



การจัดเก็บสินค้าด้วยคอมพิวเตอร์  
COMPUTERIZED WAREHOUSE CONTROL II



โดย

นายกิตติ อัครปรีดี  
นายพรกิตติ สวรรณพงษ์  
นายพฤษ ลิ้มเมตตา

ปฏิญานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตร์บัณฑิต  
สาขาเทคโนโลยีคอมพิวเตอร์อุตสาหกรรม  
ภาคเทคโนโลยีการวัดคุมทางอุตสาหกรรม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2535

๕,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032655

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

สาขาวิชา เทคโนโลยีคอมพิวเตอร์อุตสาหกรรม

คณะ วิศวกรรมศาสตร์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การจัดเก็บสินค้าด้วยคอมพิวเตอร์

COMPUTERIZED WAREHOUSE CONTROL II

ผู้จัดทำ

- |               |            |           |
|---------------|------------|-----------|
| 1. นายกิตติ   | อัครปรีดี  | 34.162142 |
| 2. นายพรกิตติ | สุวรรณพงษ์ | 34.162163 |
| 3. นายพฤษ์    | ล้อมเมตตา  | 34.162164 |



.....อาจารย์ที่ปรึกษา

(อาจารย์ภากร หตะสังภาค)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032665

### บทคัดย่อ

ปฏิญานิพนธ์นี้เป็นส่วนหนึ่งของวิชา DATA BASE ซึ่งเป็นเรื่องที่เกี่ยวข้องกับระบบฐานข้อมูล และฐานข้อมูลจะเป็นการเชื่อมโยงระหว่างตัวเก็บสินค้าซึ่งเป็นส่วนของ WARE HOUSE กับคอมพิวเตอร์ ซึ่งใช้ในการบันทึกรายละเอียดเกี่ยวกับสินค้าแต่ละชนิดที่อยู่ในWARE HOUSE ในปฏิญานิพนธ์เล่มนี้จะเป็นลักษณะของโปรแกรมฐานข้อมูลอย่างเดียว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ABSTRACT

THE PROJECT IS PART OF DATABASE. IT USES BY COMPUTER TO STOCK DESCRIPTION IN THE WAREHOUSE. IT USES CONTROL WAREHOUSE FOR WORK, BRING DESCRIPTION IN AND DESCRIPTION OUT FROM WAREHOUSE.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
วัตถุประสงค์ของปฏิญานิพนธ์	
บทนำ	
ขอบเขตของปฏิญานิพนธ์	1
บทที่ 1 พื้นฐานความรู้เกี่ยวกับการสื่อสารข้อมูลและฐานข้อมูล	2
บทที่ 2 โครงสร้างโปรแกรม	15
บทที่ 3 การทำงานของ PROCEDURE ต่างๆในไฟล์หลัก	17
บทที่ 4 การใช้งานโปรแกรม การจัดเก็บสินค้าด้วยคอมพิวเตอร์	25
บทที่ 5 โปรแกรมทั้งหมดที่ใช้เขียน โดย TURBO PASCAL	38
กิตติกรรมประกาศ	93
หนังสืออ้างอิง	94

## วัตถุประสงค์ของปฏิญานินพนธ์

1. เพื่อเสริมสร้างความรู้
2. เพื่อให้ผู้จัดทำได้มีประสบการณ์ในการทำงานร่วมกัน
3. สามารถนำความรู้ในบทเรียนไปประยุกต์ใช้กับงานจริงได้
4. เพื่อเป็นแหล่งค้นคว้าของนักศึกษารุ่นต่อ ๆ ไป
5. เพื่อนำปฏิญานินพนธ์ส่งอาจารย์ให้ครบตามหลักสูตรปริญญาตรี



## บทนำ

ความก้าวหน้าทางเทคโนโลยี นั้น ไม่มีวันหยุดนิ่งอยู่กับที่ การจัดทำ  
ปฏิญานิพนธ์เรื่อง WARE HOUSE ก็เป็นอีกก้าวหนึ่งของเทคโนโลยีที่มีผู้ให้  
ความสนใจมาก

แต่เดิมนั้นการนำสินค้าเข้าหรือออกจากคลังสินค้าจะใช้แต่แรงงานมนุษย์  
เท่านั้น จะเห็นได้ว่าเป็นการสิ้นเปลืองทรัพยากรมนุษย์มากนอกจากนี้แรง  
งานที่ได้ยังจำเป็นต้องพักผ่อนและยังเป็นแรงงานที่ก่อให้เกิดปัญหาอีกด้วย

ดังนั้นการนำเอาเทคโนโลยี WARE HOUSE เข้ามาใช้จึงนับเป็นการ  
แก้ไขปัญหานี้ได้อย่างสิ้นเชิงเพราะ เทคโนโลยี WARE HOUSE เป็นการนำ  
เครื่องจักรกลมาประยุกต์ใช้กับระบบฐานข้อมูลโดยมีคอมพิวเตอร์เป็นตัวกลางจึง  
นับได้ว่าเทคโนโลยี WARE HOUSE เป็นเทคโนโลยีที่น่าจับตามองในอนาคต

เนื้อหาของปฏิญานิพนธ์ฉบับนี้จะเป็นการเขียนโปรแกรมเพื่อควบคุมระ  
บบ STOCK ของสินค้าหรือ เครื่องมือ โดยจะใช้รูปแบบของระบบฐานข้อมูล  
ซึ่งมีในหลักสูตรของคณะวิศวกรรมศาสตร์ และผู้จัดหวังว่าคงเป็นประโยชน์แก่นัก  
ศึกษารุ่นน้อง ๆ ต่อไป ที่ท้ายที่สุดนี้คณะผู้จัดขอขอบคุณอาจารย์ทุกท่านที่ได้สั่งสอน  
เพิ่มพูนความรู้แก่คณะผู้จัดทำทุกคน

คณะผู้จัดทำ

ขอบเขตของปริญาพันธ

1. สามารถส่งรหัสไปที่ไมโครโปรเซสเซอร์เพื่อควบคุมการทำงานของ  
แขนกล
2. สามารถเพิ่มหรือลดจำนวนสินค้าใน DATA BASE เมื่อมีการส่งรหัส  
ให้นำสินค้าเข้าหรือออก
3. สามารถเพิ่มหรือลดชนิดของสินค้าได้
4. สามารถเรียกดูข้อมูลของสินค้า โดยการกำหนดรหัสของสินค้า
5. สามารถแก้ไขข้อมูลของสินค้าได้ โดยการกำหนดรหัสของสินค้า
6. เมื่อเพิ่มหรือลดจำนวนสินค้าผลรวมเกี่ยวกับสินค้าต้องเปลี่ยนแปลงตาม  
เช่น ราคารวม เป็นต้น
7. สามารถเปิดฐานข้อมูลใหม่เพื่อใช้ควบคุม WAREHOUSE ตัวอื่นได้

