.8 การรายงานผลการก่อสร้าง.....	12
2.9 การแบ่งวงเงิน.....	14
<b>บทที่ 3 คู่มือการใช้โปรแกรม</b>	
3.1 คู่มือการใช้โปรแกรม	
3.1.1 ข้อควรรู้ก่อนการใช้งาน.....	16
3.1.2 เริ่มเข้าสู่โปรแกรม .....	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
3.1.3 การเริ่มสร้างโครงการใหม่ .....	18
3.1.4 การลบ แก้ไขและเพิ่มเติมข้อมูลเดิม .....	19
3.1.5 การป้อนข้อมูลและการแสดงผล .....	21
3.2 ลักษณะการใช้โปรแกรม.....	30
FLOW CHART ของโปรแกรมคอมพิวเตอร์.....	31
<b>บทที่ 4 บทสรุปและข้อเสนอแนะ</b>	
4.1 บทสรุป.....	36
4.2 ข้อเสนอแนะ.....	37
<b>เอกสารอ้างอิง.....</b>	<b>38</b>
ภาคผนวก ก. โปรแกรมตัวอย่างสำหรับผู้สนใจ	
ภาคผนวก ข. สถิติการทำงานก่อสร้าง ต่อ 1 วันทำงาน	
ภาคผนวก ค. โปรแกรมการวางแผนงานและติดตามผลงานระบบ BAR CHART	

## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาและความสำคัญของปัญหา

ในการวางแผนงานก่อสร้างโดยทั่ว ๆ ไปแล้วมักใช้การวางแผนงาน โดยระบบ BAR CHART เนื่องจากจากระบบนี้ง่ายแก่การพิจารณาของผู้ที่สนใจและผู้ปฏิบัติงานที่เกี่ยวข้อง แต่สืบเนื่องด้วยการวางแผนงาน โดยระบบ BAR CHART นั้น ไม่ได้คำนึงถึงผลที่สืบเนื่องมาจากค่าใช้จ่ายจากวัสดุที่ใช้ในการดำเนินการและค่าจ้างแรงงานที่เกิดขึ้นในการดำเนินงานนั้น จึงมีผู้คิดการนำเสนอแผนภูมิ BAR CHART โดยเพิ่มการแสดงผลทางด้านค่าใช้จ่ายอันสืบเนื่องมาจากค่าวัสดุและค่าแรงเข้าไปในแผนภูมิ

การวางแผนงานก่อสร้างตามความเป็นจริงแล้วอาจจำเป็นต้องมีการแก้ไขแผนงานเดิมที่วางไว้แล้ว เนื่องจากเจ้าของโครงการ การแก้ไขแบบจากผู้ออกแบบ หรือเหตุอื่น ๆ นอกจากนี้แล้วเพื่อการติดตามผลของการดำเนินการแล้วจึงจำเป็นต้องมีการรายงานผลของการดำเนินงานที่เกิดขึ้นจริง (ACTUAL) ด้วย เพื่อเปรียบเทียบกับแผนงานที่วางไว้ด้วยเพื่อที่ผู้เกี่ยวข้องสามารถพิจารณาความก้าวหน้าของการดำเนินการที่กำลังเป็นอยู่ได้โดยการนำเสนอที่ใช้มักเทียบเป็นเปอร์เซ็นต์จากงานที่ต้องดำเนินการ เนื่องจากการวางแผนงาน และการนำเสนอข้อมูลที่เกิดขึ้นดังกล่าวมักมีการแก้ไขและเปลี่ยนแปลงข้อมูลอยู่เสมอ ๆ และคอมพิวเตอร์ ก็เป็นเครื่องมือที่เหมาะสมกับการปฏิบัติใน ด้านการเก็บข้อมูล และการแก้ไข เปลี่ยนแปลงข้อมูลทางผู้ดำเนินการวิจัยจึงมีความคิดที่จะใช้คอมพิวเตอร์เข้ามาใช้ใน ด้านการวางแผน และ การติดตามผลงาน โดยหวังว่าจะเป็นจุดเริ่มต้นในการพัฒนาทางด้านการประยุกต์ใช้งานคอมพิวเตอร์ในการวางแผนงานก่อสร้าง

#### 1.2 วัตถุประสงค์ของโครงการพิเศษ

1. นำเอาคอมพิวเตอร์มาใช้ในการวางแผนงาน และติดตามผลงานก่อสร้าง ในระบบ BAR CHART เพื่อเป็นแนวทางขั้นต้นในการใช้งาน

#### 1.8 ทฤษฎี หรือ แนวคิดที่ใช้ในโครงการพิเศษ

เนื่องจากลักษณะของข้อมูลที่ใช้ในการดำเนินการมีลักษณะเป็นข้อมูลแบบไม่แน่นอน คือ มีขนาดของข้อมูลไม่จำกัด และมีลักษณะพิเศษคือมักจะสืบเนื่องกันซึ่งเหมาะที่จะใช้การเก็บข้อมูลเป็นแบบ LINK LIST เพื่อสะดวกในการค้นหา และแก้ไขข้อมูล นอกจากนั้นทางผู้ดำเนินงานมีความคิดที่จะใช้การป้อน

ข้อมูลเป็นแบบทำงานลงบนหน่วยความจำก่อนแทนที่จะเป็นการทำงานลงเพิ่มข้อมูลโดยตรง เพื่อช่วยในด้านการแก้ไขข้อมูลที่เกิดขึ้น( เช่น ที่เกิดขึ้นในโปรแกรม Micro feab )

ภาษาคอมพิวเตอร์ที่ผู้จัดทำเห็นว่าน่าจะเหมาะสมในการดำเนินการ ควรเป็นภาษาชั้นสูง เนื่องจากสามารถแก้ไข และทำความเข้าใจได้ง่าย และปัจจุบันนี้ก็มีภาษาคอมพิวเตอร์ชั้นสูงหลายตัวที่มีตัว Compiler ที่สามารถทำงานได้อัจฉริยะ และง่ายแก่การเข้าใจ เพราะมีลักษณะการเขียนคล้ายกับประโยคเงื่อนไขในสมการคณิตศาสตร์ เพื่อให้ผู้เขียนสามารถเข้าใจโครงสร้างของโปรแกรมได้ไม่ยากนัก

#### 1.4 ขอบเขตของโครงการพิเศษ

โครงการนี้ใช้โปรแกรมภาษา C เขียนโปรแกรมการวางแผนและติดตามผลงาน BAR CHART ที่ใช้ในงานก่อสร้างที่มีการเพิ่มการแสดงค่าใช้จ่ายที่เกิดขึ้น สืบเนื่องมาจากการดำเนินงานนั้น ๆ โดยใช้เครื่องคอมพิวเตอร์ เพื่อสะดวกในการใช้งาน และเพิ่มประสิทธิภาพของการควบคุมงานให้ดียิ่งขึ้น โดยจะมีการเก็บข้อมูลที่จำเป็นไว้สำหรับแก้ไขและเพิ่มเติมได้ พัฒนาคำแนะนำเสนอข้อมูลให้ง่ายแก่การเข้าใจของผู้เกี่ยวข้องในการดำเนินการด้านก่อสร้าง รวมทั้งสามารถพิมพ์ผลสรุปของข้อมูลเป็นรายเดือนได้ด้วย

ในส่วนของตัว SOFTWARE คือสามารถเก็บข้อมูลได้มากกว่า 200 รายการ สามารถเก็บข้อมูลลงแผ่นได้เพื่อใช้ในการดำเนินการครั้งต่อไปได้ สามารถแก้ไขข้อมูลที่มีอยู่ได้สะดวก และสามารถแสดงผลงานที่วางแผนกับผลงานที่เกิดขึ้นจริงเปรียบเทียบกันให้เข้าใจได้ง่าย และสามารถแสดงผลออกเครื่องพิมพ์เพื่อใช้ในการนำเสนอและการเก็บเป็นเอกสารอ้างอิงได้

ในส่วนของ HARD WARE ก็จะเป็นเครื่อง IBM PC หรือ เครื่อง COMPATIBLE อื่น ๆ บนระบบปฏิบัติการ Microsoft DOS V2.XX ขึ้นไป หรือ DRDOS V5 ขึ้นไป

#### 1.5 ขั้นตอนในการดำเนินงาน

1. รวบรวมข้อมูลและวางแผนขอบเขตอย่างกว้าง ๆ ในการวิจัยพัฒนา รวมไปถึงศึกษาข้อดีหรือจุดเด่นของงานวิจัยเดิม

2. ศึกษาข้อดีข้อเสียของโปรแกรมที่ใช้กันอยู่ทั่ว ๆ ไป

3. กำหนดขอบเขตของงานวิจัย

4. ทำการวิเคราะห์ข้อดีและข้อบกพร่อง และหาทางที่เหมาะสม เพื่อดำเนินการปรับปรุงงานวิจัย

5. เขียนโปรแกรมเพื่อใช้ในการวางแผนและการติดตามผลงาน

6. สรุปผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.6 ผลที่คาดว่าจะได้รับจากโครงการพิเศษ

ผลที่คาดว่าจะได้รับจากโครงการนี้ คือ ช่วยอำนวยความสะดวกในด้านการคำนวณ และ การจัดวางแผนงาน และติดตามผลการดำเนินการในการก่อสร้างที่มีข้อมูลด้านค่าใช้จ่ายเปรียบเทียบด้วย เพื่อช่วยในการพิจารณาภาพรวมของการวางแผนงาน และ การติดตามงานที่เปลี่ยนแปลงไปที่เกิดขึ้นระหว่างการดำเนินการที่ส่งผลกระทบต่อโครงการทั้งหมด โดยการนำคอมพิวเตอร์มาใช้อำนวยความสะดวกในการจัดทำ BAR CHART โดยเขียนโปรแกรมภาษา C ขึ้น เพื่อให้สามารถทำการประเมินผลเร็วได้ขึ้นและให้เกิดประสิทธิภาพสูงกว่าการดำเนินการด้วยการใช้ตาราง BAR CHART แบบเก่าที่เขียนด้วยมือขึ้น และเพื่อความสะดวกแก่การนำไปใช้งาน ผู้จัดทำหวังว่า โครงการนี้จะป็นขั้นตอนในการริเริ่มและพัฒนาขั้นต้น เพื่อการนำเอาคอมพิวเตอร์ มาประยุกต์ใช้ในด้านการวางแผนงาน และ เป็นแนวทางให้มีการพัฒนาต่อไป



## บทที่ 2

### วิธีการวางแผนงาน

#### 2.1 บทนำ

การวางแผน ( PLANING ) คือการคิดค้นและแสวงหาวิธีการทำงานก่อสร้าง เพื่อให้เสร็จสมบูรณ์ตามเป้าหมายที่กำหนดด้วยราคาที่เหมาะสมและคุณภาพมาตรฐานที่รับได้ภายในเวลาที่กำหนดไว้ จึงเป็นความพยายามที่จะใช้ทรัพยากรของงานก่อสร้างให้มีประสิทธิภาพ และเหมาะสมที่สุด ( ทรัพยากรของงานก่อสร้าง คือ กำลังคน , วัสดุ , เครื่องมือ , เครื่องจักร และ เงิน )

#### 2.2 ความสำคัญของการวางแผนงานก่อสร้าง

การวางแผนงานเป็นเครื่องมือของผู้บริหารใช้ในการสั่งการและควบคุมงาน เพื่อให้บรรลุถึงเป้าหมายของการดำเนินงานของกิจการหรือโครงการ ประโยชน์ของการวางแผน มีดังนี้

1. ใช้ควบคุมระยะเวลาของกิจกรรมที่รวมอยู่ในโครงการ ในโครงการก่อสร้างโครงการหนึ่งจะประกอบไปด้วยกิจกรรมหลายอย่างรวมกัน ซึ่งกิจกรรมต่าง ๆ เหล่านี้ต้องกำหนดเวลาให้อยู่ในระยะเวลาที่กำหนดเวลาคำนวณงาน
2. ใช้ในการจัดหาทรัพยากรเพื่อการผลิตเช่น วัสดุ ( Material ) แรงงาน ( Labour ) และเครื่องจักรต่าง ๆ
3. ใช้ในการจัดหาเงินเพื่อการก่อสร้าง เพราะเงินเป็นทรัพยากรที่สำคัญผู้บริหารจำเป็นต้องรู้สภาพการรับและจ่ายเงินตลอดเวลาดังแต่เริ่มโครงการจนเสร็จโครงการ โดยจะมีแผนการรับและการจ่ายเงิน ( CASH FLOW ) แสดงการรับและจ่ายเงินของโครงการ
4. ใช้ในงานควบคุมงานก่อสร้าง ผู้บริหารสามารถใช้แผนงานก่อสร้างเป็นเครื่องมือในการควบคุมงานของผู้รับเหมาช่วงควบคุมการเบิกเงินตามงวดงานที่กำหนด การทำงานก่อสร้างจะมีการตรวจสอบทุก ๆ เดือนเพื่อให้ทราบถึงความก้าวหน้าของการดำเนินงานโดยนำมาเปรียบเทียบกับแผนที่วางไว้ หากความก้าวหน้าของงานไม่เป็นไปตามแผนที่วางไว้ ก็จะได้ทำการปรับปรุงการทำงานให้ดีขึ้นก่อนที่จะสายเกินไปจนทำให้ การทำงานล่าช้าเกินกำหนดตามระยะเวลาแห่งสัญญาไป

#### 2.3 รูปแบบของการวางแผนงาน

มีหลายวิธี เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Gantt chart จัดทำและเข้าใจได้ง่าย แต่ไม่สามารถระบุความสัมพันธ์ของงานได้
2. CPM สามารถพิจารณางานใดที่ช้ากว่ากำหนดแล้วทำให้งานทั้งโครงการล่าช้าออกไป
3. PERT คล้ายกับ CPM แต่ใช้กับงานที่ไม่เคยปฏิบัติมาก่อน โดยการเปรียบเทียบเวลาที่น่าจะเป็นของงานนั้น ๆ
4. Line of Balance สามารถพิจารณารายละเอียดของโครงการได้มาก แต่ยุ่งยากในการใช้งาน
5. Precedence Net Work การวางแผนงานแบบคำนึงถึงความสัมพันธ์ของงานทั้งโครงการ โดยใช้โครงข่ายแทนความสัมพันธ์

ในที่นี้จะขอกล่าวแต่วิธี BAR CHART

การวางแผนด้วยระบบ BAR CHART เริ่มใช้กันมาตั้งแต่ พ.ศ. 2463 โดยวิศวกรชื่อ HENRY L. GANTT เป็นผู้ริเริ่มคิดค้นขึ้นมาจากการวิเคราะห์เปรียบเทียบระหว่างการทำงานกับเวลาที่ต้องใช้กับงานนั้น โดยกำหนดเป็นรูปตารางและได้มีการปรับปรุงรูปแบบกันในเวลาต่อมาแต่ก็ยังคงเรียกว่า BAR CHART หรือ GANTT CHART เพื่อเป็นเกียรติแก่ผู้ริเริ่ม

ความมุ่งหมายของ BAR CHART คือ การคาดคะเนหรือกำหนดเวลาทำงานของโครงการว่าควรจะมีเท่าไรจึงแล้วเสร็จ โดยกำหนดไว้ล่วงหน้าในรูปของตารางเวลา ประกอบด้วยแกนตั้งและแกนนอน คือ แกนเวลาทำงานของแต่ละงาน โดยแสดงด้วยสัญลักษณ์รูปแท่ง (BAR) ความมุ่งหมายก็เพื่อให้ผู้ควบคุมโครงการหรือผู้ปฏิบัติงานใช้เป็นหลักในการควบคุมงานให้เป็นไปตามกำหนด โดยเปรียบเทียบกับงานจริง (ACTUAL WORK) กับงานที่วางแผน (PLANING WORK)

## 2.4 ลักษณะโดยทั่วไปของ BAR CHART

ในโครงการต่าง ๆ ย่อมประกอบไปด้วยกระบวนการทำงาน (PROCESS OF WORK) ตัวอย่างเช่น ขั้นตอนการออกแบบงานก่อสร้าง จะประกอบด้วยกระบวนการ ดังนี้

1. เซ็นสัญญา
2. รวบรวมข้อมูล
3. ทำการออกแบบ
4. เขียนแบบ
5. ส่งงาน

กระบวนการเหล่านี้เราเรียกว่า (ACTIVITY) ซึ่งจะมีในทุกโครงการ และจำนวนหน่วยงานย่อยเหล่านี้จะมีมากหรือน้อย ก็จะขึ้นอยู่กับขนาดของโครงการ จากกระบวนการข้างต้นเราจะเห็นได้ว่างานในโครงการนี้แบ่งเป็น 5 งานย่อยขั้นต่อไปในการพิจารณา คือ แต่ละงานควรมีระยะเวลาการทำงานเท่าไร

สมมติว่า

	ACTIVITY	DURATION ( วัน )
1	เซ็นสัญญา	2
2	รวบรวมข้อมูล	7
3	ทำการออกแบบ	15
4	เขียนแบบ	7
5	ส่งงาน	1

สมมติว่า โครงการนี้ ใช้คนทำงานคนเดียว วิธีการทำงานจะต้องทำงานหนึ่งให้เสร็จเสียก่อน จึงจะทำงานอื่นต่อไปได้ และจะต้องเรียงลำดับการทำงาน (SEQUENCE OF WORK) ตามลักษณะความเป็นจริงของการทำงานด้วย เมื่อเข้าตาราง BAR CHART จะได้ดังนี้

ลำดับ	หน่วยงานย่อย	d	เวลาการทำงาน																																		
			1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3	3	3				
1	เซ็นสัญญา	2	-	-																																	
2	รวบรวมข้อมูล	7			-	-	-	-	-	-																											
3	ทำการออกแบบ	15									-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
4	เขียนแบบ	7																																			
5	ส่งงาน	1																																			

สรุปประโยชน์ที่เราได้จาก BAR CHART คือสามารถแสดงเวลาทำงานของแต่ละ ACTIVITY ให้เห็นอย่างชัดเจน คือ เห็นว่า ACTIVITY จะเริ่มทำเมื่อไร และจะต้องเสร็จงานเมื่อไร

นอกจากนั้น ยังแสดงให้เห็นเวลาทำงานทั้งหมดของโครงการอย่างชัดเจน ข้อดีอีกประการหนึ่งของ BAR CHART คือ สามารถปรับแก้เวลาทำงานของแต่ละหน่วยงานย่อยได้ง่าย แต่อย่างไรก็ตาม BAR CHART ก็มีข้อเสียอยู่บ้าง คือ เราไม่สามารถทราบได้ว่า ในโครงการนั้น มี ACTIVITY ใดบ้างที่ต้องเร่งรีบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำเป็นพิเศษ หากล่าช้าไปจะเป็นผลให้โครงการทั้งโครงการต้องล่าช้าไปด้วย

สรุปข้อดีของ BAR CHART คือ

1. จัดทำง่าย
2. อ่านทำความเข้าใจง่าย แม้ระดับคนงานซึ่งอ่านหนังสือออกก็สามารถเข้าใจได้ไม่ยาก
3. ปรับแก้ได้ง่าย
4. สามารถเปรียบเทียบเวลาทำงานที่วางแผนไว้ ( PLANNING WORK ) กับเวลาที่ใช้ไปกับการทำงานจริง ( ACTUAL WORK ) ได้ชัดเจน

## 2.5 การแบ่งและคำนวณเวลา แต่ละส่วนงานของโครงการ

หมายถึง การหาระยะเวลาการทำงานของแต่ละงานย่อย ซึ่งเรียกว่า DURATION การกำหนดคว่างานย่อยใดควรมีระยะเวลาเท่าใดในการทำงานต้องพิจารณาจาก ปัจจัย 4 ประการคือ

1. ระยะเวลาทำงานใช้ชั่วโมงการทำงานปกติ ( Normal Hour ) ของบุคคลเป็นเกณฑ์ซึ่งเราเรียกเวลานี้ว่าเวลาปกติ ( Normal Duration )
2. จำนวนเครื่องมือเครื่องใช้ที่มีปริมาณที่เหมาะสมและสามารถเรียกใช้ได้
3. ขนาดของจำนวนบุคคล ( Gang Size ) ในแต่ละกิจกรรม เป็นขนาดที่พอเหมาะเป็นปกติ สำหรับทำกิจกรรมนั้น ๆ ที่ทำให้เกิดงานที่มีประสิทธิภาพที่สุด
4. อัตราการส่งวัสดุเป็นไปตามกำหนดเวลาที่ต้องการ

### 2.5.1 หลักเกณฑ์ในการพิจารณาหาจำนวนวันในการทำงาน

1. ต้องทราบปริมาณงานของหน่วยงานย่อย ซึ่งมีหน่วยต่างกัน เช่น เป็น เมตร.ตารางเมตร กิโลกรัม เป็นต้น นักวางแผนอาจจะขอทราบปริมาณงานดังกล่าวได้จาก B.O.Q (Bill Of Quantity) ของหน่วยงานประมาณราคา ซึ่งจะมีข้อมูลไว้ครบถ้วน
2. ต้องพิจารณาขีดความสามารถของผู้ดำเนินการ
3. การเผื่อเปอร์เซ็นต์การทำงาน เนื่องจากบางครั้งอาจมีเหตุขัดข้องเกิดขึ้นระหว่างการดำเนินงาน ทั้งที่ควบคุมได้และควบคุมไม่ได้

### 2.5.2 การประมาณระยะเวลาทำงาน มี 2 วิธีคือ

#### วิธีที่ 1 การประมาณระยะเวลาทำงาน โดยอาศัยสถิติการทำงาน

การประมาณว่า duration จะเป็นเท่าไรนั้นขึ้นอยู่กับปริมาณงานที่ทำ และปริมาณทรัพยากรที่ต้องใช้

เช่น งานขุดดิน 5 ลบม. ใช้คนงาน 1 คนขุดต้องกินเวลา 10 วัน ถ้าใช้คนงาน 2 คนขุดก็น่าจะใช้เวลา 5 วัน เป็นต้น ดังสมการคือ

$$D = \frac{\text{ปริมาณงาน}}{\text{จำนวนคนหรือเครื่องจักร} \times \text{ปริมาณงานที่ได้จากคน 1 คนหรือเครื่องจักร 1 เครื่อง}}$$

ในกรณีที่หาค่า  $d$  จากการทำงานของคน 1 คน ก็มักจะเรียกว่า Manday

\*\*\* ค่าสถิติความสามารถในการทำงานก่อสร้าง ต่อ 1 วันอยู่ในภาคผนวก ข. \*\*\*

### วิธีที่ 2 การประมาณระยะเวลาทำงาน โดยอาศัยประสบการณ์ของผู้ชำนาญงาน

วิธีนี้เป็นที่นิยมใช้กันในวงการก่อสร้างมาก ระยะเวลาในการทำงานจะประเมินจากงานที่เคยปฏิบัติมาแล้วว่างานนั้น ๆ มักจะใช้เวลาในการทำงานนานเท่าใด สภาพแวดล้อมหรือปัจจัยด้านอื่น ๆ เป็นอย่างไร วิธีนี้จะขึ้นอยู่กับประสบการณ์เป็นสำคัญ

### 2.6 เงื่อนไขของผู้ที่จะทำการวางแผนงานการทำงานพึงมีคือ

1. มีความรู้ด้านทฤษฎีพอสมควร
2. เคยทำงานร่วมกับผู้มีความรู้และประสบการณ์ในงานสนามมาแล้ว
3. เข้าใจรู้จักใช้วัสดุก่อสร้างให้คุ้มค่าที่สุด
4. รู้จุดบกพร่องของการปฏิบัติงานทั่วไป

### 2.7 หลักการวางแผนด้วยระบบ BAR CHART แบ่งเป็น 6 ขั้นตอนคือ

- ขั้นตอนที่ 1. ศึกษาว่าโครงการนั้นประกอบไปด้วยงานอะไรบ้าง
- ขั้นตอนที่ 2. จัดแบ่งงานออกเป็นหน่วยงานย่อย (ACTIVITY)
- ขั้นตอนที่ 3. เรียงลำดับการทำงานของหน่วยงานย่อยตามลักษณะก่อน-หลัง
- ขั้นตอนที่ 4. ทำบัญชีรายละเอียดของหน่วยงานย่อย เพื่อจัดทำตารางเวลาของหน่วยงานย่อยทั้งหมด
- ขั้นตอนที่ 5. หาปริมาณงานของแต่ละงาน เพื่อจัดทำตารางเวลาของหน่วยงานย่อยทั้งหมด
- ขั้นตอนที่ 6. จัดทำตาราง BAR CHART โดยนำข้อมูลจาก ข้อ 5 มาจัดลงในตาราง

ตัวอย่าง เช่น โครงการก่อสร้าง อาคารพาณิชย์ 3 ชั้น (พื้นที่ 300 ตร.ว.) ประกอบไปด้วยงานต่าง ๆ

ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### ขั้นตอนที่ 1

ศึกษาแบบ , รายการประกอบแบบ , สัญญาเงื่อนไขของโครงการ  
ลักษณะทั่ว ๆ ไป เป็นอาคารพาณิชย์ 3 ชั้นเป็นอาคารคอนกรีตเสริมเหล็กทั้งหลัง โครง  
หลังคาเหล็กมุงกระเบื้องลอนคู่ , ใช้ฝ้าแบบ T-BAR , มีระบบไฟฟ้า,มีรางระบายน้ำ  
งานแบ่งจ่ายเป็นรายเดือน โดยหักค่า RETAINED 10 %ของค่าใช้จ่าย

### ขั้นตอนที่ 2

แบ่งงานทั้งโครงการออกเป็นหน่วยงานย่อย โดยเรียงลำดับความขั้นตอนการทำงาน

1. เตรียมงาน
2. วางผังและตอกเข็ม
3. บุคหลุม
4. ทำฐานราก
5. กลบดิน
6. ปรับระดับกันหลุม ถมทรายอัดแน่น เทคอนกรีตหยาบ
7. ทำคานคอดินและพื้นชั้น 1
8. ทำเสาชั้น 1
9. ทำคานคอดินและพื้นชั้น 2
10. ทำเสาชั้น 2
11. ทำคานคอดินและพื้นชั้น 3
12. ทำเสาชั้น 3
13. ทำโครงหลังคา
14. อื่น ๆ
15. ตกแต่ง
16. งานประตูละ หน้าต่าง
17. งานท้อ
18. งานไฟฟ้า

### ขั้นตอนที่ 3

ทำบัญชีรายละเอียดกิจกรรมของแต่ละหน่วยงานย่อย ขั้นตอนนี้อาจข้ามไปเลยก็ได้เนื่องจากเป็น  
หน้าที่ของหน่วยงานสนามอยู่แล้วในการปฏิบัติงาน แต่ก็ควรทำเนื่องจากจะเป็นการตรวจสอบอีกครั้งหนึ่ง