บทที่ 1

พื้นฐานความรู้เกี่ยวกับการสื่อสารข้อมูลและฐานข้อมูล

ในการที่เราจะนำ COMPUTER มาใช้ในการสั่งให้ WARE HOUSE ทำงานที่เราต้องการได้นั้น จะต้องมียุคสมัยที่สนับสนุนอยู่คือ

1. COMPUTER ที่มีหน้าที่ในการเก็บข้อมูลของสินค้าใน WARE HOURE และทำการสั่งให้ WARE HOUSE นำข้อมูลเข้าหรือนำข้อมูลออกได้
2. ในส่วนของ WARE HOUSE จะต้องมีส่วนของ HARD WARE ในการรับคำสั่งจาก COMPUTER และนำไปปฏิบัติได้
3. จะต้องมีการกำหนด รหัส (CODE) ในการติดต่อสื่อสารระหว่าง WARE HOUSE และ COMPUTER

การสื่อสารข้อมูล

การสื่อสารข้อมูลในปัจจุบันนี้สามารถแบ่งออกได้ 2 ประเภทดังนี้

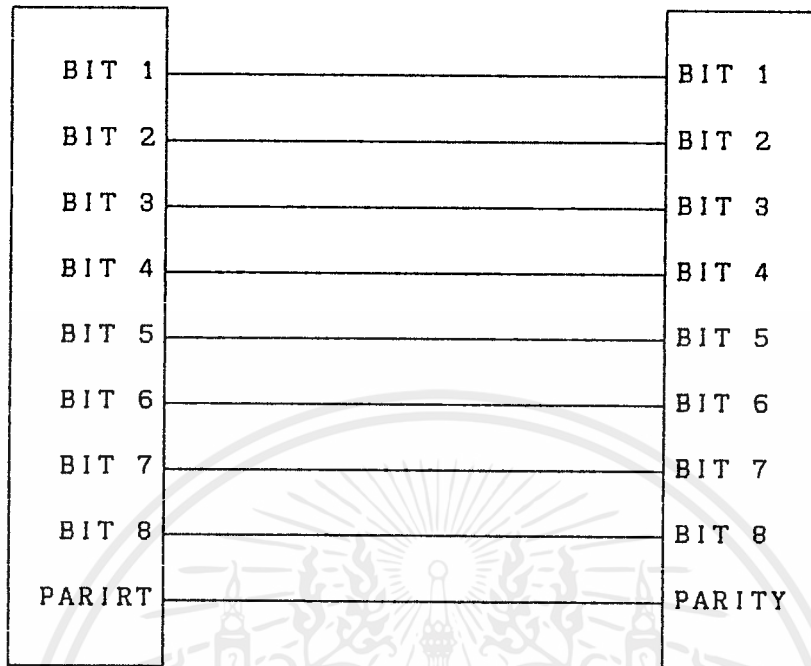
1. การสื่อสารข้อมูลแบบขนาน

การสื่อสารข้อมูลแบบนี้ ทำได้โดยการส่งข้อมูลออกมาครั้งละ 1 BYTE หรือ 8 BIT จากตัวส่งไปยังตัวรับดังนั้นตัวกลางที่จะทำการส่งและรับข้อมูลจึงต้องใช้สายในการส่งข้อมูลครั้งละ 8 เส้น (ในกรณีใช้เพียง 8 BIT)

ข้อดีของการสื่อสารข้อมูลแบบนี้ คือ ความเร็วของการส่งข้อมูลที่เด่นชัดที่สุด ส่วนข้อดีอีกข้อหนึ่งก็เห็นจะเป็นความประหยัด ส่วนใหญ่แล้วตัวกลางของการสื่อสารข้อมูลวิธีนี้ก็คือ สายสัญญาณ จึงทำให้ประหยัด

ส่วนข้อเสียของการสื่อสารข้อมูลแบบนี้คือความผิดเพี้ยนของ ข้อมูลระหว่างการสื่อสารเนื่องจากความต้านทานของสายสัญญาณจึงทำให้สื่อสารข้อมูลได้ในระยะใกล้ (ไม่ควรเกิน 100 ฟุต) นอกจากนี้การใช้สายสัญญาณหลายๆในระยะทางที่ไกลๆจะเกิดความสับสนได้ง่าย

การสื่อสารข้อมูลแบบนี้นอกจากจะใช้สายข้อมูลแล้วยังอาจต้องมีสายสัญญาณอื่นๆเพิ่มอีก เช่น สายสัญญาณที่แสดงพาริตี กล่าวคือต้องมีจำนวนของสายเท่ากับจำนวน BIT ของการสื่อสาร



อุปกรณ์รับส่งข้อมูล

อุปกรณ์รับส่งข้อมูล

การสื่อสารข้อมูลแบบขนาน

## 2. การสื่อสารข้อมูลแบบอนุกรม

การสื่อสารข้อมูลแบบนี้ทำการส่งหรือรับข้อมูลครั้งละ 1 BIT จึงช้ากว่าการสื่อสารข้อมูลแบบขนาน การสื่อสารข้อมูลแบบนี้ก่อนที่จะทำการส่งหรือรับข้อมูลต้องทำการเรียงข้อมูลก่อนส่งข้อมูลและทำการแปลงข้อมูลกลับเมื่อทำการรับข้อมูลเสมอ

ข้อดีของการสื่อสารข้อมูลแบบนี้ที่เห็นได้ชัดก็คือความเที่ยงตรงของรหัสข่าวสารในระหว่างการสื่อสาร นอกจากนี้ยังสามารถสื่อสารข้อมูลได้ในระยะทางไกลๆ และข้อดีที่เห็นได้เด่นชัดก็คือความประหยัดของสายสัญญาณในการใช้สื่อสารข้อมูล กล่าวคือเราสามารถสื่อสารข้อมูลได้หลายๆ BIT โดยใช้สายสัญญาณเพียง 3 เส้นเท่านั้น คือสายของสัญญาณในภาคส่งข้อมูล 1 เส้น สายของสัญญาณในภาครับข้อมูล 1 เส้นและสายดิน(GND)

ส่วนข้อเสียของการสื่อสารข้อมูลแบบนี้ก็ได้แก่ความยุ่งยากในการแปลงข้อมูลจากแบบขนานเป็นแบบอนุกรม

การสื่อสารข้อมูลแบบนี้ มิได้ส่งหรือรับเฉพาะข้อมูลข่าวสารเท่านั้นแต่ยังต้องส่งสถานะต่างๆของข้อมูลอีกด้วยสถานะที่กล่าวมาในข้างต้นก็ได้แก่สัญญาณ START BIT, STOP BIT และสัญญาณ อื่นๆที่เป็นตัวตรวจสอบความผิดพลาดของข้อมูลด้วย เช่น พาริตีบิต เป็นต้น

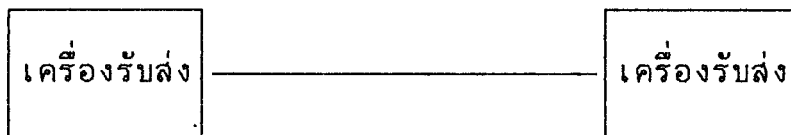
เราอาจแบ่งลักษณะของการสื่อสารข้อมูลแบบอนุกรมได้เป็น 3 แบบคือ

1. แบบซิมเพล็กซ์ (SIMPLEX) การสื่อสารข้อมูลแบบนี้ทำได้เพียงทางเดียวเท่านั้นดังรูป



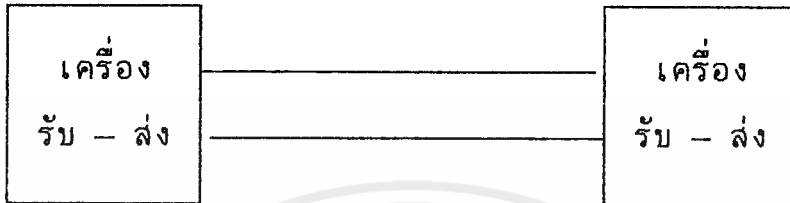
การสื่อสารข้อมูลแบบซิมเพล็กซ์

2. แบบฮาล์ฟดูเพล็กซ์ (HALF DUPLEX) การสื่อสารข้อมูลแบบนี้สามารถจะรับและส่งข้อมูลกันได้แต่ต้องผลัดกันรับและส่งข้อมูลจะส่งและรับข้อมูลพร้อมกันไม่ได้



การสื่อสารข้อมูลแบบฮาล์ฟดูเพล็กซ์

3. แบบพูลดิวเพล็กซ์ (FULL DUPLEX) การสื่อสารข้อมูลแบบนี้สามารถที่จะรับและส่งข้อมูลได้พร้อมๆกันได้



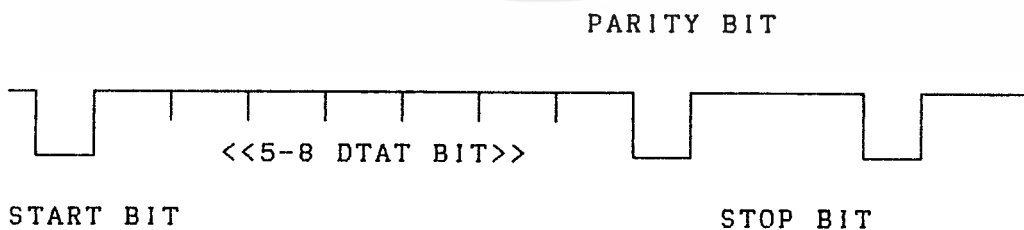
การสื่อสารข้อมูลแบบนี้จะเน้นหนักที่ระยะทางการสื่อสารและความถูกต้องระหว่างการส่งข้อมูลดังนั้นจึงต้องมีการป้องกันความผิดพลาดในการส่งข้อมูล

การป้องกันความผิดพลาดของการส่งข้อมูลแบบนี้มีด้วยกัน 2 วิธีคือ

1. การสื่อสารข้อมูลแบบซิงโครนัส
2. การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบนี้นอกจากจะส่งสัญญาณข้อมูลแล้วยังต้องเพิ่ม บิตตรวจสอบขึ้นอีกด้วยคือ บิตเริ่มต้น (START BIT) และบิตสิ้นสุด (STOP BIT)



รูปแบบการสื่อสารข้อมูลแบบอะซิงโครนัส

ขณะที่การส่งเป็นแบบว่างคือยังไม่มีสัญญาณออกมาจะมีสัญญาณหรือแรงดันตลอดเวลาเพื่อป้องกันสัญญาณรบกวนในขณะที่ยังไม่มีการส่งหรือรับข้อมูล เมื่อเริ่มเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะส่งข้อมูลสัญญาณของ อะซิงโครนัสจะเป็น 0 อยู่ช่วงหนึ่งสัญญาณสัญญาณนาฬิกาโดยเราเรียกสัญญาณ 0 ในช่วงหนึ่งสัญญาณนาฬิกาว่า บิตเริ่มต้นหลังจากเกิดสัญญาณนี้แล้วรหัสที่ตามมาคือรหัสของข้อมูลซึ่งจะมี 5-8 บิตก็แล้วแต่โดยเราจะให้บิตแรกที่เข้ามาเป็นบิตที่มีเลขนัยสำคัญต่ำสุด รหัสสุดท้ายของข้อมูลที่เข้ามาจะให้ เป็นบิตที่มรเลขนัยสำคัญสูงสุดหลังจากรหัสของข้อมูลแล้วเราจะกำหนดให้บิตที่ตามมาเป็น 0 หรือ 1 ก็ได้แล้วแต่การออกแบบการตรวจสอบโดย

เรียกบิตที่ตามมานี้ว่า บิตตรวจสอบหรือ พาริตีบิต(PARITY BIT)

ในการเริ่มใช้ พอร์ทแบบอนุกรมจำเป็นต้องตั้งค่าต่างๆสำหรับการส่งแบบอนุกรมดังนี้

1. ความเร็วในการส่ง
2. ความยาวรหัส 1 อักขระ
3. บิตตรวจสอบ
4. จำนวน STOP BIT

สำหรับคอมพิวเตอร์ความเร็วในการส่งมีให้เลือกตั้งแต่ 110, 200, 300, 1200, 2400, 4800, 9600 บอด

#### ความเร็วในการถ่ายโอนข้อมูลแบบอนุกรม

หน่วยวัดเป็นบิตต่อวินาที (bps) หน่วยที่บรรยายถึงการเปลี่ยนแปลงสัญญาณใน 1 วินาที เรียกว่า บอดเรต (BAUD RATE) หรือ อัตราบอด การเปลี่ยนแปลงของสัญญาณ 1 ครั้งอาจจะแสดงถึง การส่งข้อมูลแบบอนุกรมมากกว่า 1 บิต