ลำดับ	งานที่ต้องทำ	รายละเอียดของงาน
1	เตรียมงาน	เตรียมคนงาน, วัสดุอุปกรณ์, ปรับที่ เตรียมสาธารณูปโภค
2	วางผังและคอกเข็ม	ทำระดัปลอกคอกฝัง, คอกเข็ม
3	ขุดหลุม	ขุดหลุม
4	ทำฐานราก	เตรียมไม้แบบ, เตรียมเหล็ก, ตั้งไม้แบบ ,ผูกเหล็ก, เทคอนกรีต, แกะแบบ
5	กลับดิน	กลับดิน
6	ปรับระดับกันหลุม ถมทรายอัดแน่น เทคอนกรีตหยาบ	ปรับระดับกันหลุม ถมทรายอัดแน่น เทคอนกรีตหยาบ
7	ทำคานคอดินและพื้นชั้น 1	เตรียมไม้แบบ, เตรียมเหล็ก, ตั้งไม้แบบ ,ผูกเหล็ก, เทคอนกรีต, แกะแบบ
8	ทำเสาชั้น 1	เตรียมไม้แบบ, เตรียมเหล็ก, ตั้งไม้แบบ ,ผูกเหล็ก, เทคอนกรีต, แกะแบบ
9	ทำคานคอดินและพื้นชั้น 2	เตรียมไม้แบบ, เตรียมเหล็ก, ตั้งไม้แบบ ,ผูกเหล็ก, เทคอนกรีต, แกะแบบ
10	ทำเสาชั้น 2	เตรียมไม้แบบ, เตรียมเหล็ก, ตั้งไม้แบบ ,ผูกเหล็ก, เทคอนกรีต, แกะแบบ
11	ทำคานคอดินและพื้นชั้น 3	เตรียมไม้แบบ, เตรียมเหล็ก, ตั้งไม้แบบ ,ผูกเหล็ก, เทคอนกรีต, แกะแบบ
12	ทำเสาชั้น 3	เตรียมไม้แบบ, เตรียมเหล็ก, ตั้งไม้แบบ ,ผูกเหล็ก, เทคอนกรีต, แกะแบบ
13	ทำโครงหลังคา	แต่งเสา, ทำโครงหลังคา
14	อื่น ๆ	ทำกำแพง, มุงหลังคา
15	ตกแต่ง	ทาสี
16	งานประตูและ หน้าต่าง	บานประตู หน้าต่าง และงานกระฉก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับ	งานที่ต้องทำ	รายละเอียดของงาน
17	งานท่อ	ขุดดินทำรางระบายน้ำ และงานท่อ
18	งานไฟฟ้า	เดินสายไฟ และ ติดตั้งดวงโคม

ขั้นตอนที่ 4

จัดทำตารางประมาณเวลาทำงานและบัญชีค่าใช้จ่าย

รายการ	งานที่ต้องทำ	DESCRIPTION	ระยะเวลา	ค่าวัสดุ	ค่าแรง
1	เตรียมงาน	Preparation	5	25,000	3,500
2	วางผังและตอกเข็ม	Piling	26	545,000	5,000
3	ขุดหลุม	Excavation	15	0	7,000
4	ทำฐานราก	Footing	15	35,000	2,800
5	กลบดิน	Backfill	10	0	3,000
6	ปรับระดับกันหลุม ถมทรายอัดแน่น เทคอนกรีตหยาบ	Sand /Lean	5	55,000	3,000
7	ทำคานคอดินและพื้นชั้น 1	Beam1/Slab1	15	100,000	4,800
8	ทำเสาชั้น 1	Column1	7	30,000	2,000
9	ทำคานคอดินและพื้นชั้น 2	Beam2/Slab2	15	100,000	2,000
10	ทำเสาชั้น 2	Column2	7	30,000	2,000
11	ทำคานคอดินและพื้นชั้น 3	Beam3/Slab3	15	100,000	8,000
12	ทำเสาชั้น 3	Column3	7	30,000	2,000
13	ทำโครงสร้างหลังคา	Roof Structure	10	80,000	4,000
14	อื่น ๆ	Other	5	150,000	7,000
15	ตกแต่ง	Finishing	15	90,000	8,000
16	งานประตูและ หน้าต่าง	Doors/window	5	30,000	2,000
17	งานท่อ	Pipe work	5	20,000	1,800
18	งานไฟฟ้า	Electrical	7	20,000	2,000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ขั้นตอนที่ 5

จัดทำตารางเวลา BAR - CHART ( ส่วนนี้อาจทำได้ด้วยมือหรือไม่ก็สามารถทำได้โดยการใช้โปรแกรมใดก็ได้ )

### 2.8 รายงานผลการก่อสร้าง

เพื่อให้มีความต่อเนื่องในการเข้าใจกับการวางแผนงานก่อสร้าง จึงขอนำการรายงานผลการก่อสร้างมาเสริมท้ายบทนี้

เมื่อวางแผนงานเสร็จสิ้นแล้ว และปรับปรุงจนเป็นที่ยอมรับของทุกฝ่ายแล้ว เมื่อเริ่มทำการก่อสร้าง ก็ต้องดำเนินงานก่อสร้างให้เป็นไปตามแผนที่วางไว้ แต่ในการปฏิบัติจริงจะแตกต่างจากแผนที่วางไว้ โดยมากจะต่างจากแผนที่กำหนดไว้ไม่มาก ถ้าแตกต่างกันมากต้องตรวจสอบว่าเป็นเพราะวางแผนไม่ถูกต้องตามความเป็นจริง หรือผิดพลาดเนื่องจากการปฏิบัติงาน จะได้นำมาปรับปรุงแก้ไขในการวางแผน และ การปฏิบัติงานต่อไป

ในการรายงานผลการก่อสร้าง ส่วนที่วิศวกรสนามต้องรายงานก็คือผลของการทำงานซึ่งโดยทั่ว ๆ ไปจะรายงานสรุปไว้ในรายงานประจำเดือนว่าผลงานที่ได้ปฏิบัติจริง ( ACTUAL ) ได้ผลงานเท่าใด ซึ่งจะคิดเป็นร้อยละของค่าก่อสร้างของงานนั้น โดยคิดรวมจากผลงานต่าง ๆ ว่างานแต่ละอย่างนั้นมีผลงานคิดเป็นปริมาณงานเท่าใด เมื่อนำไปคูณด้วยค่าก่อสร้างต่อหน่วยของงานนั้น ก็จะได้ผลงานคิดเป็นบาทนำค่าก่อสร้างของงานนั้น ๆ ไปหาร แล้วคูณด้วย 100 ก็จะได้ผลงานของงานนั้นคิดเป็นร้อยละของค่าก่อสร้างของงานนั้น วิธีการคำนวณหาผลงานจริงสะสมของงานทั้งหมดเทียบกับผลงานตามที่วางแผนไว้เป็นร้อยละด้วยวิธีการเดียวกัน คือนำผลงานจริงสะสมหารด้วยราคาค่าก่อสร้างทั้งหมด เช่น สมมุติงานที่วางแผนแล้วนั้นมีการทำงานจริงเกิดขึ้นในงวดงานนั้น ๆ โดยเทียบจากปริมาณงานที่สำเร็จเป็นเกณฑ์แล้วเปรียบเทียบกับเป็นจำนวนเงินเพื่อสะดวกในการขออนุมัติเบิกจ่าย

จากตัวอย่างงานก่อสร้างอาคารพาณิชย์ 3 ชั้น ซึ่งได้วางแผนการทำงานไว้ข้างต้นของบทนี้ สมมุติว่า ในเดือนแรกมีผลงานที่ปฏิบัติจริง ดังนี้

1. งานเตรียมงาน ซึ่งประกอบไปด้วยการปรับพื้นที่ การจัดเตรียมสาธารณูปโภคต่าง ๆ แล้วเสร็จ คิดเป็นเงิน 28500 บาท
2. งานตอกเสาเข็ม เป็นงานเหมาะสมช่วงที่ผู้รับเหมารับผิดชอบงานทั้งหมด และได้ดำเนินการแล้วเสร็จไปแล้ว มีค่าใช้จ่ายคิดเป็นเงิน 550000 บาท

3. การขุดดิน ได้เริ่มดำเนินการไปแล้ว 1 วัน ใช้จ่ายไปทั้งสิ้น 500 บาท ใ้้งานประมาณ 8 ลบ.ม.จากงานทั้งหมด 45 ลบ.ม.

ก่อนที่จะคำนวณหาผลงาน จำเป็นต้องเข้าใจแนวความคิดที่จะคำนวณก่อนดังนี้  
รายงานผลการก่อสร้าง โดยทั่วไปก็เช่นเดียวกับรายงานอื่น คือ จะขึ้นอยู่กับผู้อ่าน  
การรายงานต้องสมมุติตัวเอง ว่าผู้อ่านอยากทราบรายละเอียดเกี่ยวกับอะไรบ้าง

จากแนวความคิดดังกล่าวทำให้การรายงานผลการก่อสร้างจึงต้องคิดผลงานตามปริมาณงาน  
ที่ประมาณการไว้ในบัญชีปริมาณงานก่อสร้าง เว้นแต่งานก่อสร้างที่ชำระเงินตามผลการก่อสร้างจริง  
จึงจะรายงานไปตามผลงานจริง เช่นในกรณีที่ทีมงานที่ต้องใช้จ่ายเกินจริงการรายงาน ผลงานก็จำเป็นต้องใช้  
จำนวนปริมาณที่ระบุไว้ในบัญชีปริมาณงาน แต่ทั้งนี้ก็ขึ้นอยู่กับชนิดของสัญญาที่ตกลงกันด้วย กรณีที่พูด  
ถึงนี้จะอิงการทำสัญญาแบบ LUM SUM ซึ่งเป็นแบบที่นิยมใช้กันทั่วไปในเวลานี้ เนื่องจากหลังการตกลง  
เซ็นสัญญา แล้วฝ่ายเจ้าของไม่ต้องรับภาระเพิ่มในภายหลัง เช่น จากตัวอย่างผลงานที่เกิดขึ้นในเดือนแรก

1.งานเตรียมงาน คิดเป็นเงิน 28500 คิดเป็นร้อยละ 100 ของงานนั้น แต่คิดเป็นร้อยละ 2.17  
ของค่า ก่อสร้างทั้งหมด

2.งานตอกเสาเข็ม เป็นงานเหมาช่วง มีค่าใช้จ่ายคิดเป็นเงิน 550000 คิดเป็นร้อยละ 100  
ของงาน นั้น แต่คิดเป็นร้อยละ 41.8 ของ ค่าก่อสร้างทั้งหมด

3. การขุดดิน ได้เริ่มดำเนินการไปแล้ว 1 วัน ใช้จ่ายไปทั้งสิ้น 2000 บาท จกค่าใช้จ่ายจริง  
7000 บาท ใ้้งานประมาณ 8 ลบ.ม.จากงานทั้งหมด 45 ลบ.ม. เทียบเป็นร้อยละ โดยการเฉลี่ยค่าดำเนินการ  
ต่อหน่วยออกมาได้ = 145.83 บาท/ลบ.ม. เทียบจากงานที่ทำได้เป็น  $\frac{8 \times 100}{45} \times 145.83 = 2592.6$  บาท  
หรือร้อยละ 18 ของงานนั้น และร้อยละ 0.2 ของค่าก่อสร้างทั้งหมด

$$\begin{aligned} \text{รวมผลงานในเดือนแรก} &= 2.17 + 41.8 + 0.2 \\ &= 43.99 \end{aligned}$$

คิดเป็นร้อยละ 43.99 ของค่าก่อสร้างทั้งหมด

และนำตัวเลขนี้ไปกำหนดจุด เพื่อเขียนเส้น S-curve ตามผลงานที่ปฏิบัติจริง (ACTUAL)  
ผลงานในเดือนอื่น ๆ ก็มีแนวทางในการคำนวณ เช่นเดียวกัน

นอกจากนี้ ในการคำนวณหาปริมาณงานเป็นร้อยละของค่าก่อสร้างทั้งหมดของงาน  
ก่อสร้างทั้งอาคารอื่น ๆ หน่วยงานแต่ละแห่งอาจจะมีแนวคิดในการคิดปริมาณงานต่างจากที่ระบุไว้ข้างต้น  
ซึ่งผู้ทำรายงานประจำเดือน ต้องทราบว่าแนวความคิดที่ต้องการในการหาปริมาณงานเป็นอย่างไร

บางหน่วยงานจึงไม่คิดผลงานที่ทำเสร็จ ( คิดร้อยละ 100 ) จะคิดเพียงผลงานที่ตรวจรับแล้ว  
ในกรณีที่งานยังไม่ได้ตรวจคิดเพียงร้อยละ 80-90 เท่านั้น โดยเฉพาะงานในงวดสุดท้ายจะคิดเพียงร้อยละ 90

เพื่อไว้สำหรับการเก็บงาน ทำความสะอาด แก้วใบงาน

ส่วนแผนการใช้วัสดุ แผนคนงาน และแผนการใช้เงินนั้น จะรายงานเฉพาะในองค์การธุรกิจ การก่อสร้างเท่านั้น ซึ่งทางฝ่ายควบคุมค่าใช้จ่าย ( COST CONTROL )จะเป็นผู้รายงานให้ผู้จัดการทราบว่า หน่วยงานนี้ได้ใช้วัสดุ คนงาน และเงินไปเป็นจำนวนเท่าใด แตกต่างจากการประมาณการและแผนงาน ไปเท่าใด ยอมรับได้หรือไม่ โดยมากถ้ามีความแตกต่างประมาณร้อยละ  $\pm 5$  ถือว่าปกติ ถ้าแตกต่างกัน มากกว่านี้ต้องทำการตรวจสอบด้วยว่าเป็น เพราะประมาณการผิดพลาด หรือสืบเนื่องจากการดำเนินงาน ผิดพลาด

## 2.9 การแบ่งงวดเงิน

เพื่อเป็นแนวทางในการปรับปรุงวางแผนการเงิน จึงรวบรวมวิธีการแบ่งงวดเงินของหน่วยงานต่งออกเป็น ดังนี้คือ

1. งวดเดียว คือการจ่ายเงินครั้งเดียวเมื่อเสร็จงาน จะใช้กับงานที่งบประมาณไม่มากนัก และระยะเวลาการทำงานน้อย วิธีนี้ผู้รับเหมาต้องมีทุนพอสมควร

### 2. หลายงวด มี 2 แบบ

2.1 เจ้าของงานแบ่งให้ โดยแบ่งตามขั้นตอนการก่อสร้างวิธีการนี้มีข้อเสียคือ หน่วยงานที่เป็นเจ้าของงานอาจมีความรู้ไม่เพียงพอ ทำให้การแบ่งงวดงานผิดขั้นตอน หรือผู้รับงานก่อสร้างปฏิบัติงานแตกต่างจากที่วางแผนไว้ทำให้ต้องส่งงานพร้อมกัน 2 งวดหรือมากกว่า ซึ่งหมายความว่าผู้รับเหมาจะต้องใช้ทุนมากขึ้น

2.2 ผู้รับงานก่อสร้างแบ่งงวดงานเอง และให้เจ้าของงานพิจารณาอนุมัติ โดยมากจะจ่ายทุกเดือน วิธีนี้เจ้าของงานต้องพิจารณาให้ดี แต่การเปิดโอกาสให้ผู้รับเหมาเป็นผู้แบ่งงวดงานแสดงว่าเจ้าของเชื่อในความสามารถและเชื่อใจในการดำเนินการของผู้รับเหมามาก

3. แบ่งงวดงานเป็น 6 งวด งวดละร้อยละ 15 งวดสุดท้ายร้อยละ 25 ของราคาค่าก่อสร้างทั้งหมด หมายความว่า เมื่อผู้รับเหมา มีผลงานก่อสร้างคิดเป็นร้อยละ 15 ของค่าก่อสร้างทั้งหมด ก็สามารถดำเนินการขอส่งงวดงานได้ แต่งานที่ขอส่งงวดนั้นต้องเสร็จในส่วนนั้น เช่น งานผนัง ก็ต้องก่ออิฐฉาบปูนเสร็จเรียบร้อยแล้ว วิธีนี้มักใช้กับการวางท่อร้อยสายโทรศัพท์

4. แบ่งเป็นงวดตามกำหนด แต่จ่ายก่อนเพียงร้อยละ 10 ของค่าก่อสร้างทั้งหมด ของค่าก่อสร้างทั้งหมด หลังจากเซ็นสัญญา วิธีนี้ผู้รับงานก่อสร้างได้ประโยชน์มาก เนื่องจากผู้รับงานก่อสร้างไม่จำเป็นต้องมีทุนก็สามารถดำเนินการได้ วิธีนี้ต้องมีความไว้วางใจกันในระดับสูงมาก

5. แบ่งจ่ายเฉพาะค่าแรง งานก่อสร้างบางงาน เจ้าของงานจะให้ประมูลเฉพาะค่าแรงงาน

ค่าดำเนินการ ค่าใช้จ่ายอื่น ๆ ค่าไรและภาษี โดยเจ้าของงานเป็นผู้รับจัดหาวัสดุก่อสร้าง การจ่ายเงินจะ  
จ่ายเงินทุกเดือนตามผลงานที่มี หักค้ำเงินค้ำประกัน(Retained Money) เพื่อไว้สำหรับแก้ไขงานประมาณ  
ร้อยละ 10 วิธีนี้ผู้รับเหมาต้องเสนอรายการคำนวณปริมาณงานค้ำ

วิธีนี้ ถ้าผู้รับงานก่อสร้าง ใช้กับผู้รับเหมาช่วง ( Sub-contractor ) จะจ่ายให้เพียงร้อยละ  
50-80 เมื่อตรวจรับงานแล้วจึงจ่ายส่วนที่เหลือให้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 8

### ลักษณะการใช้โปรแกรม

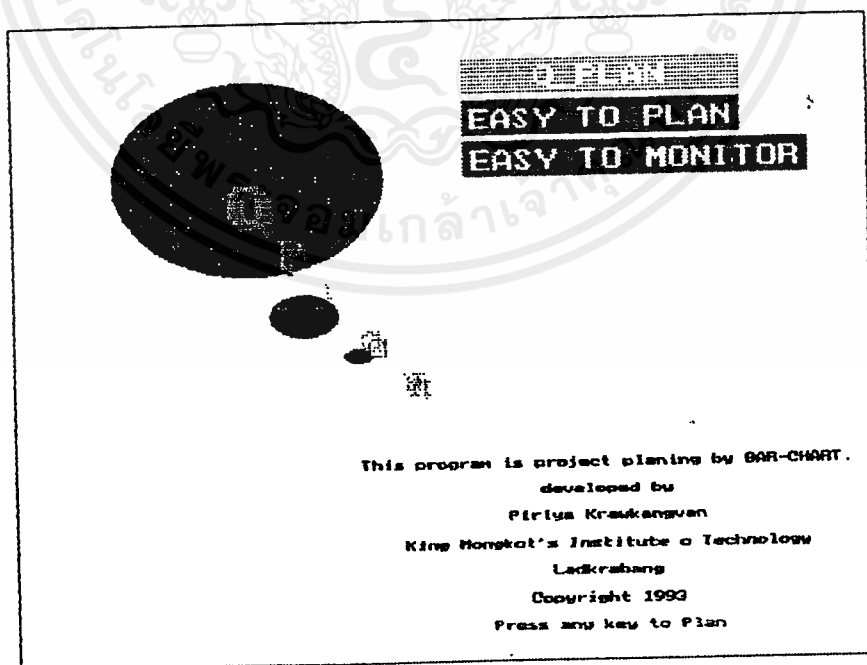
#### 8.1 คู่มือการใช้โปรแกรม

##### 8.1.1 ข้อควรรู้ก่อนใช้งาน

1. ควรแน่ใจว่าแผ่นโปรแกรมมีไฟล์สำหรับการ RUN ครบ
  2. ควรใช้กับเครื่องที่มี CPU หมายเลข 8088 ขึ้นไป ทางที่ดีควรเป็นหมายเลข 80386 ขึ้นไปเพื่อช่วยให้การทำงานรวดเร็วยิ่งขึ้น และควรมีหน่วยความจำมากกว่า 640 K bytes
  3. DISKETE ใหม่ที่จะนำมาเก็บข้อมูล (เรียกว่า DATA DISK ) จะต้องนำมา FORMAT \* เสียก่อน  
\*หมายเหตุ หากอ่านได้จากคู่มือการใช้ DOS ทั่วๆไป
  4. ชื่อไฟล์ที่ป้อนเข้าไปไม่ควรซ้ำกัน เพราะข้อมูลใหม่จะไปลบข้อมูลเดิมที่มีอยู่
  5. โปรแกรมจะไม่เปลี่ยนแปลงข้อมูลในแผ่นดิสจนกว่าจะมีการเรียกคำสั่ง SAVE เสียก่อน
- เนื่องจากเป็นโปรแกรมการทำงานแบบ VIRTUAL MEMORY
6. ก่อนจบโปรแกรมทุกครั้งควรเลือกฟังก์ชัน EXIT หรือคีย์ ESC ที่ MAIN MENU ทุกครั้ง

##### 8.1.2 เริ่มใช้โปรแกรม

ให้นำ แผ่น disk สอดเข้าไปใน Drive A: แล้วพิมพ์ A> q\_plan  
จากนั้นเครื่องจะทำการ RUN โปรแกรม แล้วก็โชว์ภาพดังรูปที่ 3.1



รูปที่ 3.1 แสดงชื่อโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ใช้จะสามารถผ่านส่วนนี้ไปด้วยการกดปุ่มใด ๆ บนแป้นคีย์ ก็สามารถเข้าสู่โปรแกรม ซึ่งจะมีภาพแสดงบนหน้าจอดังต่อไปนี้

Project planing by computer

developed by

Piriya Krawkangvan

King mongkot's Institue of Technology  
Ladkrabang  
copyright 1993

- \* use ARROW keys to move cursor
- \* Press ENTER key to select a menu item
- \* Press ESC key to exit menu
- \* Press F10 key to print out chart or summary

Begin new operation

**NEW** File Chdir Exit

รูปที่ 3.2 MAIN MENU แสดงรายการให้เลือกปฏิบัติ

ในรูปที่ 3.2 เป็น MAIN MENU จะประกอบไปด้วยฟังก์ชันต่าง ๆ ให้เลือก 4 ฟังก์ชันด้วยกัน คือ

- ฟังก์ชัน NEW ใช้เมื่อต้องการเริ่มการวางแผนทำงานใหม่
- ฟังก์ชัน FILE ใช้เมื่อต้องการทำงานเกี่ยวกับเพิ่มข้อมูล
- ฟังก์ชัน Chdir ใช้เมื่อต้องการเปลี่ยนช่องเก็บเพิ่มข้อมูล
- ฟังก์ชัน EXIT ใช้เมื่อต้องการออกจากโปรแกรม อาจใช้คีย์ BSC แทนก็ได้

ผู้ใช้สามารถเลือกจะปฏิบัติการเกี่ยวกับหน้าที่แบบใดโดยการใช้นิ้วชี้เลือกเลื่อน HILITE CURSOR ไปที่ตัวเลือกนั้น ๆ แล้วกดคีย์ ENTER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.3 การเริ่มสร้างโครงการใหม่

โดยการเรียกใช้ฟังก์ชัน NEW ที่อยู่ใน MAIN MENU โปรแกรมก็จะขึ้นที่หน้าจอ ENTER filename do not extension : ให้พิมพ์ชื่อไฟล์ลงไปโดยไม่ต้องมีนามสกุลต่อท้าย จากนั้นหน้าจอก็จะแสดงส่วนของการเก็บรายละเอียดของโครงการซึ่งมี 11 รายการที่ต้องป้อน ดังนี้

**TITLE** : This Project is for planning and monitoring  
**PROJECT** : barchart planning by computer  
**LOCATION** : civil department  
**ARCHITECT** :  
**ENGINEER** : Piriya Krawkanguan  
**OWNER** :  
**FOR PERIOD** : 1993  
**PROJECT START** : 2/ 6/93  
**TOTAL JOB** : 15  
**Time Scale** : DAY  
**UNIT/COST** : BATH

รูปที่ 3.3 รายละเอียดของโครงการ

**TITLE** : หัวเรื่องอาจจะเป็นชื่อของบริษัทที่รับผิดชอบสามารถป้อนได้ความยาวไม่เกิน 60 ตัวอักษร  
**PROJECT** : ชื่อโครงการ สามารถป้อนชื่อได้ความยาวไม่เกิน 60 ตัวอักษร  
**LOCATION** : ที่ตั้งของโครงการ สามารถป้อนที่อยู่ได้ความยาวไม่เกิน 40 ตัวอักษร  
**ARCHITECT** : สถาปนิกที่รับผิดชอบโครงการนี้ สามารถป้อนชื่อได้ความยาวไม่เกิน 40 ตัวอักษร  
**ENGINEER** : วิศวกรที่รับผิดชอบโครงการนี้ สามารถป้อนชื่อได้ความยาวไม่เกิน 40 ตัวอักษร  
**OWNER** : เจ้าของโครงการ สามารถป้อนชื่อได้ความยาวไม่เกิน 40 ตัวอักษร  
**FOR PERIOD** : ช่วงเวลาที่ใช้ในการดำเนินการก่อสร้าง สามารถป้อนได้ความยาวไม่เกิน 40 ตัวอักษร

**PROJECT START** : เวลาเริ่มต้นของโครงการ ป้อนแบบ วัน/เดือน/ปี ส่วนนี้จำเป็นต้องใส่  
**TOTAL JOBS** : 0 ( จะบอกถึงงานย่อยที่มีในโครงการ

ไม่ต้องกรอกข้อมูลลงไปเครื่องจะบันทึกเอง)

**Time Scale** : Days ( จะบอกการแสดงผลเป็นหน่วยวัน )

**COST / UNIT** : หน่วยวัดค่าใช้จ่ายที่ใช้ในโครงการ ( ควรกรอกเป็นหน่วยเงินตรา เช่น BATH )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นโปรแกรมก็จะเข้าสู่ฟังก์ชัน Add ของ PLAN MENU โดยอัตโนมัติดังรูปที่ 3.4

List of Activity :

ID	Job name	dur	start	finish	PreReq	Material cost	Labour cost
1	preparation	5	2/ 6/93	7/ 6/93	0	25000	5000
2	pile	26	5/ 6/93	30/ 6/93	1	545000	5000
3	excavation	15	1/ 7/93	15/ 7/93	2	0	7000
4	footing	15	1/ 7/93	15/ 7/93	2	25000	2000
5	backfill	10	16/ 7/93	25/ 7/93	4	0	3000
6	sand/lean	5	26/ 7/93	30/ 7/93	5	55000	3000
7	beam1	15	31/ 7/93	14/ 8/93	6	100000	8000
8	column1	7	8/ 8/93	14/ 8/93	6	20000	2000
9	beam2/slab2	15	15/ 8/93	29/ 8/93	7	100000	8000
10	column2	7	23/ 8/93	29/ 8/93	7	50000	2000
11	beam3/slab3	15	30/ 8/93	13/ 9/93	10	100000	8000
12	column3	7	7/ 9/93	13/ 9/93	10	30000	2000
13	roof beam	10	14/ 9/93	23/ 9/93	12	50000	40000
14	other	5	24/ 9/93	20/ 9/93	13	15000	7000
15	finishing	15	28/ 9/93	12/10/93	13	90000	8000

ID	Job name	dur	start	finish	PreReq	Material cost	Labour cost
----	----------	-----	-------	--------	--------	---------------	-------------

Add a new job to the list

Add Insert Erase Modify Save ON OFF

### รูปที่ 3.4 PLAN MENU เป็นรายละเอียดของงานแต่ละงาน

ในส่วนนี้จะประกอบไปด้วยรายละเอียดของแผนงานย่อยของโครงการมี 7 รายการที่ต้องป้อน ดังนี้  
ID : ส่วนนี้โปรแกรมจะทำการจัดการให้

Job name : รายละเอียดของงานย่อย สามารถป้อนได้ 20 ตัวอักษรแต่ก็สามารถเพิ่มได้ถ้าจำเป็น

dur : ระยะเวลา ( duration ) ที่ใช้ในการทำงาน สามารถป้อนได้ถึง 32767 วัน

Start : เวลาเริ่มต้นที่วางแผนไว้ของงานย่อยนั้น ๆ ป้อนแบบ วันที่/ เดือน/ปี

Finish : เวลาแล้วเสร็จตามแผนของงานย่อยนั้น ๆ ป้อนแบบ วันที่/ เดือน/ปี

PreReq : งานที่ต้องทำก่อน

Material cost : ค่าวัสดุที่ใช้ในงานนั้น สามารถป้อนได้ 10 หลักแต่จริงแล้วสามารถขยายเพิ่มได้ถึง

$$1.7 \times 10^{\pm 30}$$

Labour cost : ค่าแรงที่ใช้ในงานนั้น สามารถป้อนได้ 10 หลักแต่จริงแล้วสามารถขยายเพิ่มได้ถึง

$$1.7 \times 10^{\pm 30}$$

ฟังก์ชันนี้จะทำงานไปเรื่อย ๆ จนกว่าจะกด ESC ที่ Job name จึงจะออกจากฟังก์ชันนี้ ไปสู่ PLAN MENU แต่จะยังไม่ทำการเก็บข้อมูลนี้จนกว่าจะเรียกใช้ฟังก์ชัน SAVE

### 3.1.4 การลบ แก้ไข และ เพิ่มเติมข้อมูลเดิม

โดยการเรียกใช้ฟังก์ชัน FILE ที่อยู่ใน MAIN MENU ก็จะเข้าสู่ FILE MENU ซึ่งประกอบไปด้วยฟังก์ชันต่าง ๆ ให้เลือก 3 ฟังก์ชันด้วยกัน คือ

- LOAD ทำการเรียกเพิ่มข้อมูลเก่าขึ้นมาเพื่อทำการแก้ไข หรือ ปรับปรุง
- SAVE เก็บบันทึกข้อมูลที่อยู่ในหน่วยความจำลงแผ่นดิส
- DELETE ใช้เพื่อลบเพิ่มข้อมูลที่ไม่ต้องการออก

สมมุติว่าเรียกใช้ฟังก์ชัน LOAD หน้าจอก็จะมีชื่อ ไฟล์เดิมที่อยู่ในช่องเก็บข้อมูลปัจจุบันมาให้ ( รูป 3.4 ) แต่ถ้าไม่มีเพิ่มที่ต้องการมาให้ก็ให้กด ESC เพื่อกลับไปยัง MAIN MENU แล้วเลือกฟังก์ชัน Chdir เพื่อเปลี่ยน ไปยัง Sub-directory ที่มีไฟล์ที่ต้องการอยู่แล้วจึงกลับมาใช้ฟังก์ชัน LOAD โดยไม่ต้องออกไปจากโปรแกรม (\*\*หมายเหตุ\*\* ไฟล์ที่ใช้เป็นตัวอย่างนี้ชื่อ QTY.LST เป็นไฟล์ของ ตัวอย่าง โครงการอาคารพาณิชย์ที่ยกตัวอย่างแล้ว ในบทที่ 2 )

หลังจากเรียกฟังก์ชัน LOAD แล้วหน้าจอก็จะแสดงรายละเอียดของโครงการนั้นขึ้นมา และต่อด้วยการแสดงงานที่เคยป้อนไว้บนหน้าจอจากนั้นจะเข้าสู่ PLAN MENU เพื่อให้ผู้ใช้เลือกฟังก์ชันที่ต้องการเองต่อไป

ในรูปที่ 3.4 เป็น PLAN MENU จะประกอบไปด้วยฟังก์ชันต่าง ๆ ให้เลือก 7 ฟังก์ชันด้วยกัน คือ

- Add เป็นคำสั่งที่ใช้เพิ่มเติมข้อมูล และ ใช้ในกรณีที่เริ่มต้นป้อนข้อมูลด้วย
- Insert เป็นคำสั่งที่ใช้สำหรับแทรกข้อมูล
- Erase ใช้สำหรับลบข้อมูลที่ไม่ต้องการออกจากหน่วยความจำ
- Modify ใช้สำหรับการแก้ไขข้อมูลเดิมที่มีอยู่แล้ว
- Save ทำการเก็บข้อมูลในหน่วยความจำลงแผ่น
- ON ,OFF ใช้สำหรับกำหนดการแสดงผลของข้อมูลว่าจะแสดงข้อมูลที่มีอยู่หรือไม่ ถ้ากด ON

ก็จะแสดงผลของแผนงานบนหน้าจอเมื่อใช้คำสั่ง CHART ถ้า OFF ก็จะไม่แสดงผลของแผนงาน บนหน้าจอ เมื่อใช้คำสั่ง CHART

การเพิ่มเติมหรือทำการแก้ไขข้อมูลจากงานจริงต้องเข้าสู่ ACTUAL MENU ก่อนโดยเรียกผ่าน OPTION MENU ซึ่งจะอธิบายในหัวข้อ ถัดไป ACTUAL MENU จะมีลักษณะคล้ายกับ PLAN MENU โดยจะประกอบไปด้วยฟังก์ชันต่าง ๆ ให้เลือก 7 ฟังก์ชันด้วยกัน คือ

- Add เป็นคำสั่งที่ใช้เพิ่มเติมข้อมูล ในกรณีที่มีข้อมูลใหม่ที่ไม่ได้อยู่ในแผน
- Insert เป็นคำสั่งที่ใช้สำหรับแทรกเพิ่มเติมข้อมูล ในกรณีที่มีข้อมูลใหม่ที่ไม่ได้อยู่ในแผน
- Erase ใช้สำหรับลบข้อมูลที่ไม่ต้องการออกจากหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ModACT ใช้สำหรับการป้อนและแก้ไขข้อมูลเนื่องจากการทำงานจริงจากแผนเดิมที่วางไว้แล้ว
- Save ทำการเก็บข้อมูลในหน่วยความจำลงแผ่น
- DISPLAY ON , DISPLAY OFF ใช้สำหรับกำหนดการแสดงผลของข้อมูลว่าจะแสดงข้อมูลที่มีอยู่หรือไม่ ถ้ากด ON ก็จะแสดงผลของงานจริงบนหน้าจอเมื่อใช้คำสั่ง CHART ถ้า OFF ก็จะไม่แสดงผลของงานจริงบนหน้าจอ เมื่อใช้คำสั่ง CHART

ในที่นี้จะขอกล่าวถึงฟังก์ชัน Modify ที่ใช้ในการรายงานผลของงานที่เกิดขึ้นจริง โดยเมื่อเรียกใช้ฟังก์ชันนี้แล้วโปรแกรมจะถาม ID ของงานที่ต้องการเพิ่มข้อมูลของงานจริง เมื่อผู้ใช้ป้อนค่า ID ที่ต้องการลงไปแล้วโปรแกรมก็จะมีรายการให้ป้อนข้อมูลเพิ่มดังนี้คือ

ID : ต้องการเปลี่ยนตัวเลขแสดงค่า ID หรือ ไม่ถ้าต้องการใช้ค่าเดิมก็ป้อนลงไปใหม่เพื่อความมั่นใจ

Job name : ส่วนนี้โปรแกรมจะแสดงให้ดูว่าเป็นงานที่ต้องการใช้ใหม่

now : ส่วนนี้ให้ป้อน % เนื่องจากค่าใช้จ่ายที่เกิดขึ้นจริง

Complete : ส่วนนี้ให้ป้อน % เนื่องจากผลงานที่คำนวณอยู่ของงานนี้

begin : เวลาเริ่มต้นของงานที่เกิดขึ้นจริง ( ผู้เขียนโปรแกรมไม่ต้องการใช้ชื่อตัวแปรซ้ำกับ เวลาเริ่มต้นของแผนงานเพราะอาจทำให้เกิดความสับสนแก่ผู้ใช้ได้ )

end : เวลาที่งานนี้กำลังทำอยู่ครั้งสุดท้าย อาจจะยังไม่แล้วเสร็จได้

การเก็บข้อมูลในหน่วยความจำลงแผ่นนั้นใช้ฟังก์ชัน Save ซึ่งในตัวโปรแกรมแล้วจะใช้คำสั่งนี้ในเมนูย่อยใด ก็มีค่าเท่ากัน

### ร.1.5 การป้อนข้อมูลและการแสดงผล

List of Activity :

ID	Job name	dur	start	finish	PreReq	Material cost	Labour cost
1	preparation	5	2/ 6/93	7/ 6/93	0	25000	5000
2	pile	26	5/ 6/93	30/ 6/93	1	545000	5000
3	excavation	15	1/ 7/93	15/ 7/93	2	0	7000
4	footing	15	1/ 7/93	15/ 7/93	2	25000	2000
5	backfill	10	16/ 7/93	25/ 7/93	4	0	3000
6	sand/lean	5	26/ 7/93	30/ 7/93	5	55000	3000
7	beam1	15	31/ 7/93	14/ 8/93	6	100000	8000
8	column1	7	0/ 8/93	14/ 8/93	6	20000	2000
9	beam2/slab2	15	15/ 8/93	29/ 8/93	7	100000	8000
10	column2	7	23/ 8/93	29/ 8/93	7	50000	2000
11	beam3/slab3	15	30/ 8/93	13/ 9/93	10	100000	8000
12	column3	7	7/ 9/93	13/ 9/93	10	30000	2000
13	roof beam	10	14/ 9/93	23/ 9/93	12	50000	4000
14	other	5	24/ 9/93	20/ 9/93	13	15000	7000
15	finishing	15	20/ 9/93	12/10/93	13	90000	8000

ID Job name dur start finish PreReq Material cost Labour cost

Edit job

Plan Actual Describe Chart Summary Analysis

### รูปที่ 3.5 OPTION MENU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ในรูปที่ 3.5 เป็น OPTION MENU จะประกอบไปด้วยฟังก์ชันต่าง ๆ ให้เลือก 5 ฟังก์ชันด้วยกัน คือ
- PLAN ใช้สำหรับเพิ่มเติม หรือ แก้ไข แผนงานที่วางไว้ ( ค้างที่กล่าวไว้แล้วในหัวข้อก่อน )
  - ACTUAL ใช้สำหรับเพิ่มเติม หรือ แก้ไข งานจริงที่เกิดขึ้น ( ค้างที่กล่าวไว้แล้วในหัวข้อก่อน )
  - DESCRIP ใช้สำหรับเพิ่มเติม หรือ แก้ไข ส่วนหัวหรือรายละเอียดของแผนงานที่วางไว้
  - CHART ใช้แสดงผลงานและพิมพ์ผลงานที่วางแผน (plan) และ ที่เกิดขึ้นจริง (actual) และ S-curve และสามารถเลื่อนการแสดงผลได้ด้วยคีย์ลูกศร และจะแสดงผลออกทาง printer ด้วยเมื่อผู้ใช้กดคีย์ F10
  - SUMMARY แสดงผลสรุปการทำงานในงวดงานนั้น ๆ ออกทางหน้าจอและสามารถพิมพ์ออกทาง printer ได้ด้วยเมื่อผู้ใช้กดคีย์ F10 โดยโปรแกรมจะถามข้อมูลเพิ่มเติมก่อนเข้าหน้าจอ ที่ทำการสรุปผล
  - ANALYSIS แสดงผลการวิเคราะห์ความก้าวหน้าของโครงการโดยเทียบเป็นเปอร์เซ็นต์จาก ค่าใช้จ่ายที่เกิดขึ้นจริง / ค่าใช้จ่ายที่วางแผนไว้ สามารถแสดงผลได้ 2 แบบ คือ
    1. แบบเป็น S-curve และสามารถเลื่อนการแสดงผลได้ด้วยคีย์ลูกศร และจะแสดงผลออกทาง printer เมื่อผู้ใช้กดคีย์ F10 โดย S-curve ที่เกิดขึ้นมาจาก ค่าใช้จ่าย วางแผน (plan) เทียบกับค่าใช้จ่าย ที่เกิดขึ้นจริง ( actual )
    2. แบบเป็นบทสรุปค่าใช้จ่ายในการทำงานนั้น ๆ เมื่อผู้ใช้กดคีย์ F1 และสามารถพิมพ์ ออกทาง printer ได้ด้วยเมื่อผู้ใช้กดคีย์ F10

TITLE : This Project is for planning and monitoring					
PROJECT : barchart planning by computer					
LOCATION : civil department			ARCHITECT :		
ENGINEER : Piriya Krawkangvan			OWNER :		
FOR PERIOD : 1993			PROJECT START : 2/ 6/93		
TOTAL JOB : 15			Time Scale : DAY		
UNIT/COST : BATH			FILE : qty.des Page 1		
Id	Id Description	Total cost	Completed to Date	Cost to complete	Percent Complete
1	preparation	30000	30000	0	100
2	pile	550000	550000	0	100
3	excavation	7000	0	7000	0
4	footing	27800	0	27800	0
5	backfill	3000	0	3000	0
6	sand/lean	58000	0	58000	0
7	beam1	108000	0	108000	0
8	column1	22000	0	22000	0
9	beam2/slab2	108000	0	108000	0
10	column2	52000	0	52000	0
11	beam3/slab3	108000	0	108000	0
12	column3	32000	0	32000	0
13	roof beam	90000	0	90000	0
14	other	22000	0	22000	0
15	finishing	98000	0	98000	0
		1315800	580000	735800	44.1
A. Cost of work performed to date		580000			
B. Materials stored on site		0			
C. Total work performed and materials store		580000			
D. Less 10 % retained		58000			
E. Net work performed and materials store		522000			
F. Less amont of Previous payments		0			
G. Balance due this payment		522000			

Press Any key to Exit or F10 to print

### รูปที่ 3.6 ผลการทำงานของฟังก์ชัน Summary

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Curent FILE's name : qty.lst

Total job (on plan) : 15

SHOW PLAN : ON

SHOW ACTUAL : OFF

HEADER or DESCRIPTION for this plan

TITLE : This Project is for planning and monitoring  
 PROJECT : barchart planning by computer  
 LOCATION : civil department  
 ARCHITECT :  
 ENGINEER : Piriya Kraukangan  
 OWNER :  
 FOR PERIOD : 1993  
 PROJECT START : 2/ 6/93  
 TOTAL JOB : 15  
 Time Scale : DAY  
 UNIT/COST : BATH

If everything allright press anykey to process  
 else press ESC to exit

รูปที่ 3.7 หน้าจอก่อนการแสดงผลของฟังก์ชัน CHART

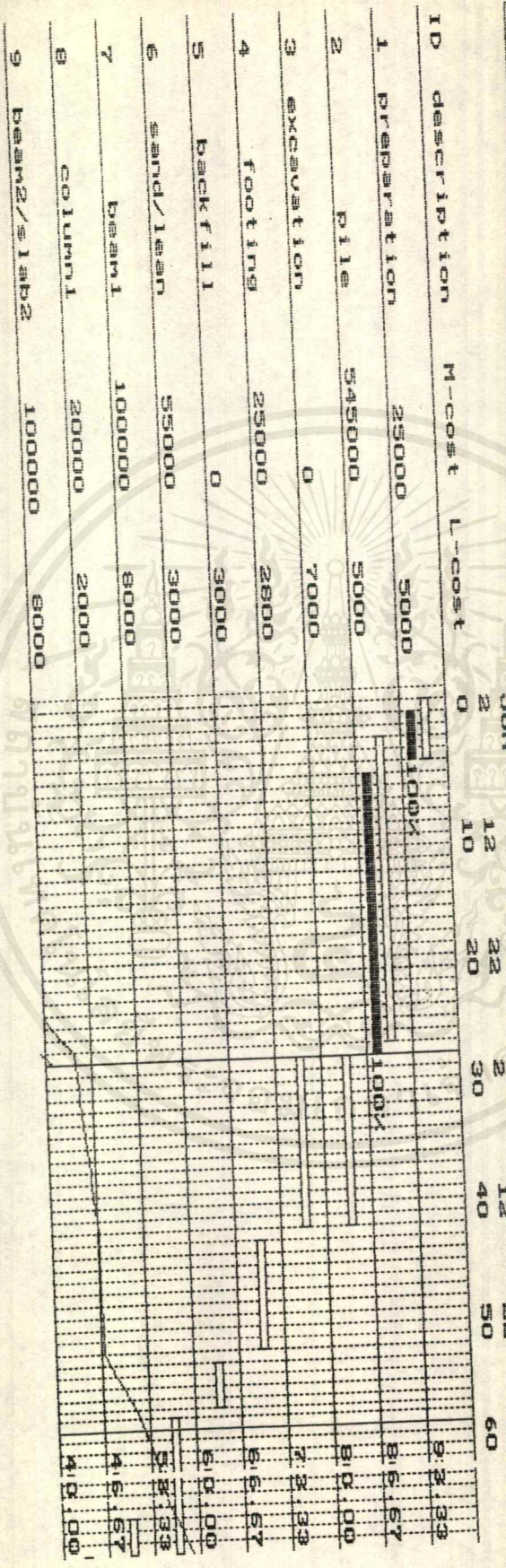
ID	description	d	M-cost	L-cost	0	10	20	30	40	50	60
1	preparation	5	25000	5000							50.00
2	pile	26	340000	3000							61.00
3	excavation	15	0	7000							7.70
4	footing	15	85000	2800							53.00
5	backfill	10	0	3000							3.30
6	sand/lean	5	55000	3000							45.00
7	beam1	15	100000	8000							35.00
8	column1	7	20000	2000							27.27
9	beam2/slab2	15	100000	8000							35.00
10	column2	7	30000	2000							3.30
11	beam3/slab3	15	100000	8000							35.00
<b>SUM</b>											
					516538	21154	690000	103000	516538	690000	

รูปที่ 3.8 ผลการทำงานของฟังก์ชัน CHART

F10 - print

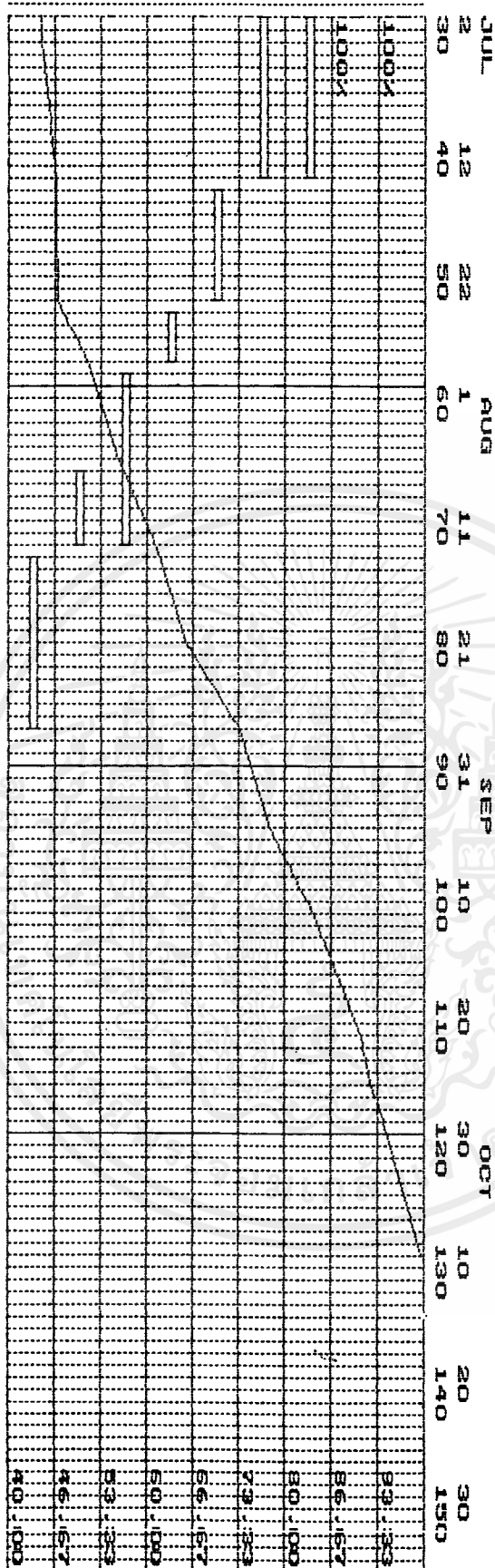
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TITLE : This Project is for planning and monitoring : civil department  
 PROJECT : barchart planning by computer LOCATION : Pivisa Kraakanavan  
 ARCHITECT : ENGINEER :  
 OWNER : FOR PERIOD : 1993  
 PROJECT START : 2/ 6/93 TOTAL JOB : 15 FILE : atv.1st Page 1  
 UNIT/COST : BATH

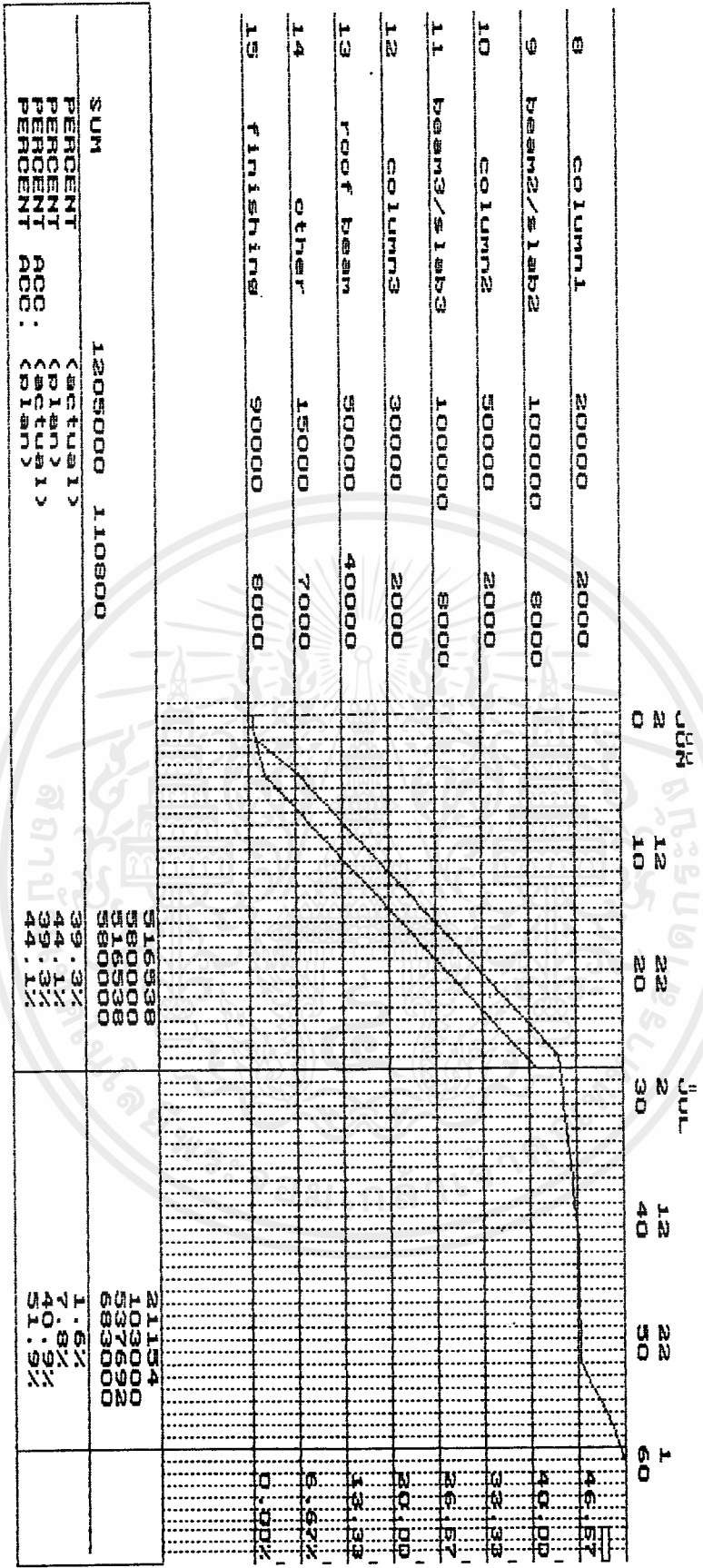


รูปที่ 3.9 ผลการพิมพ์ข้อมูลของทั้งชั้น CHART หน้า 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในองค์กรเท่านั้น ไม่ใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวน **รูปที่ 3.10 ผลการพิมพ์ข้อมูลของฟังก์ชัน CHART หน้า 2** หน้าไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 3.11 ผลการพิมพ์ข้อมูลของผังรับ CHART หน้า 3



Id	Id Description	Estimated cost	Real cost	Projected Saving	Loss	Total
1	preparation	30000	30000	0	0	0
2	pile	550000	550000	0	0	0
3	excavation	7000	0	0	0	0
4	footing	27800	0	0	0	0
5	backfill	3000	0	0	0	0
6	sand/lean	58000	0	0	0	0
7	beam1	108000	0	0	0	0
8	column1	22000	0	0	0	0
9	beam2/slab2	108000	0	0	0	0
10	column2	52000	0	0	0	0
11	beam3/slab3	108000	0	0	0	0
12	column3	32000	0	0	0	0
13	roof beam	90000	0	0	0	0
14	other	22000	0	0	0	0
15	finishing	98000	0	0	0	0

Analysis cost sheet

Press Any key to Exit or F10 to print

รูปที่ 3.13 ผลการทำงานของฟังก์ชัน ANALYSIS SUMMARY SHEET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ลักษณะของโปรแกรม