อัตราบิตสามารถหาได้จากสมการดังนี้

$$\text{อัตราบิต (BIT RATE)} = \text{อัตราบอด (BAUD RATE)} * \text{บิตใน 1 บอด}$$

ในปฏิญานินพนธ์ฉบับนี้เราจะใช้การ์ดอะแดปเตอร์สื่อสาร ซึ่งจะมี PORT ในการสื่อสารแบบอนุกรม 2 PORT คือ PORT COM 1 และ COM2 เราสามารถจะเลือก PORT ใดก็ได้ อะแดปเตอร์นี้เป็น อะซิงโครนัส อะแดปเตอร์ (ASYNCHRONOUS COMMUNICATION ADAPTER) ซึ่งจะใช้ RS-232 ซึ่ง

RS-232 จะทำหน้าที่รับส่งข้อมูลแบบอนุกรมเรียกว่า UNIVERSAL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ASYNCHRONOUS ADAPTER ไม่ว่าจะผิดใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่สำคัญของการสื่อสารแบบอะซิงโครนัสคือ

หน้าที่ในด้านการรับข้อมูล

1. เปลี่ยนสัญญาณที่เข้ามาแบบอนุกรมให้เป็นแบบขนาน
2. ตรวจสอบความผิดพลาดของสัญญาณที่รับเข้ามา
3. ตัด STOP BIT และ PARITY BIT ออก
4. ส่งสัญญาณให้ CPU รู้ว่ารับสัญญาณแล้ว

หน้าที่ในด้านการส่งข้อมูล

1. เปลี่ยนสัญญาณแบบขนานจาก CPU ให้เป็นแบบอนุกรม
2. เพิ่ม START BIT และ STOP BIT เข้ามา
3. เพิ่มสัญญาณควบคุมโมเด็มที่ต่อเชื่อม

ในการส่งข้อมูลทั้งหมดเราจะใช้ การ์ดอะแดปเตอร์ (RS-232) โดยจะแบ่ง PORT ต่างๆดังนี้

- COM 1 ADDRESS 3F8H
- COM 2 ADDRESS 2F8H

ซึ่งสามารถจะโปรแกรมในแต่ละ BIT ไว้ดังนี้

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

บิตที่ 1-0 เป็นความยาวของ CHARACTER)

00 = 5 BIT

01 = 6 BIT

10 = 7 BIT

11 = 8 BIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิตที่ 2 เป็นจำนวนของ STOP BIT

0 = 1 BIT

1 = 1.5 OR 2 BIT

บิตที่ 3-4 เป็นการตั้งค่า PARITY

00 OR 11 = NO USE

01 = Odd

10 = EVEN

บิตที่ 5-7 เป็นการตั้งค่า BAUD RATE

000 = 110 bps

001 = 150 bps

010 = 300 bps

011 = 600 bps

100 = 1200 bps

101 = 2400 bps

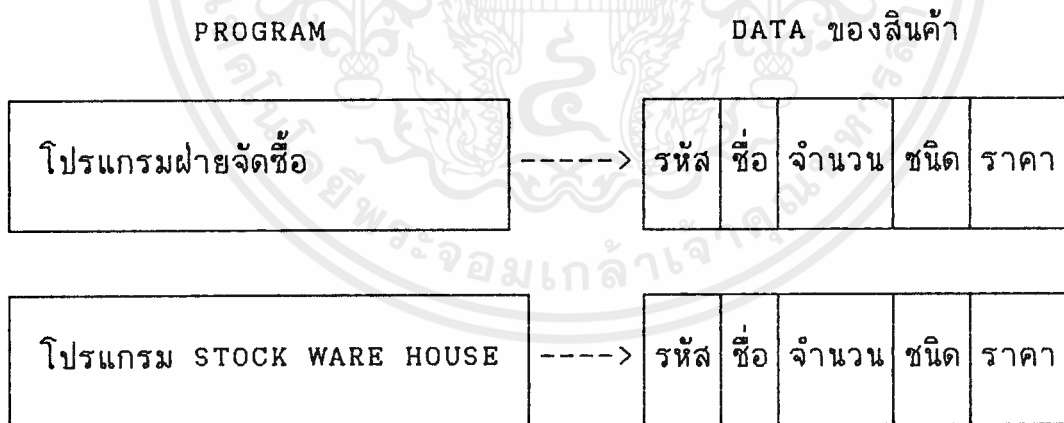
110 = 4800 bps

111 = 9600 bps

ในที่นี้เราจะใช้ FUNCTION ของ DOS และ INTERRUPT FUNCTION มาใช้ในการควบคุมการสื่อสารข้อมูลของ SERIAL PORT