โปรแกรมการวางแผนงานและติดตามผลงานด้วยระบบ BAR CHART นี้เป็นโปรแกรมที่เขียนขึ้นโดยใช้ภาษา C ในการดำเนินงาน เนื่องจากโครงสร้างของภาษามีการแบ่ง function การทำงานออกเป็นส่วนย่อย ๆ สามารถเรียกใช้จากส่วนไหนก็ได้แม้แต่การเรียกใช้แบบ RECURSIVE ( การวน LOOP แบบเรียกตัวเอง ) การจองเนื้อที่สำหรับค่าตัวแปรไม่มากนักเนื่องจากตัวแปรจะทำงานเฉพาะใน function เท่านั้น การทำงานข้าม function ต้องมีการผ่านค่าตัวแปรให้แก่ function ที่เรียกด้วยทำให้ผู้เขียนรู้ว่าใน function นั้นมีตัวแปรอะไรบ้าง นอกจากนี้ภาษา C ยังเป็นโปรแกรมที่โปรแกรมเมอร์นิยมใช้กันมาก ลักษณะการเขียน เงื่อนไข และฟังก์ชันคล้ายกับภาษาปาสคาลด้วย ดังนั้นผู้ที่สนใจสามารถนำโปรแกรมที่เขียนนี้ไปแก้ไข และปรับปรุงได้โดยง่าย

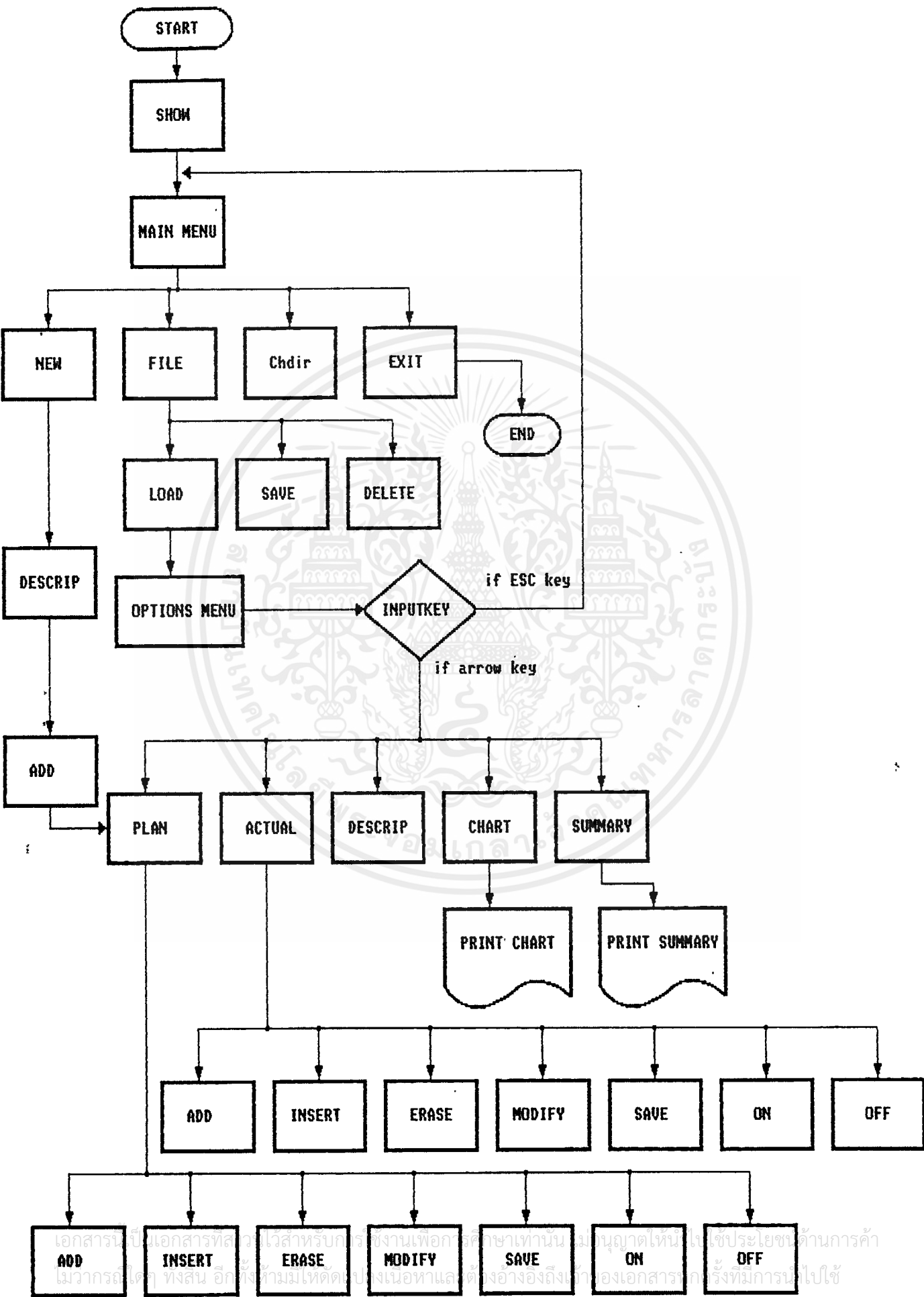
การเขียนโปรแกรมนี้ทางผู้จัดทำได้เขียนโปรแกรมเป็น PROJECT FILE โดยแยกตัวโปรแกรมออกเป็น 6 ไฟล์ย่อย คือ

1. LAMH ภายในประกอบไปด้วยตัวแปรหลักที่ใช้ในโปรแกรมทั้งหมด และการกำหนดฟังก์ชันที่ใช้งานให้สามารถเรียกได้ไม่ว่าจะอยู่ในไฟล์ใดก็ตาม
2. Q\_MENU.C ประกอบไปด้วยฟังก์ชันที่ใช้ในการเลือกการหน้าที่ที่ต้องการดำเนินการของโปรแกรม ส่วน function หลักที่เรียกฟังก์ชันอื่นในการทำงาน คือ function MAIN
3. Q\_INPUT.C เป็นไฟล์ที่เก็บ function ที่ใช้สำหรับการรับค่าตัวแปร เช่น การอ่านคีย์และฟังก์ชันที่เกี่ยวกับการอ่าน และเก็บรายละเอียดของโครงการ
4. Q\_FILE.C เป็นไฟล์ที่ดำเนินการเกี่ยวกับการแสดงผลและการเก็บข้อมูลของโปรแกรมสามารถเรียกได้ว่าเป็นส่วนหลักเลยก็ว่าได้
5. Q\_SUM.C เป็นไฟล์ที่ใช้ในการคำนวณและจองหน่วยความจำสำหรับค่าใช้จ่ายเป็นรายวันที่เกิดขึ้นในโครงการ
6. PRINTER.C เป็นไฟล์ที่รวบรวม function ที่ใช้ในการแสดงผลข้อมูลออกทางเครื่องพิมพ์

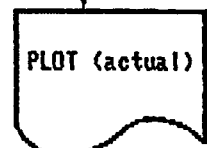
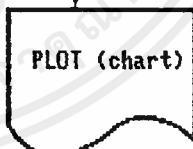
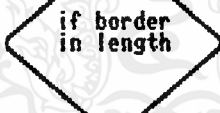
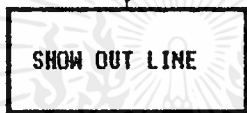
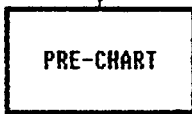
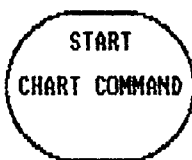
\*\*\*\* หมายเหตุ PROJECT FILE เป็นลักษณะการเขียนโปรแกรมภาษา C แบบหนึ่งที่มีกับโปรแกรมที่มีขนาด

ใหญ่ วิธีการเรียนโปรแกรมสามารถศึกษาได้จากคู่มือการใช้โปรแกรมภาษา C ( C USER GUIDE )

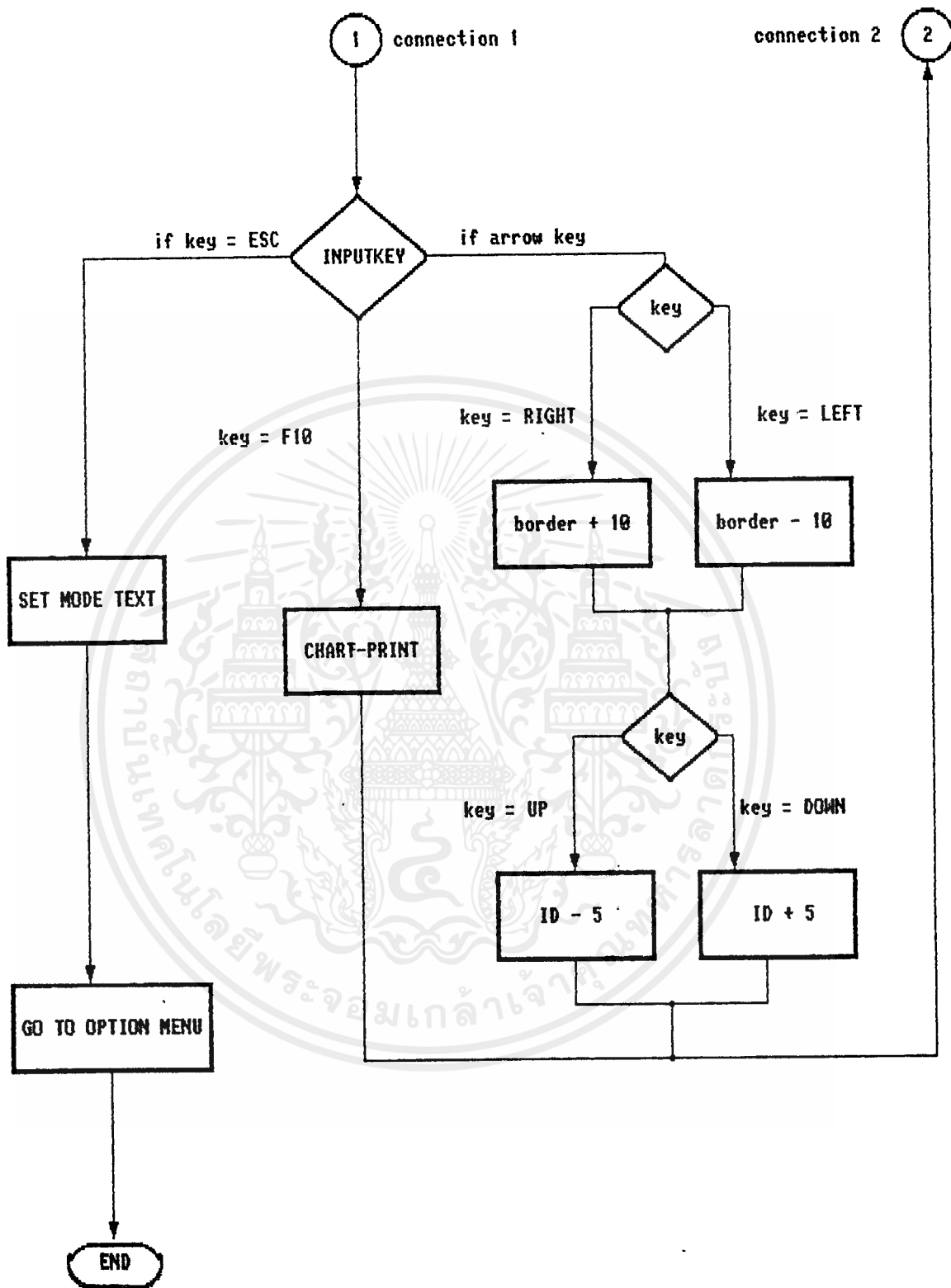
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



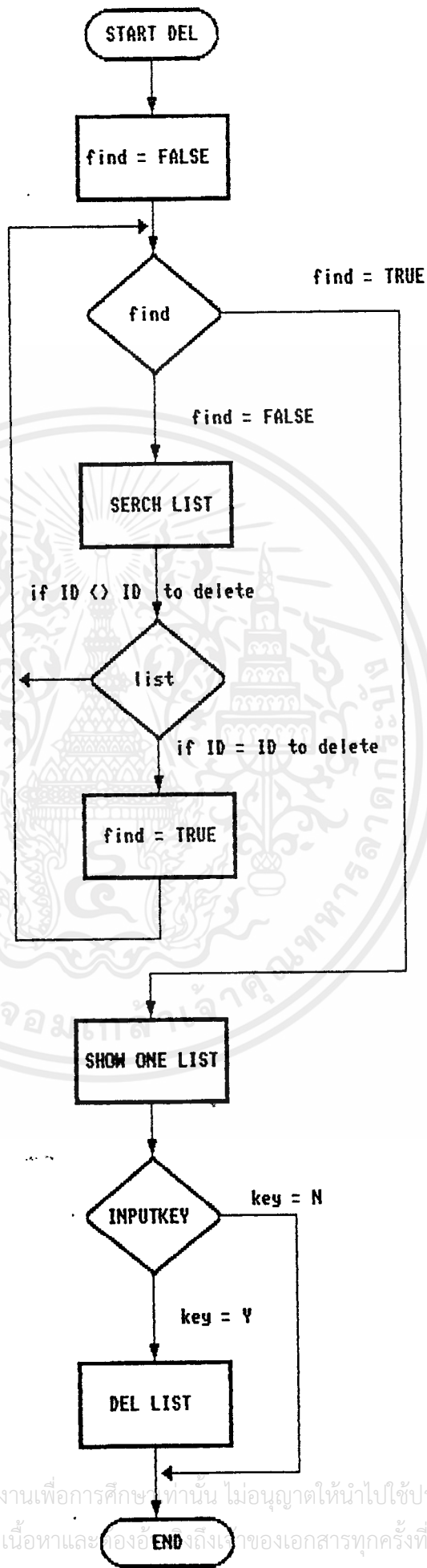
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรรขงงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ใช้ประโยชน์ด้านการค้า  
 ไม่สามารถได้ ทงสน อีกรึงห้ามมิให้ดัดแปลงเนื้อหาแลตั้งองอองถึงเจ้าของเอกสารกรรขงงานนี้ไปใช้



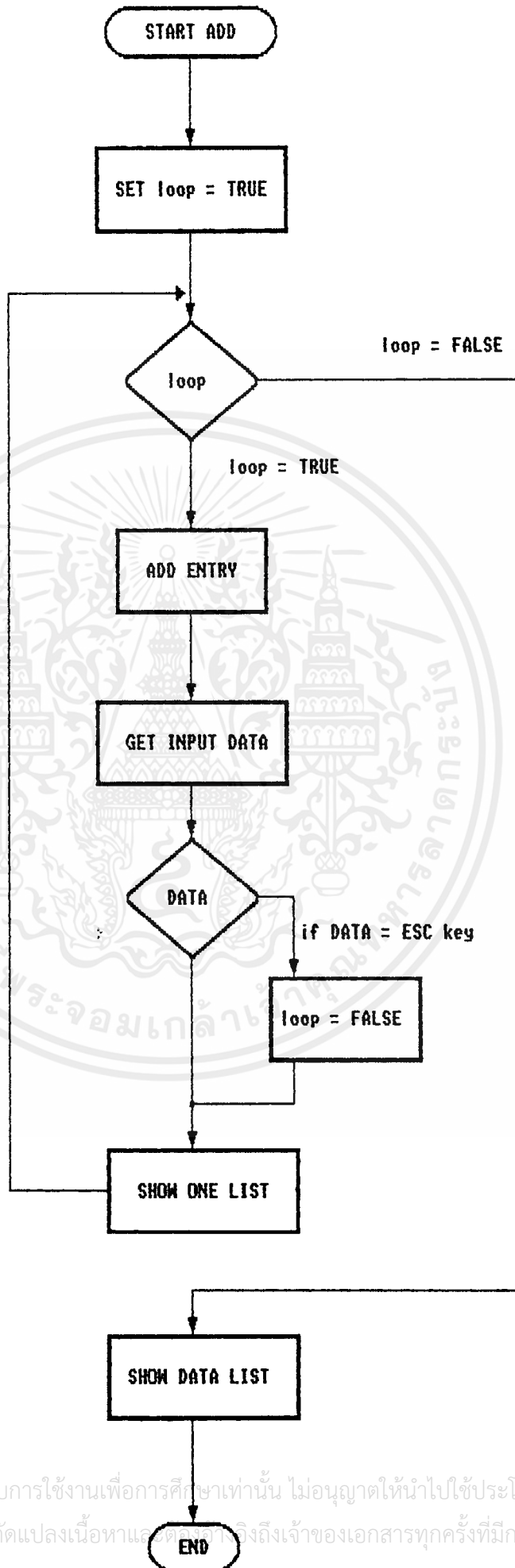
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ในองค์กรที่ออกเอกสารเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและองค์ประกอบของเอกสารทุกครั้งที่มีการนำไปใช้



## บทที่ 4

### บทสรุปและข้อเสนอแนะ

#### 4.1 บทสรุป

จากการทำโครงการนี้ ทางผู้จัดทำหวังอย่างยิ่งที่จะจัดทำโครงการพิเศษนี้ให้สามารถนำไปใช้งาน ได้จริงและสามารถอำนวยความสะดวกแก่ผู้ใช้งานมากที่สุด โดยมีลักษณะการใช้งานคร่าว ๆ ดังนี้

1. เป็นโปรแกรมที่สามารถวางแผนงานเป็น BAR CHART และสามารถแสดงผลสรุปเป็น BAR CHART และ SUMMARY SHEET ได้
2. สามารถติดตามผลงานเปรียบเทียบกับที่วางแผนไว้ได้ ทั้งยังง่ายแก่ผู้ที่คุ้นเคยกับการควบคุมงานก่อสร้างเนื่องจากมีลักษณะคล้าย การปรับปรุงผลของ MONTHLY REPORT
3. การใช้งานง่ายแก่ผู้ที่ไม่เคยใช้โปรแกรมมาก่อนเลยเนื่องจากมี HELP อยู่บน COMMAND MENU และเลือกฟังก์ชันที่ต้องการได้ โดยการเลื่อนคีย์ลูกศร ( ARROW KEYS ) ไปยังฟังก์ชันที่ต้องการแล้วกดคีย์ ENTER ในกรณีที่กดคีย์ผิดพลาดผู้ใช้สามารถกลับไปยัง MENU ก่อนหน้านี้ด้วยการกดปุ่ม ESC ที่อยู่ด้านซ้ายบนสุด
4. การป้อนข้อมูลและแก้ไขข้อมูลของโปรแกรมเป็นการใช้โครงสร้างข้อมูลแบบ DUOBLE LINK LIST ทำให้สามารถค้นหาข้อมูลที่ต้องการได้อย่างรวดเร็วเพราะทำงานบน MEMORY ของเครื่องโดยตรง ไม่ต้องทำการเปิดปิด ไฟล์เพื่ออ่านข้อมูลตลอดเวลา และจะไม่ทำการเปลี่ยนแปลงข้อมูลเก่าจนกว่าผู้ใช้จะทำการจัดเก็บข้อมูลลงไฟล์ ด้วยคำสั่ง SAVE
5. มีการเปิด ไฟล์ เพื่อเก็บข้อมูลลงในแผ่น disk ทำให้สามารถนำข้อมูลมาใช้ใหม่ได้อีก หรือ แก้ไขปรับปรุงข้อมูลได้ตามต้องการ
6. สามารถแสดงผลออกทางเครื่องพิมพ์ได้เพื่อความสะดวกในการติดตามผลงาน และการนำไปใช้งานในการขออนุมัติเบิกเงิน โดยการแสดงความก้าวหน้าของการทำงานเปรียบเทียบ

#### 4.2 ข้อเสนอแนะ

โปรแกรม การวางแผนงานโดยใช้ BAR CHART นี้ได้จัดทำมาถึงจุดหนึ่งซึ่งข้าพเจ้าหวังไว้อย่างยิ่งว่า โครงการพิเศษนี้จะเป็นจุดเริ่มต้นในการพัฒนาการประยุกต์ใช้คอมพิวเตอร์ในการวางแผนและควบคุมงานก่อสร้าง แต่ทั้งนี้เนื่องด้วยข้อจำกัดหลาย ๆ ด้านทำให้โครงการพิเศษนี้ยังมีส่วนที่สามารถขยายขอบเขตการใช้งานให้มีประสิทธิภาพมากขึ้นไปกว่านี้ หากมีผู้ประสานต่อก็จะเป็นประโยชน์อย่างยิ่ง

ทางผู้จัดทำมีข้อเสนอแนะบางประการที่คิดว่าน่าสนใจแก่ผู้ที่จะทำการพัฒนาต่อ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ในการทำงานแบบ BAR CHART นั้นมิได้คำนึงถึงความสัมพันธ์ หรือ ลำดับความสำคัญของงาน ที่ควรสนใจเป็นพิเศษ ซึ่งระบบการวางแผนงานแบบ CPM และ PERT มีการวิเคราะห์หลายอย่างที่ เป็นประโยชน์ต่อการวางแผนงานให้ละเอียดขึ้นอีกขั้นหนึ่งเนื่องทางผู้จัดทำได้เคยศึกษามานบ้าง และมีโปรแกรม ตัวอย่าง การคำนวณ CPM ในภาคผนวก ก. แต่เนื่องด้วยโปรแกรมตัวอย่างนี้มีการจอง MEMORY เป็นแบบ ARRAY ทำให้มีข้อจำกัดในการวิเคราะห์แผนงานได้ไม่มากนัก ( มากกว่า 40 รายการ ) เพื่อเป็น ตัวอย่าง และสามารถนำไปพัฒนาใช้งานต่อไปได้

2. ควรมีการแสดงผลเป็น วันที่ตามปฏิทิน เพื่อสะดวกในการติดตามงานซึ่งทางผู้จัดทำก็มีโปรแกรม ตัวอย่างให้แก่ผู้ที่สนใจด้วย ในภาคผนวก ก.

3. เนื่องจากในงานก่อสร้างจริงนอกจากต้องมีการวางแผนในเรื่องการเงินแล้ว ยังควรมีการจัดสรร ทรัพยากรของโครงการด้วย

4. สำหรับผู้ที่สนใจการแสดงผลออกทางเครื่องพิมพ์ทางผู้จัดทำแยกฟังก์ชันที่เกี่ยวข้องกับการพิมพ์นี้ อยู่ในไฟล์ชื่อ PRINTER.C เพื่อสะดวกแก่การปรับปรุง และการนำไปใช้

ผู้จัดทำหวังว่าข้อเสนอแนะ และ โปรแกรมตัวอย่างนี้ จะเป็นประโยชน์แก่ผู้ที่สนใจไม่มากนักน้อย เพื่อสะดวกในด้านการศึกษา หรือ พัฒนาต่อไป

## บรรณานุกรม

- พิกพ เต้าประจง , เทคนิคการบริหารโครงการโดย CPM และ PERT , สมาคมส่งเสริมเทคโนโลยี ( ไทย - ญี่ปุ่น ) , พิมพ์ครั้งที่ 2 , กรุงเทพฯ ฯ : โรงพิมพ์ภาพพิมพ์ , พ.ศ. 2534
- วิจิตร ตันเจตุสุทธิ , การวิจัยการดำเนินงาน , ภาควิชา วิศวกรรมอุตสาหกรรม คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย กรุงเทพฯ ฯ : บริษัท ซีเอ็ดยูเคชั่น จก.
- พต.ประพัฒน์ อุทัยภาส , โปรแกรมสำหรับการวางแผนงาน ฮาร์วาร์ดโปรเจกต์แมนเนจเมนต์ กรุงเทพฯ ฯ : บริษัท ซีเอ็ดยูเคชั่น จก. , พ.ศ. 2521
- ประกอบ บำรุงผล , การบริหารงานก่อสร้าง , แผนกวิชาบริหารงานก่อสร้าง วิชาเขตอุเทนถวาย กรุงเทพฯ ฯ : สำนักพิมพ์ศูนย์ส่งเสริมวิชาการ
- James D. Stevens , Techniques for Construction Network Scheduling , Department of civil Engineering University of Kentucky : McGraw - Hill Publishing Company
- Meredith Gibbs , The Managment of Operations 2<sup>nd</sup> edition , : Wiley ( p 361-391 )
- Jack Purdum , C Programmer 's Toolkit : Que Corp.
- ไม่ปรากฏชื่อผู้แต่ง , Borland C++ 2.0 Library Reference , USA. : Borland International , 1991.



ภาคผนวก ก.

โปรแกรมตัวอย่างสำหรับผู้สนใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตัวอย่างโปรแกรมปฏิทินการหาวันเดือนปี

```

#include <dos.h>

int Get_Date(void)
{
    struct date d;

    getdate(&d);          /* ฟังก์ชันนี้มีในภาษา C อยู่แล้วเรียกใช้ตามตัวอย่างนี้ก่อนใช้ ต้องเรียก dos.h ก่อน */
    printf("The current year is: %d\n",d.da_year);
    printf("The current day is: %d\n",d.da_day);
    printf("The current month is: %d\n",d.da_mon);
}

void Calendar()          /* ฟังก์ชันนี้ใช้การคำนวณทางคณิตศาสตร์สามารถใช้ได้โดยไม่มีเงื่อนไขใด ๆ ทั้งสิ้น */
{
    struct date d;
    int month[12] = {{31},{28},{31},{30},{31},{30},{31},{31},{30},{31},{30},{31}};
    int y,m;
    int bias,k,week,day,sum;

    printf("Enter year and month to find\n");
    scanf("%d %d",&y,&m);
    if(((y % 4 == 0) && (y % 100 != 0)) || (y % 400 == 0))
        month[1] = 29;
    bias = y + ((y - 1) / 4) - ((y - 1) / 100) + ((y - 1) / 400);
    sum = 0;
    for (k = 0;k < m-1 ;k++)
        sum = sum + month[k];
    week = (bias+ sum) % 7;
    printf(" sun mon tue wed thu fri sat\n");
    for(k = 1;k <= week*4; k++)
        printf(" ");
    for(day = 1; day <= month[m-1];day++)
    {
        printf("%4d",day);
        if ((week+day) % 7 == 0)
            printf("\n");
    }
}

```

## โปรแกรมตัวอย่างการหา Critical path

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>
#include <dos.h>

#define MAXNODE 5

int duration[MAXNODE][MAXNODE] = {{7,10,6,0,0},
                                     {0,-1,0,0,15},
                                     {0,0,0,7,0},
                                     {0,0,0,8,15},
                                     {0,0,0,0,9}};

/*-----ACTIVITY RECORD FOR FILL IN FILE -----*/
typedef struct {
    char *title;
    char *leader;
    struct date begintime; /*-----Start time of project-----*/
    int times; /*-----Unit time of project-----*/
    path; /*-----Total path in project-----*/
    char *unit_cost;
}PROJECT_DES;

#define FILENAME "temp.pln"

void SaveCriti_path(int *metric,int maxnode)
{
    FILE *file;
    PROJECT_DES project;
    ACTREC record;
    int i,j;

    if((file = fopen(FILENAME,"w+b")) != NULL)
    {
        for(i = 0;i < maxnode;i++)
            for(j = 0;j < maxnode;j++)
                if(duration[i][j] != 0){
                    duration[i][j] = (duration[i][j] == -1) ? 0 : duration[i][j];
                    record.begin = i;
                    record.end = j+1;
                    record.duration = duration[i][j];
                    record.ES = *(metric + (i * (maxnode+2)) + i + 1);
                }
    }
}
```

ภาคผนวก ก.

```
record.EC = *(metric + (i * (maxnode+2)) + j + 2);
record.LS = *(metric + ((j+2) * (maxnode+2)) + i);
record.LC = *(metric + ((j+2) * (maxnode+2)) + j + 1);
record.TF = *(metric + ((j+2) * (maxnode+2)) + j + 1) - *(metric + (i *
(maxnode+2)) + i + 1) - duration[i][j];
record.FF = *(metric + ((j+1) * (maxnode+2)) + j + 2) - *(metric + (i *
(maxnode+2)) + i + 1) - duration[i][j];
printf("begin= %d end= %d duration= %2d ",
record.begin,record.end,record.duration);
printf("ES= %2d EC= %2d LS= %2d LC = %2d TF= %2d FF= %2d",
record.ES,record.EC,record.LS,record.LC,record.TF,record.FF);
fwrite(&record,sizeof(ACTREC),1,file);
}
fclose(file);
}

void Find_criti()
{
int i,j,k;
int ES[MAXNODE+2][MAXNODE+2];

/*-----clear all matric-----*/
for(i = 0;i < MAXNODE+2;i++)
for(j = 0;j < MAXNODE+2;j++)
{
ES[i][j] = 0;
}
/*-----*/
for(j = 1;j < MAXNODE+1;j++)
for(i = 0;i < MAXNODE;i++)
{
ES[0][0] = 0;
if( duration[i][j-1] == 0)
if(i < MAXNODE-1)
i = i+1;
if( duration[i][j-1] != 0)
for(k = j;k < MAXNODE+2;k++)
{
if(duration[i][j-1] > 0 )
ES[i][j+1] = max(ES[k][i+1] + duration[i][j-1],ES[i][j+1]);
if(duration[i][j-1] == -1)
ES[i][j+1] = max(ES[k][i+1],ES[i][j+1]);
ES[k][j+1] = max(ES[i][j+1],ES[k][j+1]);
}
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

```
for(i = 0;i < MAXNODE+2;i++)
    for(j = 0;j < MAXNODE+2;j++)
        if(i >= j)
            ES[i][j] = 0;

    clrscr();
/*-----*/
for(i = 0;i < MAXNODE;i++)
    ES[MAXNODE+1][MAXNODE] = max(ES[i][MAXNODE+1],
ES[MAXNODE+1][MAXNODE]);
for(j = MAXNODE;j >= 1;j--)
    for(i = MAXNODE-1;i >= 0;i--)
    {
        if( duration[i][j-1] == 0)
            if(i > 0 )
                i = i-1;
            if( duration[i][j-1] > 0)
            {
                if( i < j ) {
                    if(ES[i+1][i] != 0)
                        ES[i+1][i] = min(ES[j+1][j] - duration[i][j-1],ES[i+1][i]);
                    if(ES[i+1][i] == 0)
                        ES[i+1][i] = min(ES[j+1][j] - duration[i][j-1],ES[j][j+1]);
                }
                ES[j+1][i]=ES[j+1][j] - duration[i][j-1];
            }
        if( duration[i][j-1] == -1)
        {
            if( i < j ) {
                if(ES[i+1][i] != 0)
                    ES[i+1][i] = min(ES[j+1][j] ,ES[i+1][i]);
                if(ES[i+1][i] == 0)
                    ES[i+1][i] = min(ES[j+1][j] ,ES[j][j+1]);
            }
            ES[j+1][i]=ES[j+1][j] ;
        }
    }
    }
printf("\n\n");
for(i = 0;i < MAXNODE+2;i++)
    {
        for(j = 0;j < MAXNODE+2;j++)
            printf("%3d ",ES[i][j]);
        printf("\n");
    }
SaveCriti_path(*ES,MAXNODE);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข.

สถิติความสามารถในการทำงานก่อสร้าง ต่อ 1 วัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สถิติความสามารถในการทำงานก่อสร้าง ต่อ 1 วัน

สถิติต่อไปนี้ แสดงถึงความสามารถในการทำงานก่อสร้างของคน 1 คนหรือคน 1 กลุ่ม (GANG) ทำงานได้ ต่อ 1 วัน (8 ชม.)

	งาน	จำนวนคน	จำนวนที่ทำได้จริง		หมายเหตุ
			จำนวน	หน่วย	
<b>1</b>	<b>งานดินและการขนย้ายวัสดุทั่วไป</b>				
	1.1 ขุดดิน	1	2.2	ม <sup>3</sup> .	
	1.2 เกี่ยยดิน	1	60	ม <sup>3</sup> .	
	1.3 ถมดิน	1	3	ม <sup>3</sup> .	
	1.4 โยกหินลงรถ	1	12-14	ม <sup>3</sup> .	
	1.5 ขุดหลุมขนาด 0.80 x 0.80 x 1.0	1	2.5	หลุม	
	1.6 โยกทรายลงจากรถ	1	40-50	ม <sup>3</sup> .	
	1.7 ขุดร่องฝังท่อ 0.66 x 0.6	1	8	ม <sup>3</sup> .	
	1.8 โยกหินขึ้นรถ	2	18-20	ม <sup>3</sup> .	
	1.9 รื้อทุบทิ้งตึกเก่า	1	3-4	ม <sup>3</sup> .	
<b>2</b>	<b>งานตอกเข็ม</b>				
	2.1 เข็มไม้ 4" x 4.00 ม.	6	90	ต้น	
	2.2 เข็มไม้ 5" x 5.00 ม.	6	30	ต้น	ตอกด้วยแรงคน
	2.3 เข็มไม้ 6" x 6.00 ม.	8	25	ต้น	
	2.4 เข็มปลายเข็ม	1	30-45	ต้น	
	2.5 ตอกเข็ม 21 ม.	-	5-8	ต้น	ตอกด้วยเครื่อง
<b>3</b>	<b>งานช่างปูน</b>				
	3.1 ทำไม้แบบคอนกรีต	1	6	ม <sup>2</sup> .	
	3.2 ก่ออิฐอมอญครึ่งแผ่น	1	1,100	แผ่น	ในที่ต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	งาน	จำนวนคน	จำนวนที่ทำได้		หมายเหตุ
			จำนวน	หน่วย	
<b>๓</b>	<b>งานช่างปูน( ต่อ )</b>				
	3.3 ก่ออิฐมอญครึ่งแผ่น	1	500-800	แผ่น	ในที่สูง
	3.4 ก่ออิฐบล็อก	1	50-80	ก้อน	
	3.5 ก่ออิฐประดับ บ.บ.ท.	1	50-70	ก้อน	
	3.6 ก่อกำแพงหิน	2	1.2-2	ม <sup>2</sup> .	
	3.7 ฉาบปูนผนัง	1	8-10	ม <sup>2</sup> .	
	3.8 เทปูนทรายหนา 5 ซม.	1	10-12	ม <sup>2</sup> .	
	3.9 ผสมคอนกรีตด้วยกระบะ	3	4-5	ม <sup>3</sup> .	
	3.10 ผสมคอนกรีตด้วยเครื่อง	4	2-25	ม <sup>3</sup> .	
	3.11 เทพื้นคอนกรีต และ อัดมัน	1	7-10	ม <sup>2</sup> .	
	3.12 เทพื้นคอนกรีต	1	1-1.5	ม <sup>3</sup> .	
	3.13 เทคอนกรีตเสา	2	1-1.2	ม <sup>3</sup> .	
	3.14 ทำพื้นหินขัด	2	15-20	ม <sup>2</sup> .	
	3.15 ปูพื้นกระเบื้องโมเสก	1	2-3	ม <sup>2</sup> .	
	3.16 กรูกระเบื้องเคลือบผนัง	1	3	ม <sup>2</sup> .	
	3.17 ทำหินล้างผนัง	2	10-14	ม <sup>2</sup> .	
	3.18 บูผนังหินอ่อน	2	3-8	ม <sup>2</sup> .	
	3.19 ขัดพื้นหินขัดด้วยแรงคน	1	3-4	ม <sup>2</sup> .	
<b>4</b>	<b>งานช่างไม้</b>				
	4.1 ตีฝ้าไม้ทับเกล็ด	1	6-8	ม <sup>2</sup> .	
	4.2 ตีฝ้าไม้ข้างลิ้น	1	4	ม <sup>2</sup> .	
	4.3 ทำฝ้าเพดานไม้	1	6	ม <sup>2</sup> .	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	งาน	จำนวนคน	จำนวนที่ทำได้		หมายเหตุ
			จำนวน	หน่วย	
4	งานช่างไม้ ( ต่อ )				
	4.4 ตั้งวงกบ	2	8-10	ช่อง	
	4.5 ท้าววงกบ	1	2	ช่อง	
	4.6 ใสพื้น ไม้	1	20	ม <sup>2</sup> .	
	4.7 ใสไม้ฝา	1	30	ม <sup>2</sup> .	
	4.8 ปูพื้น ไม้เข้าลิ้น	1	4-5	ม <sup>2</sup> .	
	4.9 ปูพื้น ไม้ตีขีด	1	10-12	ม <sup>2</sup> .	
	4.10 ปูพื้นปาร์เกต์	1	4-5	ม <sup>2</sup> .	
	4.11 วางตง-คานไม้	3	20	ม <sup>2</sup> .	
	4.12 ตีไม้ระแนงระดับพื้น	1	30-45	ม <sup>2</sup> .	
	4.13 ตีระแนงชายคา	2	20	ม <sup>2</sup> .	
	4.14 ทำบันได ไม้แบบพุก	2	2.5	ม <sup>2</sup> .	
	4.15 ทำบันได ไม้แบบเจาะ	2	3	ม <sup>2</sup> .	
	4.16 ติดตั้งอุปกรณ์ประตูหน้าต่าง	1	9-11	ช่อง	
	4.17 ทำคร่าตีไม้อัด 2 หน้า	2	15-18	ม <sup>2</sup> .	
	4.18 ทำคร่าและตีฝ้าเพดาน	3	16	ม <sup>2</sup> .	
	4.19 ทำโครงหลังคา มุงสังกะสี	6	15-20	ม <sup>2</sup> .	
	4.20 ทำโครงหลังคา มุงกระเบื้อง	6	15	ม <sup>2</sup> .	
	4.21 มุงกระเบื้องลอนคู่	1	6-8	ม <sup>2</sup> .	
	4.22 มุงสังกะสี	1	12-18	ม <sup>2</sup> .	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	งาน	จำนวนคน	จำนวนที่ทำได้		หมายเหตุ
			จำนวน	หน่วย	
<b>5</b>	<b>งานเหล็ก</b>				
	5.1 คัดเหล็ก ผูกโครงเหล็ก	2	45-60	กก.	
	5.2 ทำเหล็กลูกกรงบันได	2	15-18	ม.	
	5.3 ทำเหล็กลูกกรงหน้าต่าง	1	2-3	ช่อง	
<b>6</b>	<b>งานช่างทาสี</b>				
	6.1 ทาสีน้ำมัน	1	15-22	ม <sup>2</sup> .	
	6.2 ทาสีพาดาน	1	10	ม <sup>2</sup> .	
	6.3 ทาสีน้ำมัน	1	30	ม <sup>2</sup> .	
	6.4 ทาเซลลิ่ง 1 ครั้ง	1	20-30	ม <sup>2</sup> .	
	6.5 ผูกนั่งร้านไม้ไผ่	3	200-300	ม <sup>2</sup> .	
<b>7</b>	<b>งานสุขภัณฑ์</b>				
	7.1 ติดอ่างล้างหน้า	2	10	ที่	
	7.2 ติดตั้งที่ปัสสาวะ	2	5	ที่	
	7.3 ติดตั้งส้วมนั่งรบ	1	1-2	ที่	
	7.4 ติดตั้งส้วมนั่งยอง	1	3	ที่	
<b>8</b>	<b>งานไฟฟ้า</b>				
	8.1 เดินสายไฟ	2	6-10	จุด	
	8.2 ติดดวงโคม	1	8	จุด	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ค.

**โปรแกรมการวางแผนงานและติดตามผลงานระบบ BAR CHART**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1 #include <stdio.h>
#include <alloc.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>
#include <dir.h>
#include <stdarg.h>
#include <dos.h>
#include <ctype.h>
#include <errno.h>

#define ERROREND -1
extern int errno;
extern int Addr;

#define FALSE 0
#define TRUE !FALSE
#define CRITICALCOLOR RED
#define NORMCOLOR LIGHTGRAY
#define DATECOLOR BLACK+LIGHTGRAY
#define File_Name "noname"

typedef struct {
    char title[80];
    char project[40];
    char location[40];
    char architect[40];
    char engineer[40];
    char owner[40];
    char period[40];
    struct date day; /*-----Start time of project-----*/
    int path; /*-----Total apth in project-----*/
    char unit_cost[10];
}HEAD;

typedef struct entry{
    char Jobname[15];
    int ID,
        duration;
    struct date start,
        finish;
    int pre_entry;
    double M_cost,
        L_cost;
    int ow,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        complete;
    struct date begin,
        end;
    struct entry *preceed,*next;
}ENTRY;

```

```

typedef struct entry_rec{
    char Jobname[15];
    int ID,
        duration;
    struct date start,
        finish;
    int pre_entry;
    double M_cost,
        L_cost;
    int now,
        complete;
    struct date begin,
        end;
}ENTRY_REC;

```

```

/*=====COLOR
<FOREGROUND> F_BLACK , F_BLUE , F_GREEN , F_CYAN , F_RED
               F_MAGENTA, F_BROWN , F_LIGHTGRAY,F_DARKGRAY,
               F_LIGHTBLUE, F_LIGHTGREEN ,F_LIGHTCYAN,
               F_CYAN|F_INTENF_LIGHTRED, F_LIGHTMAGENTA ,
               F_YELLOW , F_WHITE ,F_LIGHTGRAY|F_INTEN
               F_INTEN .
<BACKGROUND> B_BLACK , B_BLUE , B_GREEN , B_CYAN , B_RED
               0x40
               B_MAGENTA, B_BROWN , B_LIGHTGRAY. MONOCHROME:.
               NORMAL , INTEN , REVERST , UNDER.
=====*/
/**** MONOCHROME *****/

```

```

#define NORMAL 7
#define INTEN 0xf
#define REVERSE 0x70
#define UNDERLINE 1
#define SINGLE 1
#define DOUBLE 2
#define XSINGLE 3
#define XDOUBLE 4
#define HEADSINGLE 4
#define HEADDDOUBLE 3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

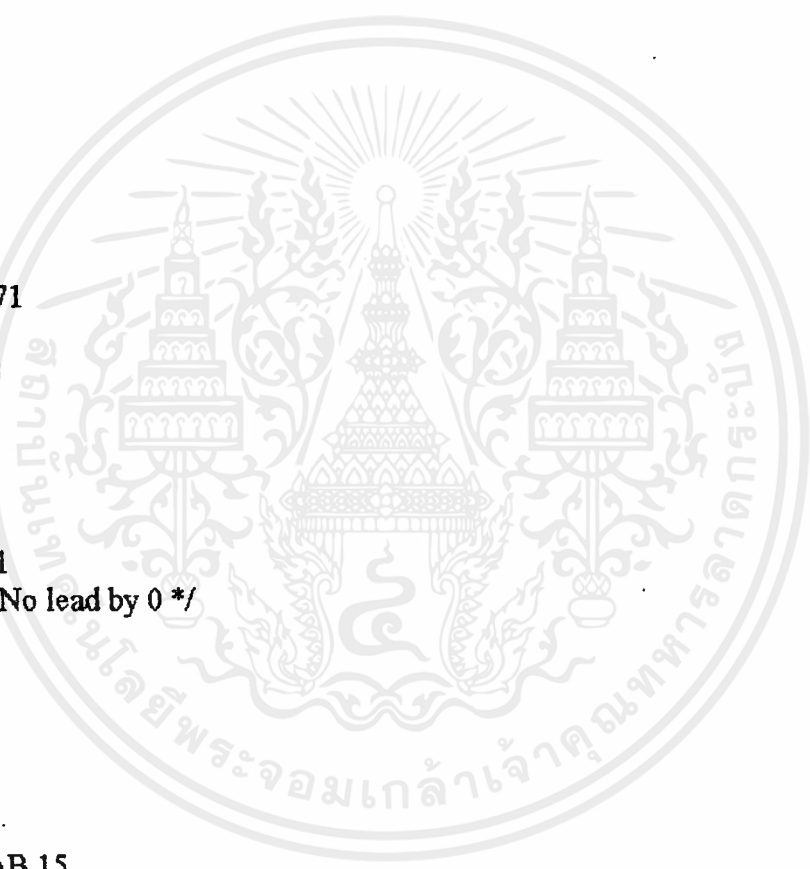
```

/*****/
#define B_BLACK 0
#define B_BLUE 0x10
#define B_GREEN 0x20
#define B_CYAN 0x30
#define B_RED 0x40
#define B_MAGENTA 0x50
#define B_BROWN 0x60
#define B_LIGHTGRAY 0x70
#define F_BLACK 0
#define F_BLUE 1
#define F_GREEN 2
#define F_CYAN 3
#define F_RED 4
#define F_MAGENTA 5
#define F_BROWN 6
#define F_LIGHTGRAY 7
#define F_INTEN 8
#define F_DARKGRAY F_BLACK|F_INTEN
#define F_LIGHTBLUE F_BLUE|F_INTEN
#define F_LIGHTGREEN F_GREEN|F_INTEN
#define F_LIGHTCYAN F_CYAN|F_INTEN
#define F_LIGHTRED F_RED|F_INTEN
#define F_LIGHTMAGENTA F_MAGENTA|F_INTEN
#define F_YELLOW F_BROWN|F_INTEN
#define F_WHITE F_LIGHTGRAY|F_INTEN
/* KEYCODE Lead-in byte is 0, follow by...*/
#define AltQ 16
#define AltW 17
#define AltE 18
#define AltR 19
#define AltT 20
#define AltY 21
#define AltU 22
#define AltI 23
#define AltO 24
#define AltP 25
#define AltA 30
#define AltS 31
#define AltD 32
#define AltF 33
#define AltG 34
#define AltH 35
#define AltJ 36
#define AltK 37

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define AltL 38
#define AltZ 44
#define AltX 45
#define AltC 46
#define AltV 47
#define AltB 48
#define AltN 49
#define AltM 50
#define F1 59
#define F2 60
#define F3 61
#define F4 62
#define F5 63
#define F6 64
#define F7 65
#define F8 66
#define F9 67
#define F10 68
#define HOME 71
#define UP 72
#define P_UP 73
#define LE 75
#define RI 77
#define END 79
#define DN 80
#define P_DN 81
/* KEYCODE :-No lead by 0 */
#define INS 82
#define DEL 83
#define CR 13
#define ESC 27
#define SPC 32
#define TAB 9
#define SHF_TAB 15
#define BSP 8
#define C_P_UP 132
#define C_P_DN 118
#define SCRL 0x46
#define ALT_1 120
/* Key define */
#define kESC 299
#define kLEFT 300
#define kRIGHT 301
#define kUP 302
#define kDOWN 303
#define kF1 304
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define kF2 305
#define kF3 306
#define kF4 307
#define kF10 310
#define kPGUP 311
#define kPGDN 312
#define kHOME 313
#define kCR 314
#define kNS 315
#define kDEL 316
#define kBSP 317
#define kSPC 318
#define kALTX 319
#define kALTQ 320
#define kTAB 321
#define CURSOROFF 0x2000 /*----- Cursor OFF ----- */
#define CURSORON 0x0607 /* ----- Cursor ON ----- */

/*-----define fuction in PRINTER.C -----*/
void print_buf(char *buf,int startx,int starty,int endx, endy,int pagex,int pagey,int density);
void print_scr(int startx,int starty,int endx, endy,int pagex,int pagey,int density);
void print_scr1(int startx,int starty,int endx, endy,int pagex,int pagey,int density);
void putbuf(char *buf,int x,int y,int val);
void getbuf(char *buf,int x,int y);
void set_graphics(int cols,int density);
void reset(void);
void print(char ch);
void rambox(char *buf,int startx,int starty,int endx, endy,int color);
void print_graph();
void print_graph1();
/*-----define fuction in Q_FILE -----*/
void KeepDescrib(HEAD *project);
void SaveDescrib(HEAD *p);
void GetDescrib();
void Plot_sum();
void Plot_actual();
void Plot(int border,int v,int scale,int length_y,int dif_ID,int starty);
void Plot_print(int border,int v,int scale,int length_y,int dif_ID,int starty,double maxs);
void Plot_real(int border,int v,int scale,int length_y,int dif_ID,int starty);
void Plot_rprint(int border,int v,int scale,int length_y,int dif_ID,int starty,double maxs);
void freealloc(double far *ptr);
void PreChart();
void Chart_command();
int gprintf(int x,int y,char *fmt,...);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void opengraphic();
void showday(int z,int j,intx,int y,int maxx);
void Summary();
void Chart_print1();
void Chart_print();
/*-----define fuction in Q_INPUT -----*/
void Beep();
int InputKey();
int GetInt(int degit);
int Getdouble(int degit);
int getstring(int length,char *str);
void get_date(struct date *d,char *t,int date_form);
void AddEntry(ENTRY **h,ENTRY **t);
void DelEntry(ENTRY *del);
void onellst(ENTRY *n);
void FreeList(ENTRY **h);
void Showlist(ENTRY *h);
void Savelist(ENTRY *h);
void Add_command();
void Del_command();
void Insert_command();
void Modify_command();
void ShowACTUAL(ENTRY *h);
void Add_ACTUAL();
void Del_ACTUAL();
void Insert_ACTUAL();
void Modify_ACTUAL();
void Edit_Project();
/*-----define fuction in Q_SUM -----*/
void Add_sum(SUM_LIST **h,SUM_LIST **t);
void Free_sum(SUM_LIST **h);
void Total_sum();
void Plot_sumthing();
unsigned int tr(struct date *n);
void show();
/*-----define fuction in Q_MENU -----*/
void setcursor(unsigned int shape);
void menu_show(MENU m[],int maxmenu);
void menu_rewrite(int x,int y,MENU m[],char attr,int count);
void menu_descrip(char *p);
void menu_hllite(MENU m[],int maxmenu,int count);
void Exit_menu();
void Dcrib_command();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Plan_menu();
void Actual_menu();
void Edit_menu();
void Load_command();
void New_dir();
void File_menu();
void New_menu();
void Main_menu();

/*----- define extern variable -----*/

#ifndef MAIN
extern char name_file[40];
extern ENTRY far *First_node;
extern ENTRY far *Last_node;
extern int now_x;
extern int now_y;
extern int PLAN;
extern int ACTUAL;
extern SUM_LIST *First_sum;
extern int max_day;
extern int max_actual;
extern HEAD *Project;
#endif
```

```

***** This function print *****/
#define SINGLE 0
#define DOUBLE 1
#define QUAD 2
#define TURBOC

void print(char ch);
unsigned char getbuf(char *buf,int x,int y);
void set_graphics(int cols,int density);

void print_buf(char *buf,int startx,int starty,int endx,int endy,int pagex,int pagey,int density)
{
register int i,x,y,px;
int cols,color,sum;

    endx++;
    endy++;
    cols = endx - startx;
    for(;pagey >= 0;pagey--)
        print('\n');
    for(y = starty;y <= endy;y+=8) {
        for(px = 0;px < pagex;px++)
            print(' ');
        set_graphics(cols,density);
        for(x = startx;x<endx;x++) {
            sum = 0;
            for(i = 0;i < 8;i++) {
                if( y+i < endy) {
                    color = getbuf(buf,x,y+i);
                    if(color) sum += 1<<(7-i);
                }
            }
            print(sum);
        }
        print('\n');
    }
}

void print_scr(int startx,int starty,int endx,int endy,int pagex,int pagey,int density)
{
register int i,x,y,px;
int cols,color,sum;

    endx++;
    endy++;
    cols = endy - starty;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    for(;pagey >= 0;pagey--)
        print("\n");
    for(x = startx,x <= endx,x+=8) {
        for(px = 0;px <pagex;px++)
            print(' ');
        set_graphics(cols,density);
        for(y = endy ;y > starty - 1;y--) {
            sum = 0;
            for(i = 0;i < 8 ;i++) {
                if( x+i < endx) {
                    color = getpixel(x+i,y);
                    if(color) sum += 1<<(7-i);
                }
            }
            print(sum);
        }
        print("\n");
    }
}

void print_scr1(int startx,int starty,int endx,int endy,int pagex,int pagey,int density)
{
    register int i,x,y,px;
    int cols,color,sum;

    endx++;
    endy++;
    cols = endx - startx;
    for(;pagey >= 0;pagey--)
        print("\n");
    for(y = starty;y <= endy,y+=8) {
        for(px = 0;px <pagex;px++)
            print(' ');
        set_graphics(cols,density);
        for(x = startx;x<endx,x++) {
            sum = 0;
            for(i = 0;i < 8;i++) {
                if( y+i < endy) {
                    color = getpixel(x,y+i);
                    if(color) sum += 1<<(7-i);
                }
            }
            print(sum);
        }
        print("\n");
    }
}

/***** Write a value to the specified buffer at specified location. *****/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void putbuf(char *buf,int x,int y,int val)
{
    buf += y*60;
    buf += x/8;
    val &= 1;
    if(val)
        *buf |= val << ( x % 8 );
    else
        *buf &= val << (x % 8);
}

```

```

unsigned char getbuf(char *buf,int x,int y)
{
    unsigned char mask;

    mask = 1;
    buf += y*60;
    buf += x/8;
    mask <= x % 8;
    return(*buf & mask);
}

```

```

void set_graphics(int cols,int density)
{
    union{
        unsigned char c[2];
        unsigned int i;
    } u;
    char den_code;

    u.i = cols;
    switch(density) {
        case SINGLE : den_code = 75;break;
        case DOUBLE : den_code = 76;break;
        case QUAD : den_code = 90;break;
    }
    print(27);print(65);print(8);          /****** 8/72 line spacing *****/
    print(27); print(den_code); print(u.c[0]);print(u.c[1]);
}

```

```

void reset(void)
{
    print(27);print(64);
}
/*----- Send a character to printer.-----*/

```

```

void print(char ch)
{
#ifdef TURBOC
    biosprint(0,ch,0);
#endif
#ifdef MICROSOFTC
    _bios_printer(0,0,ch);
#endif
}          /*----- Construct a box in RAM -----*/

void rambox(char *buf,int startx,int starty,int endx,int endy,int color)
{
register int x,y;

    for(x= startx;x < endx;x++)
        putbuf(buf,x,endy,color);
    for(y= starty;y < endy;y++)
        putbuf(buf,endx,y,color);
    for(x= startx;x < endx;x++)
        putbuf(buf,x,starty,color);
    for(y= starty;y < endy;y++)
        putbuf(buf,startx,y,color);
}

void print_graph ()
{
int maxx,maxy;

    maxx=getmaxx();
    maxy=getmaxy();
    print_scr(0,0,maxx,maxy,0,5,DOUBLE);
    reset();
}          /****** print screen Landscape *****/

void print_graph1()
{
int maxx,maxy;

    maxx=getmaxx();
    maxy=getmaxy();
    print_scr1(0,0,maxx,maxy,0,5,DOUBLE);
    reset();
}          /****** print screen Portrait *****/

```