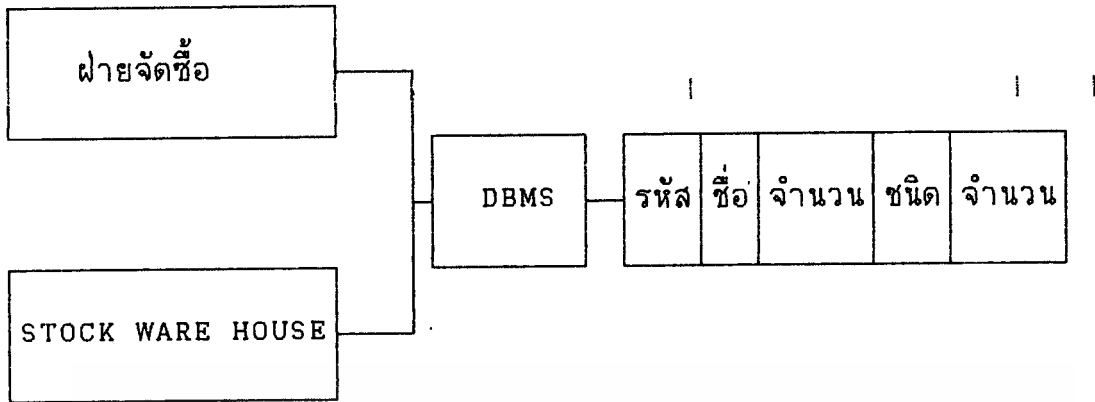
### ระบบฐานข้อมูล

ก่อนที่จะกล่าวถึงระบบฐานข้อมูล จำเป็นที่จะต้องกล่าวกลับไปถึงการประมวลผลระบบไฟล์ทั่ว ๆ ไปก่อน ลักษณะของการประมวลผลแบบไฟล์นั้น การทำงานจะเป็นลักษณะของการแยกกันทำงานอย่างเด็ดขาด แทบจะไม่มีส่วนใดที่ทำงานร่วมกันได้เลย ไม่ว่าจะเป็นตัวโปรแกรมหรือข้อมูล ในช่วงแรกของการนำคอมพิวเตอร์มาใช้งาน การประมวลผลระบบไฟล์นิยมนำมาใช้มาก แต่เมื่อคอมพิวเตอร์ถูกนำมาใช้งานเพิ่มมากขึ้น ระบบไฟล์ก็ก่อให้เกิดปัญหาโดยเฉพาะข้อมูลที่ต้องงานร่วมกัน เช่น ระบบ STOCK ของ WARE HOUSE ถ้านำระบบไฟล์มาใช้ข้อมูลของฝ่ายจัดซื้อและฝ่าย STOCK WARE HOUSE จะเกิดปัญหาทันที ปัญหาแรกที่จะเกิดคือ ฝ่ายจัดซื้อต้องไปเช็คจำนวนสินค้าที่ระบบ WARE HOUSE ทุกครั้งที่จะจัดซื้อ หรือ WARE HOUSE ต้องไปเช็คจำนวนสินค้าจากฝ่ายจัดซื้อว่าซื้อมามากน้อยเท่าไร จากสองตัวอย่างที่กล่าวมา จะเห็นว่าเกิดปัญหาแล้ว นอกจากนี้ยังมีปัญหาในเรื่องขนาดของข้อมูลที่อยู่ใน Memory อีกด้วยเพราะข้อมูลซ้ำซ้อนกันดังรูป



จากตัวอย่างที่ยกมา จะเห็นว่าระบบฐานข้อมูล สามารถเข้ามาแก้ปัญหาส่วนนี้ได้ เพราะระบบฐานข้อมูลมีหลักการอยู่ที่การนำข้อมูลที่ใช้ร่วมกันมารวมกัน โดยมีระบบ DATA BASE MANAGEMENT SYSTEM หรือ DBMS เป็นตัว

### บริหารข้อมูล ดังรูป



จะเห็นว่าระบบฐานข้อมูลสามารถแก้ปัญหาที่เกิดขึ้นจากระบบไฟล์ได้ ทำให้การทำงานเร็วยิ่งขึ้นและไม่สิ้นเปลืองหน่วยความจำ ระบบฐานข้อมูลนั้นจะมีปัญหาอยู่บ้างตรงที่การจัดระบบกระทำได้ยาก แต่ถ้าจะกล่าวถึงประโยชน์ของระบบฐานข้อมูลซึ่งมีมากมายดังนี้

1. ลดความซ้ำซ้อนของข้อมูล (redundancy can be reduced) จะเห็นได้ว่าการประมวลผลโดยใช้ไฟล์ธรรมดา นั้น จำเป็นที่ผู้ใช้แต่ละกลุ่มจะต้องมีไฟล์ส่วนตัวเอาไว้ ดังนั้นจึงเกิดเหตุการณ์ที่ข้อมูลชนิดเดียวกันถูกเก็บไว้หลาย ๆ แห่ง หรือที่เราเรียกกันว่าความซ้ำซ้อน การนำข้อมูลทั้งหมดมาเก็บไว้ในที่เดียวกันในฐานข้อมูลนี้เป็นการลดความซ้ำซ้อนลงไปได้ แต่อย่างไรก็ตามก็ยังมีงานบางประเภทอาจต้องเก็บข้อมูลชุดเดียวกันไว้มากกว่าหนึ่งแห่ง ถึงแม้จะมีความซ้ำซ้อนอยู่บ้าง ระบบฐานข้อมูลก็จะทราบอยู่ตลอดเวลาว่ามีความซ้ำซ้อนอยู่ที่ใดบ้าง

2. สามารถหลีกเลี่ยงความขัดแย้งของข้อมูลได้ในระดับหนึ่ง (Inconsistency can be avoided to some extent) ประโยชน์ในข้อนี้ก็สืบเนื่องมาจากในข้อแรก เพราะอย่างที่เรากล่าวถึงแล้วว่า การเก็บข้อมูลไว้หลาย ๆ แห่ง ทำให้เกิดปัญหาว่าข้อมูลชุดเดียวกัน อาจมีค่าแต่ละแห่งไม่ตรงกัน ดังนั้นระบบฐานข้อมูลก็จะช่วยควบคุมดูแลให้ข้อมูลเหมือนกันทุกแห่ง

3. สามารถใช้ข้อมูลร่วมกันได้ (The Data can be Shared) การใช้ข้อมูลร่วมกันได้นี้ ไม่จำกัดอยู่เฉพาะโปรแกรมที่ใช้ข้อมูลอยู่ในปัจจุบันเท่านั้น แต่กินความถึงโปรแกรมประยุกต์ที่จะพัฒนาขึ้นมาใหม่ด้วย ที่สามารถใช้ข้อมูลที่มีอยู่แล้วได้เลย โดยไม่จำเป็นต้องเพิ่มข้อมูลเข้าไปอีก

4. สามารถควบคุมความเป็นมาตรฐานได้ (Standards can be Enforced) จากการที่เรานำข้อมูลมาเก็บรวมกันไว้ในฐานข้อมูลเช่นนี้ ทำให้ผู้มีหน้าที่ควบคุมดูแลการใช้ระบบฐานข้อมูลสามารถกำหนดมาตรฐานของข้อมูลขึ้นมาได้เช่น ให้ใช้หน่วยมาตรการวัดที่เหมือนกัน รูปแบบในการเขียนวันที่ให้เหมือนกัน เป็นต้น ซึ่งการที่ข้อมูลใช้มาตรฐานเดียวกันเช่นนี้จะทำให้การแลกเปลี่ยนข้อมูลเป็นไปอย่างถูกต้องและรวดเร็ว

5. สามารถจัดหาระบบความปลอดภัยที่รัดกุมได้ (Security restrictions can be applied) คำว่าระบบความปลอดภัยในที่นี้หมายถึง การป้องกันไม่ให้ผู้ใช้ที่ไม่มีสิทธิมาใช้ข้อมูลในระบบได้ ซึ่งการรักษาความปลอดภัยที่รัดกุมนี้จะถูกกำหนดโดย ความสามารถในการควบคุมการเป็นมาตรฐาน