```

#define MAIN
#include <conio.h>
#include <dos.h>
#include "lam.h"

#define OFF 0
#define ON !OFF

/*----- extern variable-----*/

ENTRY *First_node = NULL;
ENTRY *Last_node = NULL;
SUM_LIST *First_sum = NULL;
HEAD *Project = NULL;
extern int menu_count = 0;
char name_file[40] = "";
int now_x = 0;
int now_y = 0;
int PLAN = ON;
int ACTUAL = OFF;

typedef struct{
    char *prompt;          /* First line menu prompt */
    int plen;              /* Its length */
    char *desc;            /* Second line menu prompt */
}MENU;

#define OPTIONS(a,b) {a,sizeof(a)-1,b},
#define OPTIONEND {0}
#define BACKMENU BLACK
#define MENUCOLOR WHITE
#define FALSE 0
#define TRUE !FALSE
MENU firstmenu[] = {
    OPTIONS("New ", "Begin new operation")
    OPTIONS("File ", "Utility with file")
    OPTIONS("Chdir ", "New current data path")
    OPTIONS("Exit ", "Terminate this program")
    OPTIONEND
};
MENU filemenu[] = {
    OPTIONS("Load ", "Load project file form disk")
    OPTIONS("Save ", "Keep current project on disk")
    OPTIONS("Delete", "Delete file")
    OPTIONEND
};

```

```

MENU editmenu[] = {
    OPTIONS("Plan ","Edit plan")
    OPTIONS("Actual ","Edit job")
    OPTIONS("Describ ","Chang current data description")
    OPTIONS("Chart ","Show data on Bar-chart ")
    OPTIONS("Summary ","Show data on Summary-chart ")
    OPTIONEND
};

MENU planmenu[] = {
    OPTIONS("Add ","Add a new job to the list")
    OPTIONS("Insert ","Insert a new job into the job sequence")
    OPTIONS("Erase ","Erase job")
    OPTIONS("Modify ","Modify a job's name ,duration ,etc.")
    OPTIONS("Save ","Store list on disk")
    OPTIONS("ON ","Display plan schedule")
    OPTIONS("OFF ","Undisplay plan schedule")
    OPTIONEND
};

void setcursor(unsigned int shape)
{
    union REGS reg;

    reg.h.ah = 1;
    reg.x.cx = shape;
    int86(0X10, &reg, &reg);
}

void menu_show(MENU m[],int maxmenu)
{ int i,j;
  gotoxy(1,25);
  clrscr();
  for( i = 0;i < maxmenu;i++)
    for(j = 0;j < m[i].plen;j++)
    {
      if(isupper(*(m[i].prompt + j)))
      {
        textcolor(RED);
        printf("%c",*(m[i].prompt + j));
      }
      else
      {
        textcolor(LIGHTCYAN);
        printf("%c",*(m[i].prompt + j));
      }
    }
  /*-----end for-----*/
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        gotoxy(1,25);
    }

void menu_rewrite(int x,int y,MENU m[],char attr,int count)
{
int p;
    gotoxy(x,y);
    for( p = 0;p < m[count].plen;p++)
    {
        gotoxy(x+p,y);
        _AH = 8;
        _BH = 0;
        geninterrupt(0x10);
        _AH = 9;
        _BL = attr;
        _CX = 1;
        geninterrupt(0x10);
    } /* for loop */
}

void menu_descrip(char *p)
{
int tempx,tempy;

    tempx = wherex();
    tempy = wherey();
    gotoxy(1,24);
    textattr(GREEN+LIGHTGRAY);
    clreol();
    printf(p);
    gotoxy(tempx + 1,tempy);
}

void menu_hllite(MENU m[],int maxmenu,int count)
{
int i,x,y, char attr;
struct text_info prev;
    gettextinfo(&prev);
    menu_show(m,maxmenu);
    attr = WHITE + (MENUCOLOR << 4);
    x = 1;
    if(count > 0)
        for(i=0;i < count;i++)
            x = x + m[i].plen;
    if(count == maxmenu)
        x = 1;
    menu_rewrite(x,25,m,attr,count);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    menu_descrip(m[count].desc);
    textattr(B_BLACK||F_LIGHTGRAY);
}

void Exit_menu()
{
    FreeList(First_node);
    free(Project);
    textattr(B_BLACK+F_GREEN);
    clrscr();
    printf("Good bye ");
    textattr(0x07);
    printf("\n");
    setcursor(CURSORON);
    exit(0);
}

void Dcrib_command()
{
    clrscr();
    GetDescrib(Project);
    gotoxy(17,1);
    switch(InputKey())
    {
        case kCR      : SaveDescrib(Project);break;
        case kESC     : break;
        default       : KeepDescrib(Project);SaveDescrib(Project);break;
    }
}

void Plan_menu()
{
    int loop,count;

    loop = TRUE;
    count = 0;
    do
    {
        menu_show(planmenu,7);
        menu_hilite(planmenu,7,count);
        switch(InputKey())
        {
            case kLEFT : count-- ;
                        if(count < 0)
                            count = 6;
                        break;

```

```

case kRIGHT : count++;
              if(count > 6 )
                count = 0;
              break;
case kCR : switch(count)
           {
             case 0 :Add_command();break;
             case 1 :Insert_command();break;
             case 2 :Del_command();break;
             case 3 :Modify_command();break;
             case 4 :Savelist(First_node);break;
             case 5 :PLAN = ON;break;
             case 6 :PLAN = OFF;break;
           }
           break;
case kESC : loop = FALSE;break;
           } /*-----end switch loop -----*/
}while(loop != FALSE);
}

void Actual_menu()
{
int loop,count;

loop = TRUE;
count = 0;
do
{
menu_show(planmenu,7);
menu_hilite(planmenu,7,count);
switch(InputKey())
{
case kLEFT : count-- ;
              if(count < 0)
                count = 6;
              break;
case kRIGHT : count++;
              if(count > 6 )
                count = 0;
              break;
case kCR : switch(count)
           {
             case 0 :Add_ACTUAL();break;
             case 1 :Insert_ACTUAL();break;
             case 2 :Del_ACTUAL();break;
             case 3 :Modify_ACTUAL();break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        case 4 :Savelist(First_node);break;
        case 5 :ACTUAL = ON;break;
        case 6 :ACTUAL = OFF;break;
    }
    break;
    case kESC : loop = FALSE;break;
} /*-----end switch loop -----*/
}while(loop != FALSE);
}

void Edit_menu()
{
int loop,count;

loop = TRUE;
count = 0;

do
{
menu_show(editmenu,5);
menu_hilite(editmenu,5,count);
switch(InputKey())
{
case kLEFT : count-- ;
if(count < 0)
count = 4;
break;
case kRIGHT : count++;
if(count > 4 )
count = 0;
break;
case kCR : switch(count)
{
case 0 :Plan_menu();break;
case 1 :Actual_menu();break;
case 2 :Dcrib_command();break;
case 3 :Chart_command();break;
case 4 :Summary();break;
}
break;
case kESC : loop = FALSE;break;
} /*-----end switch loop -----*/
}while(loop != FALSE);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void Load_command()
{
struct fblk fblk;
int done;
char temp[40];

gotoxy(1,1);
textattr(0x07);
clrscr();
printf("Directory listing of *.lst\n");
done = findfirst("*.lst",&fblk,FA_ARCH);
while (!done)
{
printf(" %s\n", fblk.ff_name);
done = findnext(&fblk);
}
if(done != 0)
{
gotoxy(1,22);
cprintf("Enter file name to Oparate : .lst");
gotoxy(29,22);
getstring(12,name_file);
Edit_Project();
}
}
void New_dir()
{
char path[50];

gotoxy(1,22);
printf("\nEnter new directory : ");
gets(path);
if(chdir(path))
{
perror("chdir()");
exit(1);
}
}
void File_menu()
{
int loop,count;

loop = TRUE;
count = 0;
do
{

```

```

menu_show(filemenu,3);
menu_hilite(filemenu,3,count);
switch(InputKey())
{
    case kLEFT : count-- ;
                if(count < 0)
                count = 2;
                break;
    case kRIGHT : count++;
                if(count > 2 )
                count = 0;
                break;
    case kCR : loop = FALSE;
              switch(count)
              {
                  case 0 :Load_command();break;
                  case 1 :Savelist(First_node);break;
                  case 2 :break;
              }
              break;
    case kESC : loop = FALSE;break;
              } /*-----end switch loop -----*/
}while(loop != FALSE);
; }

void New_menu()
{
    clrscr();
    printf("Enter filename donot extendion : ");
    getstring(12,name_file);
    KeepDescrib(Project);
    SaveDescrib(Project);
    clrscr();
    gotoxy(1,1);
    puts("\nList of Activity : ");
    printf("\n ID   Job name   dur   start   finish   PreReq   Material cost   Labour cost");
    now_x = wherex();
    now_y = wherey();
    Add_command();
}

void Main_menu()
{
    int loop,count;
    int maxmenu = 4;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

textattr(0x07);
clrscr();
loop = TRUE;
printf("          Project planing by computer\n\n");
printf("          developed by \n\n");
printf("          Piriya Krawkangvan      \n\n\n");
printf("          King mongkot's Institue of Technology\n ");
printf("          Ladkrabang\n ");
printf("          copyright 1993\n ");
gotoxy(1,15);
printf("          * use ARROW keys to move cursor\n");
printf("          * Press ENTER key to select a menu item\n");
printf("          * Press ESC key to exit menu \n");
printf("          * Press F10 key to print out chart or summary");
count = 0;
do
{
    menu_hilite(firstmenu,4,count);
    switch(InputKey())
    {
        case kLEFT : count-- ;
                    if(count < 0)
                        count = maxmenu - 1;
                    break;
        case kRIGHT : count++;
                    if(count > maxmenu - 1)
                        count = 0;
                    break;
        case kCR : switch(count)
                    {
                        case 0 : New_menu();break;
                        case 1 : File_menu();break;
                        case 2 : New_dir();break;
                        case 3 : Exit_menu();break;
                        /*-----EXIT PROGRAM-----*/
                    }
                    if(count<2)
                    {
                        Plan_menu();
                        Edit_menu();
                    }
                    break;
        case kESC : loop = FALSE;break;
    } /*-----end switch loop -----*/
}while (loop != FALSE);
}/* end program */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void main()
{
    Project = (HEAD *)malloc(sizeof(HEAD));
    show();
    Main_menu();
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>
#include <alloc.h>
#include <dos.h>
#include <graphics.h>
#include "lam.h"

#define CRITICALCOLOR RED
#define NORMCOLOR LIGHTGRAY
#define DATECOLOR BLACK+LIGHTGRAY
#define FALSE 0
#define TRUE !FALSE
#define OFF 0
#define ON !OFF
#define DAY 5

int gprintf(int x,int y,char *fint, ...);
void opengraphic();
void showday(int z,int j,int x,int y,int maxx);
void Chart_print();

int max_day;
int max_actual;
double far *pt = NULL;
double far *ac = NULL;
double maxsum;
double maxactual;

void KeepDescrip(HEAD *Project)
{
char str[12];

    free(Project);
    Project = (HEAD *)malloc(sizeof(HEAD));
    window(1,1,80,22);
    textattr(0x07);
    clrscr();
    window(1,1,80,25);
    gotoxy(1,1);

    printf("TITLE   :");getstring(80,Project->title);
    printf("\n");
    printf("PROJECT  :");getstring(40,Project->project);

```

```

printf("\n");
printf("LOCATION : ");getstring(40,Project->location);
printf("\n");
printf("ARCHITECT : ");getstring(40,Project->architect);
printf("\n");
printf("ENGINEER : ");getstring(40,Project->engineer);
printf("\n");
printf("OWNER   : ");getstring(40,Project->owner);
printf("\n");
printf("FOR PERIOD : ");getstring(40,Project->period);
printf("\n");
printf("PROJECT START : ");
getstring(12,str);
get_date(&Project->day,str,1);
printf("\n");
Project->path = 0;
printf("TOTAL JOB : %2d\n",Project->path);
printf("Time Scale : DAY \n");
printf("UNIT/COST : ");getstring(10,Project->unit_cost);
printf("\n");
}

```

```

void SaveDescrib(HEAD *p)

```

```

{
FILE *file;
char filename[20];

strcpy(filename,name_file);
strcat(filename,".des");
if (file = fopen(filename,"w+b")) != NULL)
{
    fwrite(p,sizeof(HEAD),1,file);
    fclose(file);
}
}

```

```

void GetDescrib()

```

```

{
FILE *file;
char filename[20];

free(Project);
Project = (HEAD *)malloc(sizeof(HEAD));
strcpy(filename,name_file);
strcat(filename,".des");

```

```

if( (file = fopen(filename,"r+b")) != NULL)
{
    fread(Project,sizeof(HEAD),1,file);
    printf("TITLE      : %s\n",Project->title);
    printf("PROJECT     : %s\n",Project->project);
    printf("LOCATION      : %s\n",Project->location);
    printf("ARCHITECT    : %s\n",Project->architect);
    printf("ENGINEER     : %s\n",Project->engineer);
    printf("OWNER        : %s\n",Project->owner);
    printf("FOR PERIOD   : %s\n",Project->period);
    printf("PROJECT START : %2d/%2d/%2d \n",
        Project->day.da_day,Project->day.da_mon,Project->day.da_year);
    printf("TOTAL JOB    : %2d\n",Project->path);
    printf("Time Scale  : DAY \n");
    printf("UNIT/COST    : %s\n",Project->unit_cost);
    fclose(file);
}
/* ----- end loop if -----*/
}

void Plot_sum()
{
    ENTRY *n;
    unsigned int x,y,i,fn,st;
    double cost_per_day;
    double far *pt1 = NULL;
    int maxx,maxy;

    pt = (double far *)fcalloc(max_day,sizeof(double));
    pt1=pt;
    for(i = 1;i <= max_day,i++)
        *pt1++ = 0;
    n = First_node;
    while(n !=NULL)
    {
        cost_per_day = (n->M_cost + n->L_cost) / n->duration;
        pt1=pt;
        st=tr(&n->start);
        fn=tr(&n->finish);
        for(x = 1; x <= fn; x++)
        {
            if(x >= st)
                *pt1 = *pt1 + cost_per_day;
            pt1++;
        }
        n = n->next;
    }
    /*-----end for while loop -----*/
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    maxsum=0;
    pt1=pt;
    for(x=0; x<max_day; x++,pt1++)
        maxsum += *pt1;
}

void Plot_actual(ENTRY *s)
{
    ENTRY *n;
    unsigned int x,y,i,fn,st;
    double cost_per_day,now;
    double far *ac1 = NULL;
    int maxx,maxy;

    ac = (double far *)farcalloc(max_actual,sizeof(double));
    ac1=ac;
    for(i = 1;i <= max_actual;i++)
        *ac1++ = 0;
    n = s;
    while(n != NULL || n->now == 0)
    {
        ac1=ac;
        st=tr(&n->begin);
        fn=tr(&n->end);
        now = (double) n->now;
        cost_per_day = ((n->M_cost + n->L_cost) * (now / 100)) / n->duration;
        for(x = 1; x <= fn; x++)
        {
            if(x >= st)
                *ac1 = *ac1 + cost_per_day;
            ac1++;
        }
        n = n->next;
    }
    maxactual=0;
    ac1=ac;
    for(x=0; x<max_actual; x++,ac1++)
        maxactual += *ac1;
}

void Plot(int border,int v,int scale,int length_y,int dif_ID,int starty)
{
    struct viewporttype vinfo;
    register int x,i;
    double far *pt1 = NULL;
    double sum=0,y = 0;

```

```

int maxx,maxy,col,z,w,j,k;
double temp_sum = 0;
int month_day[12]={31},{28},{31},{30},{31},{30},{31},{31},{30},{31},{30},{31}};

    pt1 = pt;
    maxx = getmaxx();
    maxy = getmaxy();
    x = v;
    y = maxy - 70;
    col=getcolor();
    setlinestyle(SOLID_LINE,1,1);
    rectangle(0,maxy-65,maxx,maxy-15);
    setcolor(LIGHTRED);
    w = Project->day.da_mon -1;
    z = month_day[w] - Project->day.da_day;
    temp_sum = 0;
    if(starty != 57) k = 31;
    else k=70;
    for(i = 1; i <= border + (maxx - v)/DAY && i <= max_day;i++)
    {
        sum = sum + *pt1;
        temp_sum = temp_sum + *pt1;
        y = length_y - (sum/maxsum)*scale;
        if(i == border+1)
            moveto(x,y);
        if(y > maxy-k ){
            if( i >= border)
                x += DAY;
            moveto(x,maxy - k);
        }
        if( i > border && y < maxy - k)
        {
            x += DAY;
            if(y > starty )
                lineto(x,y);
        }
        if( i >= z)
        {
            j = (z - border)*DAY + v;
            setcolor(WHITE);
            if(j > v)
                line(j+5,starty,j+5,maxy - k);
            if(j > v && dif_ID < 11)
            {
                gprintf(j-60,maxy - 50,"%-10.0f",temp_sum);
                gprintf(j-60,maxy - 30,"%-10.0f",sum);
            }
        }
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        setlinestyle(SOLID_LINE,1,1);
        line(j+5,starty,j+5,maxy - 15);
    }
    temp_sum = 0;
    w++;
    if(w > 11) w = 0;
        z = z + month_day[w];
    setcolor(LIGHTRED);
    moveto(x,y);
}
pt1++;
}
j = (z - border) * DAY + v;
setcolor(WHITE);
if(j > v && dif_ID < 11)
{
    gprintf(j-60,maxy - 50,"%-10.0f",temp_sum);
    gprintf(j-60,maxy - 30,"%-10.0f",sum);
    setlinestyle(SOLID_LINE,1,1);
    line(j+5,starty,j+5,maxy - 15);
}
setcolor(col);
}
void Plot_print(int border,int v,int scale,int length_y,int dif_ID,int starty,double maxs)
{
    struct viewporttype vinfo;
    register int x,i;
    double far *pt1 = NULL;
    double sum=0,y = 0;
    int maxx,maxy,col,z,w,j,k;
    double temp_sum = 0;
    int month_day[12]={ {31},{28},{31},{30},{31},{30},{31},{31},{30},{31},{30},{31}};

    pt1 = pt;
    maxx = getmaxx();
    maxy = getmaxy();
    x = v;
    y = maxy - 70;
    col=getcolor();
    setlinestyle(SOLID_LINE,1,1);
    if(dif_ID < 11)
    {
        setviewport(0,maxy-115,maxx,maxy-15,1);
        clearviewport();
        setviewport(0,0,maxx,maxy,1);
        rectangle(0,maxy-115,maxx,maxy-15);
    }
}

```

```

        line(5,maxy - 65,maxx-5,maxy - 65);
    }
    setcolor(LIGHTRED);
    w = Project->day.da_mon -1;
    z = month_day[w] - Project->day.da_day;
    temp_sum = 0;
    if(starty != 57) k = 31;
    else k=70;
    for(i = 1; i <= border + (maxx - v)/DAY && i <= max_day,i++)
    {
        sum = sum + *pt1;
        temp_sum = temp_sum + *pt1;
        y = length_y - (sum/maxsum)*scale;
        if(i == border+1)
            moveto(x,y);
        if(y > maxy-k ){
            if( i >= border)
                x += DAY;
            moveto(x,maxy - k);
        }
        if( i > border && y < maxy - k)
        {
            x += DAY;
            if(y > starty )
                lineto(x,y);
        }
        if( i >= z)
        {
            j = (z - border )*DAY + v;
            setcolor(WHITE);
            if(j > v)
                line(j+5,starty,j+5,maxy - k);
            if(j > v && dif_ID < 11)
            {
                gprintf(j-60,maxy - 100,"%-10.0f",temp_sum);
                gprintf(j-60,maxy - 80,"%-10.0f",sum);
                gprintf(j-60,maxy - 50,"%3.1f%%",temp_sum*100 / maxs);
                gprintf(j-60,maxy - 30,"%3.1f%%",sum*100 / maxs);
                setlinestyle(SOLID_LINE,1,1);
                line(j+5,starty,j+5,maxy - 15);
            }
            temp_sum = 0;
            w++;
            if(w > 11)
                w = 0;
            z = z + month_day[w];
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        setcolor(LIGHTRED);
        moveto(x,y);
    }
    pt1++;
}
j = (z - border ) * DAY + v;
setcolor(WHITE);
if(j > v && dif_ID < 11)
{
    gprintf(j-60,maxy - 100,"%-10.0f",temp_sum);
    gprintf(j-60,maxy - 80,"%-10.0f",sum);
    gprintf(j-60,maxy - 50,"%3.1f%%",temp_sum * 100 / maxs);
    gprintf(j-60,maxy - 30,"%3.1f%%",sum * 100 / maxs);
    setlinestyle(SOLID_LINE,1,1);
    line(j+5,starty,j+5,maxy - 15);
}
setcolor(col);
}

void Plot_real(int border,int v,int scale,int length_y,int dif_ID,int starty)
{
    struct viewporttype vinfo;
    register int x,i;
    double far *ac1;
    double sum=0,y = 0;
    int maxx,maxy,col,z,w,j,k;
    double temp_sum = 0;
    int month_day[12]={{31},{28},{31},{30},{31},{30},{31},{31},{30},{31},{30},{31}};

    ac1 = ac;
    maxx = getmaxx();
    maxy = getmaxy();
    x = v;
    y = maxy - 70;
    col=getcolor();
    setcolor(LIGHTGREEN);
    w = Project->day.da_mon - 1;
    z = month_day[w] - Project->day.da_day;
    temp_sum = 0;
    if(starty != 57) k = 31;
    else k=70;
    for(i = 1; i <= border + (maxx - v)/DAY && i <= max_actual;i++)
    {
        sum = sum + *ac1;
        temp_sum = temp_sum + *ac1;
        y = length_y - (sum/maxsum)*scale;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(i == border+1) moveto(x,y);
if(y > maxy-k){
    if( i >= border)
        x += DAY;
    moveto(x,maxy - k);
}
if(i > border && y < maxy - k)
{
    x += DAY;
    if( y > starty )
        lineto(x,y);
}
if( i >= z)
{
    j = (z - border ) * DAY + v;
    setcolor(GREEN);
    if(j > v && dif_ID < 11)
    {
        gprintf(j-60,maxy - 62,"%-10.0f",temp_sum);
        gprintf(j-60,maxy - 40,"%-10.0f",sum);
    }
    temp_sum = 0;
    w++;
    if(w > 11) w = 0;
    z = z + month_day[w];
    setcolor(LIGHTGREEN);
    moveto(x,y);
}
acl++;
}
/*----- end for loop -----*/
j = (z - border ) * DAY + v;
setcolor(GREEN);
if(j > v && dif_ID < 11)
{
    gprintf(j-60,maxy - 62,"%-10.0f",temp_sum);
    gprintf(j-60,maxy - 40,"%-10.0f",sum);
}
setcolor(col);
}

```

```

void Plot_rprint(int border,int v,int scale,int length_y,int dif_ID,int starty,double maxs)

```

```

{
    struct viewporttype vinfo;
    register int x,i;
    double far *acl;
    double sum=0,y = 0;

```

```

int maxx,maxy,col,z,w,j,k;
double temp_sum = 0;
int month_day[12]={31},{28},{31},{30},{31},{30},{31},{31},{30},{31},{30},{31}};

ac1 = ac;
maxx = getmaxx();
maxy = getmaxy();
x = v;
y = maxy - 70;
col=getcolor();
setcolor(LIGHTGREEN);
w = Project->day.da_mon -1;
z = month_day[w] - Project->day.da_day;
temp_sum = 0;
if(starty != 57) k = 31;
else k=70;
for(i = 1; i <= border + (maxx - v)/DAY && i <= max_actual;i++)
{
    sum = sum + *ac1;
    temp_sum = temp_sum + *ac1;
    y = length_y - (sum/maxsum)*scale;
    if(i == border+1) moveto(x,y);
    if(y > maxy-k ){
        if( i >= border)
            x += DAY;
        moveto(x,maxy - k);
    }
    if(i > border && y < maxy - k)
    {
        x += DAY;
        if( y > starty )
            lineto(x,y);
    }
    if( i >= z)
    {
        j = (z - border ) *DAY + v;
        setcolor(GREEN);
        if(j > v && dif_ID < 11)
        {
            gprintf(j-60,maxy - 112,"%-10.0f",temp_sum);
            gprintf(j-60,maxy - 90,"%-10.0f",sum);
            gprintf(j-60,maxy - 62,"%3.1f%%",temp_sum*100/ maxs);
            gprintf(j-60,maxy - 40,"%3.1f%%",sum*100 / maxs);
        }
        temp_sum = 0;
        w++;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if(w > 11) w = 0;
        z = z + month_day[w];
        setcolor(LIGHTGREEN);
        moveto(x,y);
    }
    ac1++;
} /* ----- end for loop -----*/
j = (z - border ) * DAY + v;
setcolor(GREEN);
if(j > v && dif_ID < 11)
{
    gprintf(j-60,maxy - 112,"%-10.0f",temp_sum);
    gprintf(j-60,maxy - 90,"%-10.0f",sum);
    gprintf(j-60,maxy - 62,"%3.1f%%",temp_sum*100 / maxs);
    gprintf(j-60,maxy - 40,"%3.1f%%",sum*100 / maxs);
}
setcolor(col);
}

void freealloc(double far *ptr)
{
    farfree(ptr);
}

void PreChart()
{
    char filename[20];

    textattr(0x07);
    clrscr();
    strcpy(filename,name_file);
    strcat(filename,".lst");
    printf("\nCurent FILE's name : %s\n",filename);
    printf("\nTotal job (on plan) : %d\n\n",Last_node->ID);
    if(PLAN != OFF) printf("SHOW PLAN : ON\n");
    else printf("SHOW PLAN : OFF\n");
    if(ACTUAL != OFF) printf("SHOW ACTUAL : ON\n\n");
    else printf("SHOW ACTUAL : OFF\n\n");
    printf("HEADER or DESCRIPTION for this plan\n");
    printf("-----\n");
    GetDescrib();
}

void Chart_command()
{
    ENTRY *h;
    register int i;

```

```

int j,maxx,maxy,
int x,y,z,v;
double total_Lcost = 0;
double total_Mcost = 0;
int bx,fx,loop,ch;
int bder_day = 0;
int d,now_ID = 1;
int start,finish,begin,end;
int month_day[12]={{31},{28},{31},{30},{31},{30},{31},{31},{30},{31},{30},{31}};
double temp;

```

```

PreChart();
printf("\n      If everything allright press anykey to process\n");
printf("      else press ESC to exit");
ch = InputKey();
if(ch != kESC)
{
    opengraphic();
    maxx=getmaxx();
    maxy=getmaxy();
    h = First_node;
    while(h != NULL)
    {
        total_Mcost = total_Mcost + h->M_cost;
        total_Lcost = total_Lcost + h->L_cost;
        h = h->next;
    }
    h = First_node;
    j = Project->day.da_mon - 1;
    z = Project->day.da_day;
    Plot_sum();
    if( ACTUAL == ON)
        Plot_actual(First_node);
    x = maxx/2-30;
    do {
        gprintf(0,30," ID description d M-cost L-cost");
        y = 57;
        gprintf(0,maxy-35,"          %7.0f",total_Mcost);
        gprintf(195,maxy-35,"%7.0f",total_Lcost);
        gprintf(30,maxy-35," SUM ");
        d = bder_day;
        showday(z,j,x,0,maxx);
        for(i = x;i < maxx;i += 10*DAY)
        {
            gprintf(i,30,"%-3u",d);
            d+=10;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
setfillstyle(SOLID_FILL,DARKGRAY);
bar(0,y,x-DAY-2,maxy-70);
setfillstyle(SOLID_FILL,BLUE);
bar(x-DAY,y,maxx,maxy-70);
setlinestyle(SOLID_LINE,1,1);
rectangle(0,maxy-65,maxx,maxy-15);
outtextxy(maxx-100,maxy-textheight("H")-2,"F10 - print");
setlinestyle(DOTTED_LINE,1,1);
for(i=-DAY;i+x<maxx;i+=DAY)
    line(x+i,y,x+i,maxy-70);
setlinestyle(SOLID_LINE,1,1);
line(DAY,y,maxx,y);
y+=(maxy-127)/11;
for(v=now_ID; v<now_ID+11 && h!=NULL; v++){
    line(DAY,y,maxx,y);
    gprintf(0,y-10,"%3d %10s %d %7.0f%7.0f",h->ID,h->Jobname,
        h->duration,h->M_cost,h->L_cost);
    start = tr(&h->start); /*---- translat date to be day-number ----*/
    finish = tr(&h->finish);
    begin = tr(&h->begin);
    end = tr(&h->end);
    if(PLAN == ON)
    {
        setfillstyle(SOLID_FILL,CYAN);
        bx = (start > bder_day) ? (start - bder_day)*DAY : 0;
        fx = (finish <= (bder_day + (maxx - x)/5)) ? (finish - bder_day)*DAY : maxx-x;
        if(finish <= bder_day)
        {
            bx = 0;
            fx = 0;
        }
        bar(x + bx,y - 20,x + fx,y - 15);
        temp = (double)(v - now_ID + 1)*100 / 11;
        gprintf(maxx - 40,y-10,"%3.2f%%",100 - temp );
        y +=(maxy-127)/11;
    }
    if(ACTUAL == ON)
    {
        setfillstyle(SOLID_FILL,YELLOW);
        y +=(maxy-127)/11;
        bx = (begin > bder_day) ? (begin - bder_day)*DAY : 0;
        fx = (end <= (bder_day + (maxx - x)/5)) ? (end - bder_day)*DAY : maxx-x;
        if(end <= bder_day)
        {
            bx = 0;

```

```

        fx = 0;
    }
    bar( x + bx,y-10,x + fx,y-5);
    if(h->complete != 0)
        gprintf(x + fx+1,y-10,"%3d",h->complete);
    y +=(maxy-127)/11;
}
h = h->next;
i = x- 65;
}
Plot(bder_day,x,maxy-127,maxy-70,1,57);
if(ACTUAL == ON)
    Plot_real(bder_day,x,maxy-127,maxy-70,1,57);
ch = InputKey();
switch(ch)
{
    case kLEFT : bder_day -= 10;z -=10;break;
    case kRIGHT : bder_day += 10;z +=10;break;
    case kUP : now_ID = 2;break;
    case kDOWN : now_ID += 2;break;
    case kESC : loop = FALSE;break;
    case kF10 : cleardevice();
                Chart_print();
                break;
}
if(bder_day <= 0)
{
    bder_day = 0;
    z = Project->day.da_day;
    j = Project->day.da_mon - 1;
}
if(z > month_day[j])
{
    z = z-month_day[j];
    j++;
    if(j > 11 ) j=0;
}
if(z < 0)
{
    j--;
    if(j < 0 ) j = 11;
    z = z + month_day[j];
}
if(now_ID <= 1)
    now_ID = 1;
h = First_node;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        while(h->ID != now_ID && h != NULL)
        {
            h = h->next;
        }
        cleardevice();
    }while(loop != FALSE); /*----- end loop do - while -----*/
    farfree(pt);
    if(ACTUAL ==ON)
        farfree(ac);
    closegraph();
    PreChart();
}

int gprintf(int x,int y,char *fint, ...)
{
    va_list argptr;
    int cnt;
    char buffer[80];

    va_start(argptr, fint);
    cnt = vsprintf(buffer, fint, argptr);
    va_end(argptr);
    outtextxy(x,y,buffer);
    return(cnt);
}

void opengraphlc()
{
    int mode,driver = DETECT;

    initgraph(&driver,&mode,"");
}

void showday(int z,int j,int x,int y,int maxx)
{
    int i,o,p,v,col;
    int month_day[12]={{31},{28},{31},{30},{31},{30},{31},{31},{30},{31},{30},{31}};
    char *month_name[]={"JAN","FEB","MAR","APR","MAY","JUN","JUL","AUG",
        "SEP","OCT","NOV","DEC"};

    o = z;
    p = j;
    col=getcolor();
    setcolor(BROWN);

```

```

for(i = x,v=x,i < maxx;i += 10*DAY)
{
    gprintf(i,y + 15,"%-3u",o);
    o+=10;
    if(o > month_day[p])
    {
        o = o - month_day[p];
        gprintf(v,y,"%3s",*(month_name+p));
        v+=month_day[p] * DAY;
        p++;
        if(p>11) p=0;
    }
}
setcolor(col);
}
void Summary()
{
    unsigned int x,y,maxx,maxy,page = 1;
    double temp = 0,summary = 0;
    ENTRY *job = NULL;
    char *sum[] = { "Cost of work performed to date",
        "Materials stored on site ",
        "Total work performed and materials store",
        "Less 10 % retained",
        "Net work performed and materials store",
        "Less amont of Previous payments",
        "Balance due this payment"
    };
    double material,prepayment,total_cost = 0;
    char filename[20];

    strcpy(filename,name_file);
    strcat(filename, ".des");
    textattr(0x07);
    clrscr();
    material = 0;
    prepayment = 0;
    printf("Materials stored on site (schedule attach) : ");
    material = Getdouble(7);
    printf("\nLess amont of Previous payments : ");
    prepayment = Getdouble(10);
    opengraphic();
    maxx = getmaxx();
    maxy = getmaxy();
    job = First_node;
    summary = 0;

```

```

while(job != NULL)
{
    rectangle(5,5,maxx - 5,maxy - 15);
    x = 10;
    y = 10;
    gprintf(x,y,"TITLE : %s",Project->title);
    y +=15;
    gprintf(x,y,"PROJECT : %s",Project->project);
    gprintf(maxx/2,y,"LOCATION : %s",Project->location);
    y +=15;
    gprintf(x,y,"ARCHITECT : %s",Project->architect);
    gprintf(maxx/2,y,"ENGINEER : %s",Project->engineer);
    y +=15;
    gprintf(x,y,"OWNER : %s",Project->owner);
    gprintf(maxx/2,y,"FOR PERIOD : %s",Project->period);
    y +=15;
    gprintf(x,y,"PROJECT START : %2d/%2d/%2d ",
        Project->day.da_day,Project->day.da_mon,Project->day.da_year);
    Project->path = Last_node->ID;
    gprintf(maxx/3 + 10,y,"TOTAL JOB : %2d",Project->path);
    gprintf(2*maxx/3,y,"Time Scale : DAY");
    y +=15;
    gprintf(x,y,"UNIT/COST : %s",Project->unit_cost);
    gprintf(2*maxx/3,y,"FILE : %s",filename);
    gprintf(maxx-80,y,"Page %d",page);
    SaveDescrib(Project);
    y +=15;
    line(5,y,maxx-5,y);
    gprintf(x,y+2,"Id Id Description Total Completed Cost to Percent ");
    gprintf(x,y+10," cost to Date complete Complete");
    y +=20;
    line(5,y,maxx-5,y);
    y +=5;
    while(y < maxy - 15 && job != NULL)
    {
        temp = (job->M_cost + job->L_cost);
        gprintf(x+3,y,"%2d %s" job->ID,job->Jobname);
        gprintf(maxx/3 + 50,y," %10.0f%10.0f%10.0f %3d",
            job->M_cost + job->L_cost,
            (double) job->complete/100* temp,
            (double) (100 - job->complete)/100*temp,job->complete);
        y += 15;
        line(x,y-5,maxx-5,y-5);
        total_cost = total_cost + temp;
        summary = summary + (job->complete/100 *temp);
        job = job->next;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }          /*-----end while loop -----*/
page++;
if(y < maxy - 125)
{
    gprintf(maxx/3 + 50,y," %10.0f%10.0f%10.0f %3.1f",total_cost,
            summary,total_cost - summary,summary * 100/ total_cost);
    y += 15;
    line(x,y-5,maxx-5,y-5);
    gprintf(x+3,y,"A. %s",*sum);
    gprintf(maxx/3 + 155,y,"%10.0f",summary);
    y += 15;
    gprintf(x+3,y,"B. %s",*(sum+1));
    gprintf(maxx/3 + 155,y,"%10.0f",material);
    summary = summary + material;
    y += 15;
    gprintf(x+3,y,"C. %s",*(sum+2));
    gprintf(maxx/3 + 155,y,"%10.0f",summary);
    y += 15;
    gprintf(x+3,y,"D. %s",*(sum+3));
    gprintf(maxx/3 + 155,y,"%10.0f",summary/10);
    summary = 0.9 * summary;
    y += 15;
    gprintf(x+3,y,"E. %s",*(sum+4));
    gprintf(maxx/3 + 155,y,"%10.0f",summary);
    y += 15;
    gprintf(x+3,y,"F. %s",*(sum+5));
    gprintf(maxx/3 + 155,y,"%10.0f",prepayment);
    summary = summary - prepayment;
    y += 15;
    gprintf(x+3,y,"G. %s",*(sum+6));
    gprintf(maxx/3 + 155,y,"%10.0f",summary);
}

outtextxy(maxx-300,maxy-textheight("H")-2,"Press Any key to Exit or F10 to print");
switch(InputKey())
{
case kF10 : outtextxy(maxx-300,maxy-textheight("H")-2,"");
            print_graph1();
            break;
default   : break;
}

}          /*-----end while loop main when job = NULL -----*/
closegraph();
}          /* -----end Summary function -----*/

```

```

void Chart_print1()
{
ENTRY *h;
register int i;
int j,maxx,maxy;
int x,y,z,v;
double total_Lcost = 0;
double total_Mcost = 0;
double far *pt1 = NULL;
int bx,fx,loop,ch;
int bder_day = 0;
int d,now_ID = 1,last_ID;
int start,finish,begin,end;
double temp;
int month_day[12]={ {31},{28},{31},{30},{31},{30},{31},{31},{30},{31},{30},{31}};

maxx=getmaxx();
maxy=getmaxy();
h = First_node;
while(h != NULL)
{
    total_Mcost = total_Mcost + h->M_cost;
    total_Lcost = total_Lcost + h->L_cost;
    h = h->next;
}
last_ID = Last_node->ID;
h = First_node;
j = Project->day.da_mon - 1;
z = Project->day.da_day;
x = maxx/2-30;
loop=TRUE;
do {
    if(x==maxx/2-30)
        gprintf(0,30," ID description M-cost L-cost");
    y = 57;
    d = bder_day;
    showday(z,j,x,0,maxx);
    for(i = x;i < maxx;i += 10*DAY)
    {
        gprintf(i,30,"%-3u",d);
        d+=10;
    }
    setlinestyle(DOTTED_LINE,1,1);
    for(i=-DAY;i+x<maxx;i+=DAY)
        line(x+i,y,x+i,maxy-70);
    setlinestyle(SOLID_LINE,1,1);
}

```

```

line(DAY,y,maxx,y);
y += (maxy-127)/11;
for(v=now_ID; v<now_ID+11 && h!=NULL; v++)
{
    line(DAY,y,maxx,y);
    if(x==maxx/2-30)
        gprintf(0,y-10,"%3d %10s %7.0f%7.0f",h->ID,
            h->Jobname,h->M_cost,h->L_cost);
    start = tr(&h->start); /*---- translat date to be day-number ----*/
    finish = tr(&h->finish);
    begin = tr(&h->begin);
    end = tr(&h->end);
    if(PLAN == ON)
    {
        setfillstyle(SOLID_FILL,BLACK);
        bx = (start > bder_day) ? (start - bder_day)*DAY : 0;
        fx = (finish <= (bder_day + (maxx - x)/5)) ? (finish - bder_day)*DAY : maxx-x;
        if(finish <= bder_day)
        {
            bx = 0;
            fx = 0;
        }
        bar(x + bx,y - 20,x + fx,y - 15);
        rectangle(x + bx,y - 20,x + fx,y - 15);
        temp = (double) h->ID*100 / Last_node->ID;
        gprintf(maxx - 40,y-10,"%3.2f%%",100 - temp);
        y +=(maxy-127)/11;
    }
    if(ACTUAL == ON)
    {
        setfillstyle(SOLID_FILL,YELLOW);
        y -=(maxy-127)/11;
        bx = (begin > bder_day) ? (begin - bder_day)*DAY : 0;
        fx = (end <= (bder_day + (maxx - x)/5)) ? (end - bder_day)*DAY : maxx-x;
        if(end <= bder_day)
        {
            bx = 0;
            fx = 0;
        }
        bar(x + bx,y-10,x + fx,y - 5);
        if(h->complete != 0)
            gprintf(x+fx+1,y-10,"%3d%%",h->complete);
        y +=(maxy-127)/11;
    }
}
h = h->next;
i = x- 65;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
Plot(bder_day,x,(maxy-127)/11*last_ID,
(last_ID-now_ID+1)*(maxy-127)/11+57,last_ID-now_ID,57);
if( bder_day <=0 && last_ID-now_ID<10){
    gprintf(0,maxy-35,"          %7.0f",total_Mcost);
    gprintf(195,maxy-35,"%7.0f",total_Lcost);
    gprintf(30,maxy-35," SUM ");
}
if(h != NULL)
{
    h = h->preceed;
    h = h->preceed;
    now_ID -= 2;
}
x=DAY;
print_graph();          /* ----- print Screen-----*/
setfillstyle(SOLID_FILL,RED);
bar(maxx/2-115,maxy/2-10,maxx/2+115,maxy/2+10);
outtextxy(maxx/2-110,maxy/2-5,"Press any key to continue...");
getch();
if(bder_day + (maxx -x)/ DAY > max_day)
{
    bder_day = 0;
    if(last_ID-now_ID >= 10)
        ch = kDOWN;
    else
        ch = kESC;
}
else
    ch = kRIGHT;
switch(ch)
{
    case kLEFT : bder_day -= 30;z -=30;break;
    case kRIGHT : bder_day += 30;z +=30;break;
    case kUP : now_ID -= 11;break;
    case kDOWN : now_ID += 11;break;
    case kESC : loop = FALSE;break;
}
if(bder_day <= 0)
{
    bder_day = 0;
    z = Project->day.da_day;
    j = Project->day.da_mon - 1;
    x = maxx/2-30;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น. ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if(z > month_day[j])
    {
        z = z-month_day[j];
        j++;
        if( j > 11 )j= 0;
    }
    if(z < 0)
    {
        j--;
        if( j < 0 )j = 11;
        z = z + month_day[j];
    }
    if(now_ID <= 1)
        now_ID = 1;
    h = First_node;
    while(h->ID != now_ID && h != NULL)
    {
        h = h->next;
    }
    cleardevice();
    }while(loop != FALSE);
}

void Chart_print()
{
    ENTRY *h;
    register int i;
    int j,maxx,maxy;
    int x,y,z,v;
    double total_Lcost = 0;
    double total_Mcost = 0;
    int bx,fx,loop,ch;
    int bder_day = 0;
    int d,now_ID = 1,last_ID;
    int start,finish,begin,end,page = 1;
    double temp;
    int month_day[12]={{31},{28},{31},{30},{31},{30},{31},{31},{30},{31},{30},{31}};
    char filename[20];

    strcpy(filename,name_file);
    strcat(filename,".lst");
    maxx=getmaxx();
    maxy=getmaxy();
    h = First_node;

```

```

while(h != NULL)
{
    total_Mcost = total_Mcost + h->M_cost;
    total_Lcost = total_Lcost + h->L_cost;
    h = h->next;
}
last_ID = Last_node->ID;
h = First_node;
j = Project->day.da_mon - 1;
z = Project->day.da_day;
x = 10;
y = 10;
gprintf(x,y,"TITLE : %s",Project->title);
y +=15;
gprintf(x,y,"PROJECT : %s",Project->project);
gprintf(maxx/2,y,"LOCATION : %s",Project->location);
y +=15;
gprintf(x,y,"ARCHITECT : %s",Project->architect);
gprintf(maxx/2,y,"ENGINEER : %s",Project->engineer);
y +=15;
gprintf(x,y,"OWNER : %s",Project->owner);
gprintf(maxx/2,y,"FOR PERIOD : %s",Project->period);
y +=15;
gprintf(x,y,"PROJECT START : %2d/%2d/%2d ",
    Project->day.da_day,Project->day.da_mon,Project->day.da_year);
Project->path = Last_node->ID;
gprintf(maxx/3 + 10,y,"TOTAL JOB : %2d",Project->path);
gprintf(2*maxx/3,y,"Time Scale : DAY");
y +=15;
gprintf(x,y,"UNIT/COST : %s",Project->unit_cost);
gprintf(2*maxx/3,y,"FILE : %s",filename);
gprintf(maxx-80,y,"Page %d",page);
SaveDescrib(Project);
y +=15;
rectangle(5,5,maxx-5,y); /* y = 100 */
y +=15;
x = maxx/2-30;
loop=TRUE;
do {
    if(x==maxx/2-30)
        gprintf(0,y+30," ID description M-cost L-cost");
    d = bder_day;
    if(page == 1) y = 115;
    else y = 0;
    showday(z,j,x,y,maxx);
    y+=30;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(i = x; i < maxx; i += 10*DAY)
{
    gprintf(i,y,"%-3u",d);
    d+=10;
}
y +=15;
setlinestyle(DOTTED_LINE,1,1);
for(i=-DAY;i+x<maxx;i+=DAY)
    line(x+i,y,x+i,maxy-31);
setlinestyle(SOLID_LINE,1,1);
line(DAY,y,maxx,y);
if(page ==1) d = 9;
else d = 11;
y += (maxy-127)/11;
for(v=now_ID; v<now_ID+d && h!=NULL; v++)
{
    line(DAY,y,maxx,y);
    if(x==maxx/2-30)
        gprintf(0,y-10,"%3d %10s %7.0f%7.0f",h->ID,
                h->Jobname,h->M_cost,h->L_cost);
    start = tr(&h->start);
    finish = tr(&h->finish);
    begin = tr(&h->begin);
    end = tr(&h->end);
    if(PLAN == ON)
    {
        setfillstyle(SOLID_FILL,BLACK);
        bx = (start > bder_day) ? (start - bder_day)*DAY : 0;
        fx = (finish <= (bder_day + (maxx - x)/5)) ? (finish - bder_day)*DAY : maxx-x;
        if(finish <= bder_day)
        {
            bx = 0;
            fx = 0;
        }
        bar(x + bx,y - 20,x + fx,y - 15);
        rectangle(x + bx,y - 20,x + fx,y - 15);
        temp = (double) h->ID*100 / Last_node->ID;
        gprintf(maxx - 40,y-10,"%3.2f%%",100 - temp);
        y +=(maxy-127)/11;
    }
    if(ACTUAL == ON)
    {
        setfillstyle(SOLID_FILL,YELLOW);
        y =(maxy-127)/11;
        bx = (begin > bder_day) ? (begin - bder_day)*DAY : 0;
        fx = (end <= (bder_day + (maxx - x)/5)) ? (end - bder_day)*DAY : maxx-x;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if(end <= bder_day)
        {
            bx = 0;
            fx = 0;
        }
        bar( x + bx,y-10,x + fx,y - 5);
        if(h->complete != 0)
            fprintf(x+fx+1,y-10,"%3d%%",h->complete);
        y +=(maxy-127)/11;
    }
    h = h->next;
    i = x- 65;
}
if(page == 1)
{
    Plot_print(bder_day,x,(maxy-127)/11*last_ID,
    (last_ID-now_ID+1)*(maxy-127)/11 + 160,11,160,total_Lcost + total_Mcost);
    if (ACTUAL == ON)
        Plot_rprint(bder_day,x,(maxy-127)/11*last_ID,
        (last_ID-now_ID+1)*(maxy-127)/11 + 160,11,160,total_Lcost + total_Mcost);
}
else
{
    Plot_print(bder_day,x,(maxy-127)/11*last_ID,
    (last_ID-now_ID+1)*(maxy-127)/11 + 45,last_ID-now_ID,45,total_Lcost + total_Mcost);
    if (ACTUAL == ON)
        Plot_rprint(bder_day,x,(maxy-127)/11*last_ID,
        (last_ID-now_ID+1)*(maxy-127)/11 + 45,last_ID-now_ID,45,
        total_Lcost + total_Mcost);
}
if( bder_day <= 0 && last_ID-now_ID < 10){
    rectangle(0,maxy-115,maxx,maxy-15);
    fprintf(0,maxy-80,"          %7.0f",total_Mcost);
    fprintf(195,maxy-80,"%7.0f",total_Lcost);
    fprintf(30,maxy-80," SUM ");
    line(5,maxy - 65,maxx-5,maxy - 65);
    fprintf(30,maxy-50," PERCENT (plan)");
    fprintf(30,maxy-30," PERCENT ACC. (plan)");
    fprintf(30,maxy-40," PERCENT ACC. (actual)");
    fprintf(30,maxy-62," PERCENT (actual)");
}
if(h != NULL)
{
    h = h->preceed;
    h = h->preceed;
    now_ID = 2;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
x = DAY;
/* print Screen*/
print_graph();
setfillstyle(SOLID_FILL,RED);
bar(maxx/2-115,maxy/2-10,maxx/2+115,maxy/2+10);
outtextxy(maxx/2-110,maxy/2-5,"Press any key to continue...");
getch();
if(bder_day + (maxx - x)/ DAY > max_day)
{
    bder_day = 0;
    if(last_ID - now_ID >= 10)
    {
        ch = kDOWN;
        page++;
    }
    else
        ch = kESC;
}
else
{
    ch = kRIGHT;
}
switch(ch)
{
case kLEFT : bder_day -= 30;z -=30;break;
case kRIGHT : bder_day += 30;z +=30;break;
case kUP : if(page == 2)
            now_ID -= 9;
            else
            now_ID -= 11;
            break;
case kDOWN : if(page == 2)
              now_ID += 9;
              else
              now_ID += 11;
              break;
case kESC : loop = FALSE;break;
}
if(bder_day <= 0)
{
    bder_day = 0;
    z = Project->day.da_day;
    j = Project->day.da_mon - 1;
    x = maxx/2-30;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(z > month_day[j])
{
    z = z-month_day[j];
    j++;
    if(j > 11 )j = 0;
}
if(z < 0)
{
    j--;
    if(j < 0 )j = 11;
    z = z + month_day[j];
}
if(now_ID <= 1)
    now_ID = 1;
h = First_node;
while(h->ID != now_ID && h != NULL)
{
    h = h->next;
}
cleardevice();
}while(loop != FALSE);
}
```

```

#include <dos.h>
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
#include "lam.h"

void Beep()
{
    sound(1000);
    delay(10);
    nosound();
}

/*-----
int InputKey()
: Input key and return key-message.
-----*/

int InputKey()
{
char key1,key2;

key1=getch(); /* Read first byte */
if( key1 == 0 ) /* Control Key ? */
{
/* control key */
key2 = getch(); /* get second byte */
switch(key2)
{
case UP : return(kUP);
case DN : return(kDOWN);
case LEFT : return(kLEFT);
case RI GHT : return(kRIGHT);
case P_UP : return(kPGUP);
case P_DN : return(kPGDN);
case HOME : return(kHOME);
case F1 : return(kF1);
case F2 : return(kF2);
case F3 : return(kF3);
case F4 : return(kF4);
case F10 : return(kF10);
case AltX : return(kALTQ);
case AltQ : return(kALTQ);
default : return(0);
}
}
else
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

switch(key1)
{
case ESC : return(kESC);
case CR  : return(kCR);
case INS : return(kINS);
case DEL : return(kDEL);
case BSP : return(kBSP);
case SPC : return(kSPC);
case TAB : return(kTAB);
default  : return(key1);
}
} /*-----end else -----*/
}

int GetInt(int degit)
{
int ch;
int a;
int count,loop;
int startx,starty;
char string[7];
struct text_info text;

gettextinfo(&text);
strcpy(string,""); /* Init string */
startx = text.curx;
starty = text.cury;
count=0;
loop=TRUE;
do{
ch = InputKey();
switch(ch)
{
case '0': /* if == '0'...'9' */
case '1':
case '2':
case '3':
case '4':
case '5':
case '6':
case '7':
case '8':
case '9':
case '-':

if(count<6)
/* Check limit */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        string[count] = ch;
        count++;
        gotoxy(startx,starty);
        cprintf("%%-6s",string);
        gotoxy(startx+count,starty);
    }
    break;
case kBSP :      /*----- Back Space -----*/
    count--;
    if(count <0)
    {
        count=0;
    }
    for(a=count;a<6;a++)
        string[a]=string[a+1];
    gotoxy(startx,starty);
    cprintf("%%-6s",string);
    gotoxy(startx+count,starty);
    break;
case kCR : string[7] = '\0';
            loop = FALSE;
            break;
}
}while(loop == TRUE);
gotoxy(startx+degit,starty);
return(atoi(string));
}

double Getdouble(int degit)
{
int ch;
int a;
int count,loop;
int startx,starty;
char string[7];
struct text_info text;

    gettextinfo(&text);
    strcpy(string," ");
    startx = text.curx;
    starty = text.cury;
    count=0;
    loop=TRUE;
    do{
        ch = InputKey();
        /* Init string */

```

```

switch(ch)
{
    case '0':          /* -----if = '0'...'9' -----*/
    case '1':
    case '2':
    case '3':
    case '4':
    case '5':
    case '6':
    case '7':
    case '8':
    case '9':
    case '-':
        if(count<6)
        {
            /* -----Check limit -----*/
            string[count] = ch;
            count++;
            gotoxy(startx,starty);
            cprintf("%-6s",string);
            gotoxy(startx+count,starty);
        }
        break;
    case kBSP :        /* -----Back Space -----*/
        count--;
        if(count < 0)
        {
            count=0;
        }
        for(a=count;a<6;a++)
            string[a]=string[a+1];
        gotoxy(startx,starty);
        cprintf("%-6s",string);
        gotoxy(startx+count,starty);
        break;
    case kCR : string[7] = '\0';
                loop = FALSE;
                break;
}
}while(loop == TRUE);
gotoxy(startx+degit,starty);
return(atof(string));
}

```