6. สามารถควบคุมความคงสภาพของข้อมูลได้ (Integrity can be Maintained) จะขอยกตัวอย่างเกี่ยวกับความสามารถควบคุมความคงสภาพของข้อมูล เช่น ข้อมูลเกี่ยวกับอายุของพนักงานในบริษัทอาจมีค่า 300 แทนที่จะเป็น 30 ซึ่งความผิดพลาดแบบนี้เกิดขึ้นได้ง่ายจากความสับสนในการพิมพ์ข้อมูลก็ได้ ในลักษณะของความไม่ถูกต้องเช่นนี้ ผู้ออกแบบระบบฐานข้อมูลสามารถใส่กฎเกณฑ์ เพื่อควบคุมความคงสภาพไว้ เช่นตามตัวอย่างนี้ก็อาจใส่กฎว่า ค่าอายุจะต้องเป็นตัวเลขระหว่าง 16-60 เป็นต้น ดังนั้นเมื่อมีการใส่ข้อมูลใหม่หรือแก้ไขข้อมูล ระบบฐานข้อมูลก็จะควบคุมดูแลให้ข้อมูลดังกล่าวถูกต้องตามกฎเกณฑ์

7. สามารถสร้างสมดุลในความขัดแย้งของความต้องการได้ (Conflicting Requirements can be Balanced) การที่ผู้ใช้ทั้งหมดขององค์กรใช้ข้อมูลจากฐานข้อมูลร่วมกันเช่นนี้ ทำให้ระบบฐานข้อมูลทราบถึงความต้องการและความสำคัญของผู้ใช้งานทั้งหมด จึงสามารถกำหนดโครงสร้างของฐานข้อมูล เพื่อให้บริการที่ดีที่สุดได้ เช่น เลือกเก็บข้อมูลที่ต้องใช้บ่อยไว้ในสื่อข้อมูลที่มีความเร็วเป็นพิเศษ เป็นต้น เป็นการสร้างสมดุลของความต้องการไม่เกิดความขัดแย้งในหมู่ผู้ใช้ เพราะการออกแบบนั้นกระทำบนแนวทางที่มุ่งจะให้ประโยชน์ส่วนรวมดีที่สุดแล้ว

8. เกิดความเป็นอิสระของข้อมูล (Data Independence) ลักษณะของข้อมูลที่ไม่เป็นอิสระคือ ข้อมูลที่ถูกนำมาประยุกต์ใช้ยังมีความผูกพันอยู่กับวิธีการจัดเก็บและเรียกใช้ข้อมูล ซึ่งในลักษณะการเขียนโปรแกรมประยุกต์บางประเภทเราอาจจำเป็นต้องใส่เทคนิคการจัดเก็บและเรียกใช้ข้อมูลไว้ในตัวโปรแกรมด้วย ซึ่งก็หมายความว่าถ้าต้องมีการเปลี่ยนแปลงวิธีจัดเก็บและการเรียกใช้ข้อมูลแล้ว ผู้ใช้ก็จำเป็นจะต้องสร้างวิธีการประยุกต์ใช้ขึ้นมาใหม่ ซึ่งเป็นความไม่สะดวกอย่างยิ่งและทำให้หมดโอกาสที่จะปรับปรุงโครงสร้างของข้อมูล เพื่อให้ใช้งานได้อย่างมีประสิทธิภาพมากขึ้น

### ระดับของข้อมูล

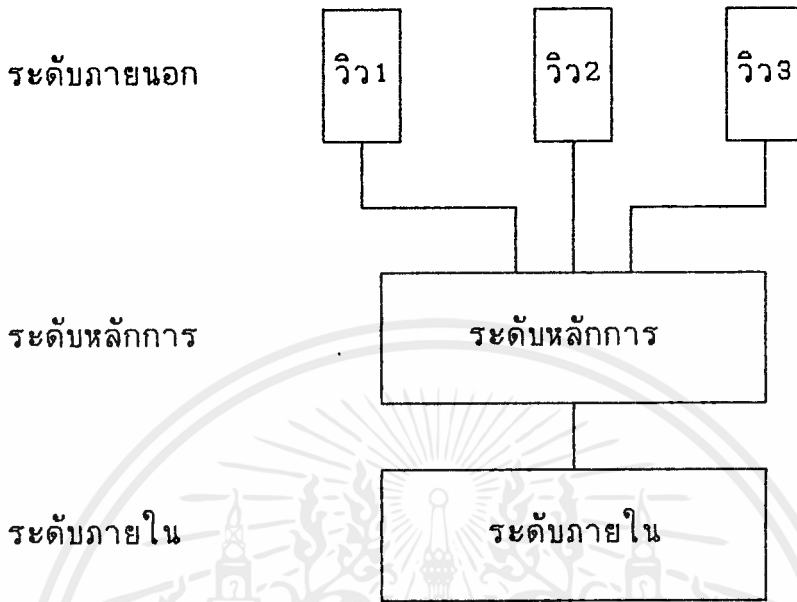
สาเหตุสำคัญของการให้ความสำคัญของข้อมูลเป็นลำดับก็เป็นเพราะว่าในระบบฐานข้อมูลนั้นผู้ใช้แต่ละคนจะถูกกำหนดให้ใช้ข้อมูลเท่าที่จำเป็นเท่านั้น หรืออาจกล่าวอีกนัยหนึ่งว่าเป็นการรักษาความปลอดภัยของข้อมูลแบบหนึ่งนั่นเอง การแบ่งระดับของข้อมูลในระบบฐานข้อมูลจะแบ่งเป็น 3 ระดับ คือ

-ระดับภายใน (Internal) เป็นระดับต่ำที่สุด อันได้แก่ระดับของการจัดเก็บข้อมูลจริง ๆ

-ระดับหลักการ (Conceptual Level) เป็นระดับที่อยู่ถัดขึ้นมาได้แก่ระดับการมอง entity และความสัมพันธ์ระหว่าง entity ทั้งหมดรวมทั้งกฎเกณฑ์ต่าง ๆ เกี่ยวกับผู้มีสิทธิจะใช้ ฯลฯ

-ระดับภายนอก (External) เป็นระดับที่อยู่สูงที่สุดที่จะสามารถมองเห็นจากการใช้งานของผู้ใช้แต่ละคน

ซึ่งความสัมพันธ์ของระดับต่าง ๆ จะสามารถแสดงได้ดังตาราง



ระบบการจัดการฐานข้อมูล

ในการใช้ระบบฐานข้อมูลนั้นผู้ใช้ไม่จำเป็นต้องทราบรายละเอียดเกี่ยวกับการจัดเก็บข้อมูลจริง ๆ แต่เป็นหน้าที่ของระบบการจัดการฐานข้อมูลซึ่งจะเป็นตัวจัดการโดยระบบการจัดการฐานข้อมูลนั้นจะมีหน้าที่ดังนี้