```

int getstring(int length,char *str)
{
    struct text_info text;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int message,ret,loop;
int count;
int x,y,attr,i;

gettextinfo(&text);
x=text.curx;
y=text.cury;
attr = text.attribute;
count=0;loop=TRUE;
textattr(F_YELLOW|B_BLUE);
for(count = 0;count < length;count++)
    cprintf(" ");
count = 0;
do{
    gotoxy(x+count,y);
    message = InputKey();
    if(message==kTAB) {loop=FALSE;ret=kTAB;}
    if(message==kESC) {loop=FALSE;ret=kESC;}
    if(message==kCR) {loop=FALSE;ret=kCR;}
    if(message==kBSP)
    {
        count--;
        if (count<0) count=0;
        gotoxy(x+count,y);cprintf(" ");
        gotoxy(x+count,y);
        for(i=count;i<length;i++)
            str[i] = ' ';
        str[i+1]='\0';
    }
    if( ((message>='a') && (message<='z'))
        || ((message>='A') && (message<='Z'))
        || ((message>='0') && (message<='9'))
        || (message == '/')
        || (message == 318) /*-----space bar -----*/
        || (message == '-')
    )
    {
        if(count>=length-1) count=length-1;
        else
        {
            if(message != 318)
                str[count] = message;
            else
                str[count] = ' ';
            cprintf("%c",str[count]);
            count++;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        } /*-----end else -----*/
    } /*-----end if-----*/
}while(loop==TRUE);
str[count]='\0';
gotoxy(x+length,y);
textattr(attr);
return(ret);
}

```

```
void get_date(struct date *d,char *t,int date_form)
```

```

{
int k = 0;
int j=0;
int count = 0;
char temp[10];

while(*t != '\0')
{
    if(*t == '/')
    {
        j = 0;
        count++;
        k = -1;
        temp[0] = '\0';
    }
    if(*t != '/')
    {
        temp[k] = *t;
        temp[k+1] = '\0';
        j = atoi(temp);
    }
    if(date_form == 1)
    {
        if(count == 0)
            d->da_day = j;
        if(count == 1)
            d->da_mon = j;
        if(count == 2)
            d->da_year = j;
    }
    t++;
    k++;
} /*----- end while loop-----*/
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void AddEntry(ENTRY**h,ENTRY **t)
{
ENTRY *new;

new = (ENTRY *)malloc(sizeof(ENTRY));
strcpy(new->Jobname,"");
new->ID = 0;
new->duration = 0;
new->start.da_year = 0;
new->start.da_day = 0;
new->start.da_mon = 0;
new->finish.da_year = 0;
new->finish.da_day = 0;
new->finish.da_mon = 0;
new->pre_entry = 0;
new->M_cost = 0;
new->L_cost = 0;
new->now = 0;
new->complete = 0;
new->begin.da_year = 0;
new->begin.da_day = 0;
new->begin.da_mon = 0;
new->end.da_year = 0;
new->end.da_day = 0;
new->end.da_mon = 0;
if(*t != NULL)
{
(*t)->next = new;
new->preceed = *t;
}
if(*h==NULL)
{
*h = new;
new->preceed = NULL;
}
*t = new;
(*t)->next = NULL;
}

```

```

void DelEntry(ENTRY *del)
{
if(del == NULL) return;
if(del == First_node)
if(del == del->next)
{
First_node = NULL;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        free(del);
        return;
    }else
        First_node = del->next;
    del->next->preceed = del->preceed;
    del->preceed->next = del->next;
    if(del == Last_node)
        Last_node = del->preceed;
    free(del);
}

void onellst(ENTRY *n)
{
    printf(" %2d %10s %2d %2d/%2d/%2d %2d/%2d/%2d",n->ID,n->Jobname,
        n->duration,n->start.da_day,n->start.da_mon,n->start.da_year,
        n->finish.da_day,n->finish.da_mon,n->finish.da_year);
    printf(" %2d %10.0f %10.0f",n->pre_entry,n->M_cost,n->L_cost);
    now_x = wherex();
    now_y = wherexy();
    if(now_y > 20)
    {
        movetext(0,4,80,21,0,3);
        now_y = 20;
        gotoxy(0,20);
        delline();
    }
    gotoxy(1,23);
    clreol();
}
}

```

```

void FreeList(ENTRY **h)
{ ENTRY *n;

    n = *h;
    while(n != NULL)
    {
        free(n);
        n = n->next;
    }
    First_node = NULL;
}

```

```

void Showlist(ENTRY *h)
{
    ENTRY *n;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

textatfr(0x07);
clrscr();
gotoxy(1,1);
puts("\nList of Activity : ");
n = h;
printf("\n ID Job name dur start finish PreReq Material cost Labour cost");
while(n != NULL)
{
printf("\n %2d %10s %2d %2d/%2d/%2d %2d/%2d/%2d %2d %10.0f %10.0f",
n->ID,n->Jobname,n->duration,
n->start.da_day,n->start.da_mon,n->start.da_year,
n->finish.da_day,n->finish.da_mon,n->finish.da_year,
n->pre_entry,n->M_cost,n->L_cost);
now_x = wherex();
now_y = wherey();
if(now_y > 20)
{
movetext(0,4,80,21,0,3);
now_y = 20;
gotoxy(0,20);
delline();
}
n = n->next;
} /*-----end while loop -----*/
}

```

```

void Savellst(ENTRY *h)
{
FILE *file;
char *temp;
ENTRY_REC record;
int x,y;

x = wherex();
y = wherey();
gotoxy(1,23);
clreol();
gotoxy(1,22);
clreol();
cprintf("Enter file name to Oparate : %s.lst",name_file);
gotoxy(29,22);
strcpy(temp,name_file);
strcat(temp, ".lst");
if (file = fopen(temp,"w+b")) != NULL)
while(h != NULL)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

strcpy(record.Jobname,h->Jobname);
record.ID          = h->ID;
record.duration    = h->duration;
record.start.da_year    = h->start.da_year;
record.start.da_day = h->start.da_day;
record.start.da_mon     = h->start.da_mon;
record.finish.da_year   = h->finish.da_year;
record.finish.da_day    = h->finish.da_day;
record.finish.da_mon    = h->finish.da_mon;
record.pre_entry       = h->pre_entry;
record.M_cost          = h->M_cost;
record.L_cost          = h->L_cost;
record.now             = h->now;
record.complete       = h->complete;
record.begin.da_year   = h->begin.da_year;
record.begin.da_day    = h->begin.da_day;
record.begin.da_mon    = h->begin.da_mon;
record.end.da_year     = h->end.da_year;
record.end.da_day      = h->end.da_day;
record.end.da_mon      = h->end.da_mon;
fwrite(&record,sizeof(ENTRY_REC),1,file);
h = h->next;
} /*-----end while loop -----*/
gotoxy(x,y);
fclose(file);
}

void Add_command()
{
int loop;
int id = 1;
char str[12];

loop = TRUE;
Showlist(First_node);
AddEntry(&First_node,&Last_node);
gotoxy(1,22);
printf(" ID Job name dur start finish PreReq Material cost Labour cost\n");
do{
textattr(0x07);
if(Last_node->preceed != NULL)
id = Last_node->preceed->ID + 1;
Last_node->ID          = id;
printf("%3d ",id);
if(getstring(15,Last_node->Jobname) == kESC)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        loop = FALSE;
        DelEntry>Last_node);
    }
    else
    {
        Last_node->duration = GetInt(3);
        printf(" ");
        getstring(11,str);
        get_date(&Last_node->start,str,1);
        printf(" ");
        getstring(11,str);
        get_date(&Last_node->finish,str,1);
        printf(" ");
        Last_node->pre_entry = GetInt(3);
        printf(" ");
        Last_node->M_cost = Getdouble(10);
        printf(" ");
        Last_node->L_cost = Getdouble(10);
        max_day= max(tr(&Last_node->finish),max_day);
        max_actual= max(tr(&Last_node->end),max_actual);
        AddEntry(&First_node,&Last_node);
    }
}
if(loop == TRUE)
    onelist>Last_node->preceed);
}while(loop == TRUE);
gotoxy(1,23);
clreol();
}/*-----end Add command -----*/

void Del_command()
{
int ID,find = FALSE;
ENTRY *temp;

gotoxy(1,22);clreol();
cprintf("DELETE ID-number : ");
ID = GetInt(3);
temp = First_node;
do{
    if(ID == temp->ID)
        find = TRUE;
    else
        temp = temp->next;
}while(find == FALSE);
gotoxy(1,22);
printf(" ID Job name dur start finish PreReq Material cost Labour cost");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

printf("\n %2d %10s %2d %2d/%2d/%2d %2d/%2d/%2d %2d %10.0f %10.0fn",
temp->ID,temp->Jobname,temp->duration,
temp->start.da_day,temp->start.da_mon,temp->start.da_year,
temp->finish.da_day,temp->finish.da_mon,temp->finish.da_year,
temp->pre_entry,temp->M_cost,temp->L_cost);
clrcol();
printf(" Do you want to delete the list ? \n");
clrcol();
printf(" Y or N :");
switch(InputKey()){
    case 'y' :
    case 'Y' :    if(temp != First_node)
                  DelEntry(temp);
                  else{
                    First_node = temp->next;
                    temp->next->preceed = NULL;
                    free(temp);
                  }
                  break;
    case 'n' :
    case 'N' :    break;
}
Showlist(First_node);
}

void Insert_command()
{
int ID_before,find = FALSE;
ENTRY *ins,*temp;
char str[12];

gotoxy(1,22);clrcol();
cprintf("before ID-number :");
ID_before = GetInt(3);
temp = First_node;
do{
    if(ID_before == temp->ID)
        find = TRUE;
    else
        temp = temp->next;
}while(find == FALSE);
AddEntry(&First_node,&ins);
gotoxy(1,22);
printf(" ID Job name dur start finish PreReq Material cost Labour cost\n");
ins->ID = GetInt(3);

```

```

if(ins->ID == 0)
{
    DelEntry(ins);
}
else
{
    getstring(15,ins->Jobname);
    ins->duration = GetInt(3);
    getstring(11,str);
    get_date(&ins->start,str,1);
    getstring(11,str);
    get_date(&ins->finish,str,1);
    ins->pre_entry = GetInt(3);
    ins->M_cost = Getdouble(10);
    ins->L_cost = Getdouble(10);
}
if(temp == First_node)
{
    ins->preceed = NULL;
    ins->next = First_node;
    First_node->preceed = ins;
    First_node = ins;
}
else
{
    temp->preceed->next = ins;
    ins->preceed = temp->preceed;
    ins->next = temp;
    temp->preceed = ins;
}
Showlist(First_node);
}

```

```

void Modify_command()
{
    int ID,find = FALSE;
    ENTRY *temp;
    char str[12];

    Showlist(First_node);
    gotoxy(1,22);clrscr();
    cprintf("Modify ID-number :");
    ID = GetInt(3);
    if(ID != kESC)
    {
        temp = First_node;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

do{
    if(ID == temp->ID)
        find = TRUE;
    else
        temp = temp->next;
}while(find == FALSE);
gotoxy(1,22);
printf(" ID Job name dur start finish PreReq Material cost Labour cost");
printf("\n %2d %10s %2d %2d/%2d/%2d %2d/%2d/%2d %2d %10.0f %10.0f",
temp->ID,temp->Jobname,temp->duration,
temp->start.da_day,temp->start.da_mon,temp->start.da_year,
temp->finish.da_day,temp->finish.da_mon,temp->finish.da_year,
temp->pre_entry,temp->M_cost,temp->L_cost);
gotoxy(1,23);
temp->ID = GetInt(3);
if(temp->ID != 0)
{
    getstring(15,temp->Jobname);
    temp->duration = GetInt(3);
    getstring(11,str);
    get_date(&temp->start,str,1);
    getstring(11,str);
    get_date(&temp->finish,str,1);
    temp->pre_entry = GetInt(3);
    temp->M_cost = Getdouble(10);
    temp->L_cost = Getdouble(10);
}
Showlist(First_node);
} /*----- if key = ESC -----*/
}
void ShowACTUAL(ENTRY *h)
{
ENTRY *n;

clrscr();
gotoxy(1,1);
puts("\nList of Activity : ");
n = h;
printf("\n ID Job name dur now complete begin end");
while(n != NULL) {
    printf("\n %2d %10s %2d %2d %2d %2d/%2d/%2d %2d/%2d/%2d",
n->ID,n->Jobname,n->duration,n->now,n->complete,
n->begin.da_day,n->begin.da_mon,n->begin.da_year,
n->end.da_day,n->end.da_mon,n->end.da_year);
    now_x = wherex();
    now_y = wherey();
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        n = n->next;
    }          /*-----end while loop -----*/
}
void Add_ACTUAL()
{ int loop;
char str[12];

loop = TRUE;
AddEntry(&First_node,&Last_node);
gotoxy(1,22);
printf(" ID Job name dur now complete begin  end\n");
do{
    Last_node->ID          = GetInt(3);
    if(Last_node->ID == 0)
    {
        loop = FALSE;
        DelEntry(Last_node);
    }
    else
    {
        getstring(15,Last_node->Jobname);
        Last_node->duration = GetInt(3);
        Last_node->now      = GetInt(3);
        Last_node->complete = GetInt(3);
        getstring(11,str);
        get_date(&Last_node->begin,str,1);
        getstring(11,str);
        get_date(&Last_node->end,str,1);
        AddEntry(&First_node,&Last_node);
    }
    if(loop == TRUE)
        onelist(Last_node->preceed);
}while(loop == TRUE);
gotoxy(1,23);
clreol();
}/*-----end Add ACTUAL -----*/
void Del_ACTUAL()
{
int ID,find = FALSE;
ENTRY *temp;

gotoxy(1,22);clreol();
cprintf("DELETE ID-number :");
ID = GetInt(3);
temp = First_node;

```

```

do{
    if(ID == temp->ID)
        find = TRUE;
    else
        temp = temp->next;
}while(find == FALSE);
gotoxy(1,22);
printf(" ID Job name dur now complete begin end");
printf("\n %2d %10s %2d %2d %2d %2d %2d",temp->ID,temp->Jobname,
        temp->duration, temp->now,temp->complete,temp->begin,temp->end);
clrhol();
printf(" Do you want to delete the list ? \n");
clrhol();
printf(" Y or N :");
switch(InputKey())
{
    case 'y':
    case 'Y': temp->now = 0;
              temp->complete = 0;
              temp->begin.da_year = 0;
              temp->begin.da_day = 0;
              temp->begin.da_mon = 0;
              temp->end.da_year = 0;
              temp->end.da_day = 0;
              temp->end.da_mon = 0;
              break;
}
ShowACTUAL(First_node);
}
void Insert_ACTUAL()
{
int ID_before,find = FALSE;
ENTRY *ins,*temp;
char str[12];

gotoxy(1,22);clrhol();
cprintf("before ID-number :");
ID_before = GetInt(3);
temp = First_node;
do{
    if(ID_before == temp->ID)
        find = TRUE;
    else
        temp = temp->next;
}while(find == FALSE);
AddEntry(&First_node,&ins);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gotoxy(1,22);
printf(" ID Job name dur now complete begin end\n");
ins->ID = GetInt(3);
if(ins->ID == 0)
{
    DelEntry(ins);
}
else
{
    getstring(15,ins->Jobname);
    ins->duration = GetInt(3);
    ins->now      = GetInt(3);
    ins->complete = GetInt(3);
    getstring(11,str);
    get_date(&ins->begin,str,1);
    getstring(11,str);
    get_date(&ins->end,str,1);
}
if(temp == First_node)
{
    ins->preceed = NULL;
    ins->next = First_node;
    First_node->preceed = ins;
    First_node = ins;
}
else
{
    temp->preceed->next = ins;
    ins->preceed = temp->preceed;
    ins->next = temp;
    temp->preceed = ins;
}
ShowACTUAL(First_node);
}

```

```

void Modify_ACTUAL()

```

```

{
    int ID,find = FALSE;
    ENTRY *temp;
    char str[12];

    gotoxy(1,22);clrhol();
    cprintf("Modify ID-number :");
    ID = GetInt(3);
    temp = First_node;

```

```

do{
    if(ID == temp->ID)
        find = TRUE;
    else
        temp = temp->next;
}while(find == FALSE);
gotoxy(1,22);
printf(" ID Job name dur now complete begin end\n");
printf(" %2d %10s %2d %2d %2d %2d/%2d/%2d %2d/%2d/%2d",
    temp->ID,temp->Jobname,temp->duration,temp->now,temp->complete,
    temp->begin.da_day,temp->begin.da_mon,temp->begin.da_year,
    temp->end.da_day,temp->end.da_mon,temp->end.da_year);
gotoxy(1,23);
temp->ID = GetInt(3);
if(temp->ID != 0)
{
    gotoxy(25,23);
    temp->now = GetInt(3);
    printf(" ");
    temp->complete = GetInt(3);
    printf(" ");
    getstring(11,str);
    get_date(&temp->begin,str,1);
    getstring(11,str);
    get_date(&temp->end,str,1);
}
ShowACTUAL(First_node);
}

void Edit_Project()
{
FILE *file;
char filename[20];
ENTRY_REC record;
int x,y;
    max_day= 0;
    gotoxy(1,23);
    clrcol();
    gotoxy(1,22);
    clrcol();
    gotoxy(1,1);
    clrscr();
    Project = NULL;
    GetDescrib();
    getch();
    puts("\nList of Activity : ");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

printf("\n ID Job name dur start finish PreReq Material cost Labour cost");
y = wherey();
strcpy(filename,name_file);
strcat(filename, ".lst");
FreeList(&First_node);
if (file = fopen(filename,"r+b")) != NULL)
    fread(&record,sizeof(ENTRY_REC),1,file);
while(!feof(file))
{
    AddEntry(&First_node,&Last_node);
    strcpy(Last_node->Jobname,record.Jobname);
    Last_node->ID = record.ID;
    Last_node->duration = record.duration;
    Last_node->start.da_day = record.start.da_day;
    Last_node->start.da_mon = record.start.da_mon;
    Last_node->start.da_year = record.start.da_year;
    Last_node->finish.da_day = record.finish.da_day;
    Last_node->finish.da_mon = record.finish.da_mon;
    Last_node->finish.da_year = record.finish.da_year;
    Last_node->pre_entry = record.pre_entry;
    Last_node->M_cost = record.M_cost;
    Last_node->L_cost = record.L_cost;
    Last_node->now = record.now;
    Last_node->complete = record.complete;
    Last_node->begin.da_day = record.begin.da_day;
    Last_node->begin.da_mon = record.begin.da_mon;
    Last_node->begin.da_year = record.begin.da_year;
    Last_node->end.da_day = record.end.da_day;
    Last_node->end.da_mon = record.end.da_mon;
    Last_node->end.da_year = record.end.da_year;
    Last_node->next = NULL;
    max_day= max(max_day,tr(&Last_node->finish));
    max_actual= max(tr(&Last_node->end),max_actual);
    gotoxy(1,y);
printf("\n %2d %10s %2d %2d/%2d/%2d %2d/%2d/%2d %2d %10.0f %10.0f",
    Last_node->ID,Last_node->Jobname,Last_node->duration,
    Last_node->start.da_day,Last_node->start.da_mon,Last_node->start.da_year,
    Last_node->finish.da_day,Last_node->finish.da_mon,Last_node->finish.da_year,
    Last_node->pre_entry,Last_node->M_cost,Last_node->L_cost);
    y++;
    now_y = y;
    if(now_y > 20)
    {
        movetext(0,1,80,21,0,0);
        now_y = 20;
        y = 20;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        gotoxy(0,y);
        delline();
    } /*-----end if loop-----*/
    fread(&record,sizeof(ENTRY_REC),1,file);
} /*-----end while loop -----*/
fclose(file);
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include "lam.h"
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

char *name[] = { "Q", "P", "I", "a", "n" };
int month_day[12] = { {31}, {28}, {31}, {30}, {31}, {30}, {31}, {31}, {30}, {31}, {30}, {31} };
char month_name[12][3] = { "JAN", "FEB", "MAR", "APR", "MAY", "JUN", "JUL", "AUG",
                          "SEP", "OCT", "NOV", "DEC" };
unsigned int tr(struct date *n);

void Add_sum(SUM_LIST **h, SUM_LIST **t)
{ SUM_LIST *new;

  new = (SUM_LIST *)malloc(sizeof(SUM_LIST));
  new->sum = 0;
  new->month = 0;
  new->day = 0;
  if(*t != NULL)
    (*t)->nextsum = new;
  if(*h == NULL)
    *h = new;
  *t = new;
  (*t)->nextsum = NULL;
}

void Free_sum(SUM_LIST **h)
{
SUM_LIST *n;

  n = *h;
  while(n != NULL)
  {
    free(n);
    n = n->nextsum;
  }
  First_sum = NULL;
}

void Total_sum()
{
SUM_LIST *this, *temp;
ENTRY *n;
int a, x, i, cost_per_day;
int month = 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

a = max_day / 30;
if(max_day % 30 > 0)
    a = a + 1;
for (i = 0; i < a; i++)
    Add_sum(&First_sum, &this);
n = First_node;
while(n != NULL)
{
    temp = First_sum;
    month = 1;
    cost_per_day = (n->M_cost + n->L_cost) / n->duration;
    x = n->start.da_day;
    while( x < n->finish.da_day)
    {
        if(month * 30 <= n->finish.da_day)
        {
            if(month * 30 - x < 30)
                temp->sum = temp->sum + ((month*30 - x) * cost_per_day);
            else
                temp->sum = temp->sum + (30 * cost_per_day);
            temp->month = month;
            x = 30 * (month);
            month++;
            temp = temp->nextsum;
        }
        if( n->finish.da_day < month * 30)
            x = 30 * (month);
    }
    temp->sum = temp->sum + (n->finish.da_day - (x - 30)) * cost_per_day;
    temp->month = month;
    n = n->next;
}
}

```

```

void Plot_sumthing()
{
    ENTRY *n;
    unsigned int x,i,cost_per_day,fn,st;
    unsigned long *pt,*pt1 = NULL;

    pt = calloc(max_day,sizeof(unsigned long));
    pt1=pt;
    for(i = 1;i <= max_day;i++)
        *pt1++ = 0;
    n = First_node;
}

```

```

while(n !=NULL)
{
    cost_per_day = (n->M_cost + n->L_cost) / n->duration;
    pt1=pt;
    st=tr(&n->start);
    fn=tr(&n->finish);
    for(x = 1; x <= fn; x++)
    {
        if(x >= st )
            *pt1 = *pt1 + cost_per_day;
        pt1++;
    }
    n = n->next;
}
free(pt);
}

unsigned int tr(struct date *n)
{
    unsigned int i,y,ret = 0;

    y=0;
    i = Project->day.da_mon - 1;
    if(n->da_mon-1 <= 0)
        ret = 0;
    else
    {
        if(i < n->da_mon-1 )
            do{
                do{
                    if(i > 11)
                    {
                        y++;
                        i = 0;
                        ret = ret + month_day[0];
                        i++;
                    }
                    else
                    {
                        ret = ret + month_day[i];
                        i++;
                    }
                }while(i < n->da_mon-1 );
            }while(y < n->da_year - Project->day.da_year);
        ret = ret + n->da_day - Project->day.da_day;
    }
    /*----- end else-----*/
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    return(ret);
}

void show()
{ /* request auto detection */
int gdriver = DETECT, gmode, errorcode;
int style, midx, midy,
int x,y,maxx,maxy,size = 1;

/*----- initialize graphics and local variables -----*/
initgraph(&gdriver, &gmode, "");
/* read result of initialization */
errorcode = graphresult();
if (errorcode != grOk) /* -----an error occurred -----*/
{
    printf("Graphics error: %s\n", grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    exit(1); /* -----terminate with an error code -----*/
}
rectangle(5,5,getmaxx() - 5,getmaxy()- 5);
maxx = getmaxx();
maxy = getmaxy();
midx = maxx / 2 ;
midy = maxy / 2 ;
settextjustify(CENTER_TEXT, CENTER_TEXT);
size = 4; /* -----loop through the available text styles -----*/
setfillstyle(SOLID_FILL,WHITE);
fillellipse(midx - 140 ,midy -120 ,100 ,70);
fillellipse(midx - 100 ,midy-20 ,25 ,15);
fillellipse(midx - 60,midy+10 ,10 ,5);
midx = maxx / 2 - 140;
midy = maxy / 2 - 100;
for (style=DEFAULT_FONT; style<=GOTHIC_FONT; style++)
{
    if (style == TRIPLEX_FONT)
        size = 4;
    if (style == SMALL_FONT)
        size = 6;
    settextstyle(style, HORIZ_DIR, size);
    setcolor(LIGHTBLUE);
    outtextxy(midx ,midy , name[style]);
    midx += 30;
    midy += 30;
    delay(70);
    size = 4;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
size = 4;
midx = maxx / 2 -135;
midy = maxy / 2 -95;
for (style=DEFAULT_FONT; style<=GOTHIC_FONT; style++)
{
    if (style == TRIPLEX_FONT)
        size = 4;
    if (style == SMALL_FONT)
        size = 6;
    settextstyle(style,HORIZ_DIR, size);
    setcolor(LIGHTRED);
    outtextxy(midx ,midy , name[style]);
    midx += 30;
    midy += 30;
    delay(70);
    size = 4;
}
settextstyle(DEFAULT_FONT,HORIZ_DIR,2);
setfillstyle(SOLID_FILL,RED);
bar(maxx-300,maxy-445,maxx-100,maxy-420);
setfillstyle(SOLID_FILL,YELLOW);
bar(maxx-300,maxy-415,maxx-100,maxy-390);
bar(maxx-300,maxy-385,maxx-50,maxy-360);
setcolor(BLACK);
outtextxy(maxx-200,maxy - 430,"Q_PLAN" );
setcolor(BLUE);
outtextxy(maxx-200,maxy - 400,"EASY TO PLAN" );
outtextxy(maxx-175,maxy - 370,"EASY TO MONITOR" );
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
setcolor(WHITE);
outtextxy(maxx-200,maxy - 150,"This program is project planing by BAR-CHART." );
outtextxy(maxx-200,maxy - 130,"developed by" );
outtextxy(maxx-200,maxy - 110,"Piriya Krawkangvan");
outtextxy(maxx-200,maxy - 90,"King Mongkot's Institute o Technology");
outtextxy(maxx-200,maxy - 70,"Ladkrabang");
outtextxy(maxx-200,maxy - 50,"Copyright 1993" );
setcolor(YELLOW);
outtextxy(maxx-200,maxy - 30,"Press any key to Plan" );
getch();
closegraph();
}

```