1. ทำหน้าที่ติดต่อกับตัวจัดการระบบไฟล์ เนื่องจากการใช้งานส่วนใหญ่ของระบบฐานข้อมูล คือลักษณะการใช้งานกับข้อมูลขนาดใหญ่ ดังนั้นระบบการจัดการฐานข้อมูลก็จะทำการประสานงานกับตัวจัดการระบบไฟล์ในการจัดเก็บ เรียกใช้และแก้ไขข้อมูล

2. การควบคุมความคงสภาพ เป็นการป้องกันความผิดพลาดของข้อมูลจากผู้ใช้ เช่น การใส่ข้อมูลผิดโดยเฉพาตัวเลข 300 ซึ่งที่ถูกเป็น 30 ระบบการจัดการฐานข้อมูลก็จะช่วยดูแลส่วนนี้ด้วย

3. การควบคุมระบบความปลอดภัย ได้แก่การป้องกันมิให้ผู้ใช้ไม่มีสิทธิใช้เข้ามารับรู้หรือแก้ไขข้อมูล

4. การสร้างระบบสำรองและการฟื้นฟูสภาพ ได้แก่การจัดทำไฟล์สำรองข้อมูล เมื่อเกิดปัญหาต่าง ๆ ขึ้นเพื่อให้ระบบข้อมูลกลับสู่สภาวะปกติได้เร็วที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. การควบคุมสภาวะพร้อมกัน เป็นหน้าที่ที่สำคัญมากอย่างหนึ่งของระบบการจัดการฐานข้อมูล ในกรณีที่มีผู้ใช้ข้อมูลพร้อม ๆ กันแล้วมีข้อมูลบางส่วนที่ทำการแก้ไขข้อมูลอยู่ระบบการจัดการฐานข้อมูลจะไม่ยอมให้ผู้ใช้ผู้อื่นเข้ามาแก้ไขข้อมูลนั้นได้

### ผู้ใช้ระบบ

ระบบฐานข้อมูลนั้นถูกสร้างขึ้นเพื่อ อำนวยความสะดวกแก่ผู้ใช้นั้นระบบการจัดการฐานข้อมูลจึงแบ่งผู้ใช้ออกเป็นกลุ่ม 4 กลุ่มดังนี้

1. ผู้เขียนโปรแกรมประยุกต์ ได้แก่โปรแกรมเมอร์ผู้ซึ่งกลุ่มนี้จะเป็นผู้ที่สร้างโปรแกรมด้วยภาษาชั้นสูงโดยเราจะเรียกโปรแกรมที่ถูกผู้ซึ่งกลุ่มนี้เขียนขึ้นว่าเป็นโปรแกรมหลัก

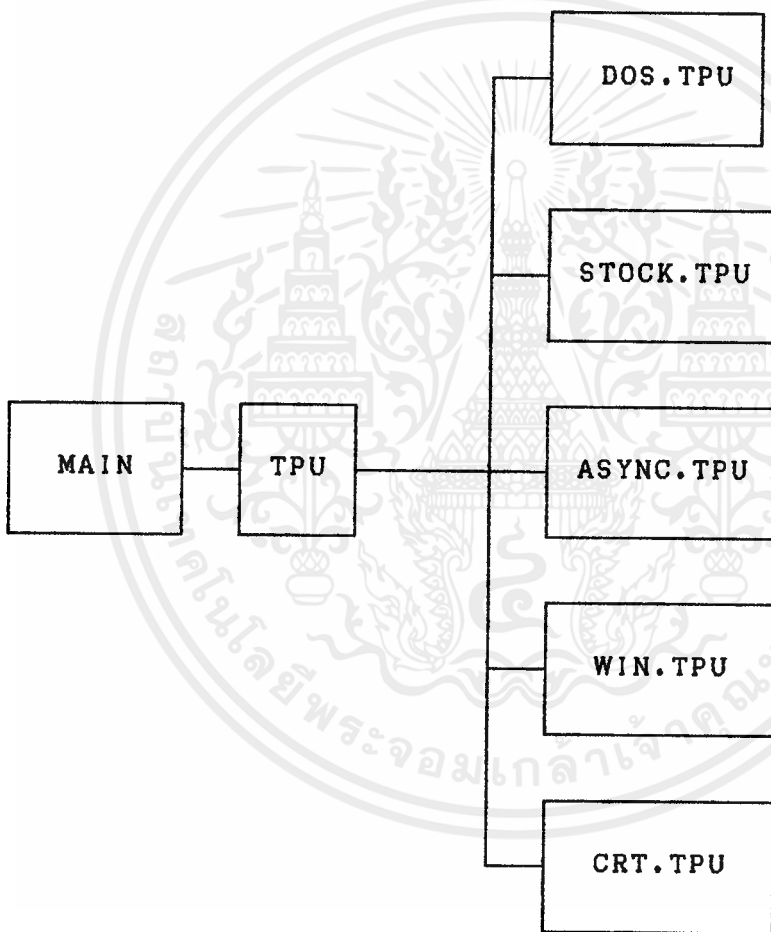
2. ผู้ใช้ทั่วไป ได้แก่ผู้ที่เรียกใช้ข้อมูลโดยผ่านทางโปรแกรมหลัก เท่านั้น ผู้ใช้เหล่านี้ไม่จำเป็นต้องรู้เกี่ยวกับการเขียนโปรแกรม เนื่องจากคำสั่งของโปรแกรมหลักจะถูกออกแบบมาให้ใช้งานง่ายอยู่แล้ว

3. โอเพอร์เรเตอร์ คือกลุ่มผู้ซึ่งจะทำการใส่ข้อมูลอย่างเดี่ยว ผู้ซึ่งกลุ่มนี้ไม่จำเป็นต้องมีความรู้ใดๆ ทั้งสิ้นที่เกี่ยวกับระบบคอมพิวเตอร์ ลักษณะการทำงานได้แก่การป้อนข้อมูลตามที่จอภาพระบุไว้เท่านั้น

4. ผู้ใช้พิเศษ คือกลุ่มผู้ซึ่งนำระบบข้อมูลไปใช้ในลักษณะงานที่ไม่ใช่งานในลักษณะการประมวลผลข้อมูล เช่น การสร้างระบบฐานรอบรู้และระบบผู้เชี่ยวชาญ เป็นต้น

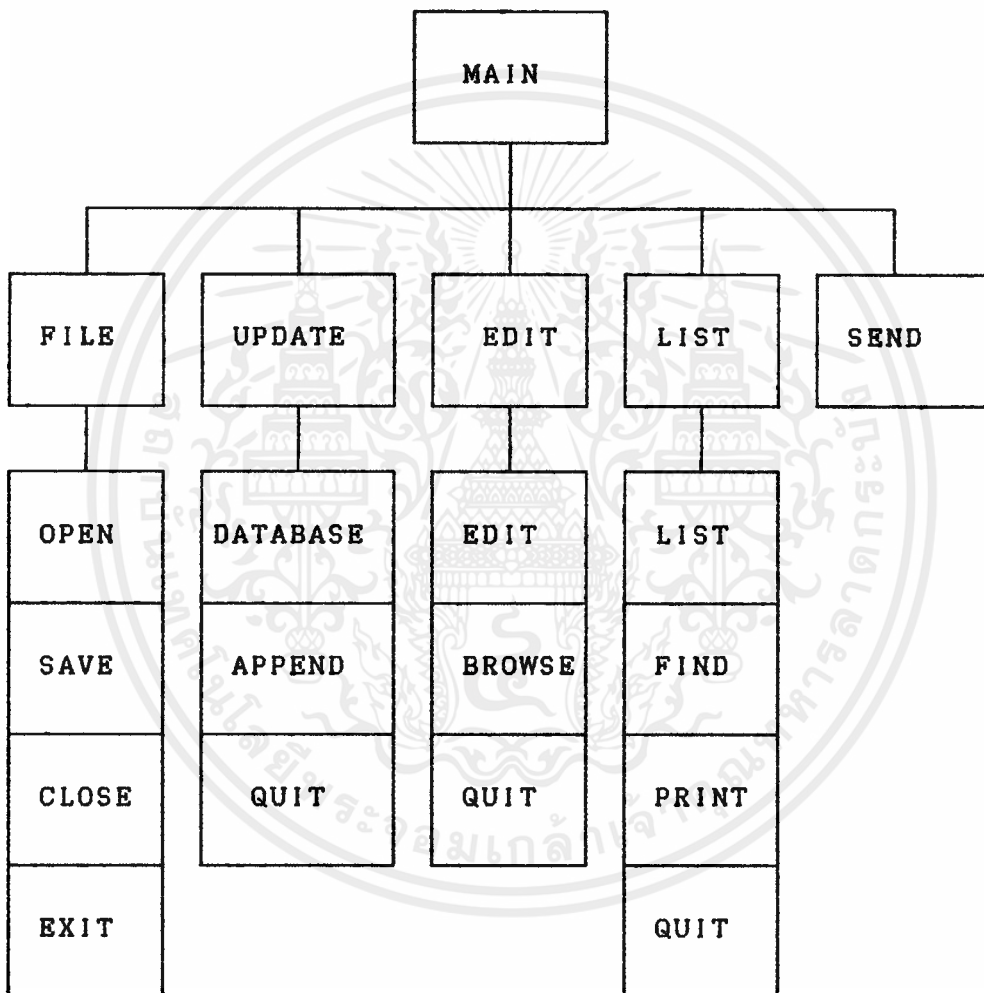
โครงสร้างโปรแกรม

BOLCKDIAGRAM แสดงการเรียกใช้ TURBO UNIT



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BLOCKDIAGRAM แสดงการติดต่อกับโปรแกรมย่อยต่าง ๆ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การทำงานของ PROCEDURE ต่างๆในไฟล์หลัก

โปรแกรมย่อยที่ใช้ในโปรแกรม STOCKW.PAS

PROCEDURE INVERSE;

เป็นโปรแกรมย่อยในการทำให้สีของพื้น และสีของตัวอักษรสลับกัน

PROCEDURE NORMAL;

เป็นโปรแกรมย่อยที่ทำให้สีพื้นและสีของตัวอักษรกลับเป็นเหมือนเดิม มักจะใช้ร่วมกับ โปรแกรมย่อย INVERSE และมักจะใช้เกี่ยวกับแถบแสง

PROCEDURE TABLE(C1,R1,C2,R2:INTEGER);

เป็นโปรแกรมย่อยในการสร้างกรอบเส้นคู่

PROCEDURE TABLE1(C1,R1,C2,R2 :INTEGER);

เป็นโปรแกรมย่อยในการสร้างกรอบเส้นเดี่ยว

PROCEDURE CLEARWIN(D1,D2,D3,D4:INTEGER);

เป็นโปรแกรมย่อยในการลบกรอบของหน้าต่างที่ถูกสร้างขึ้นโดย โปรแกรมย่อย TABLE และ TABLE1

FUNCTION LOOP:STRING;

เป็นโปรแกรมที่ใช้ตรวจสอบตัวอักษรว่าขณะนี้ใส่ตัวอักษรเกินจำนวนที่กำหนดแล้วหรือยัง

PROCEDURE LOOP1;

เป็นโปรแกรมที่ใช้กำหนดค่าของตัวเลขไว้เพื่อไม่ให้เกินค่าที่กำหนดไว้

PROCEDURE OPEN\_F;

เป็นโปรแกรมที่ใช้เปิดแฟ้มข้อมูลที่มีนามสกุลเป็น .WHP และโหลดข้อมูล  
สู่หน่วยความจำ

FUNCTION CHECK\_FILE\_OPEN:BOOLEAN;

เป็นโปรแกรมที่ใช้สำหรับตรวจสอบว่าสามารถที่จะเปิดแฟ้มข้อมูลได้หรือ  
ไม่ และใช้ตรวจสอบว่าแฟ้มข้อมูลนี้ถูกสร้างขึ้นหรือยัง

PROCEDURE FOPEN;

เป็นโปรแกรมที่ใช้สร้างไฟล์ครั้งแรก

PROCEDURE INP\_DAT(VAR I : INTEGER);

เป็นโปรแกรมที่ใช้ในการเก็บข้อมูลเป็น RECORD

PROCEDURE WRITEF(VAR I : INTEGER);

เป็นโปรแกรมที่ใช้ในการเก็บข้อมูลลงสู่แผ่นดิสก์

PROCEDURE CLOSEF;

เป็นโปรแกรมที่ใช้ในการปิดแฟ้มข้อมูล

PROCEDURE CREATE-1-2;

เป็นโปรแกรมที่เรียกใช้โปรแกรมย่อย FOPEN

PROCEDURE APPEND;

เป็นโปรแกรมที่ใช้ในการเพิ่มเติม RECORD ในแฟ้มข้อมูลที่ยังมี RECORD  
ไม่ครบ 21 RECORD

PROCEDURE EDIT;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในหน่วยงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
เป็นโปรแกรมที่ใช้ในการแก้ไขข้อมูล  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROCEDURE FIND;

เป็นโปรแกรมที่ใช้ในการค้นหาข้อมูล

PROCEDURE STORK\_IN\_OUT(COMM, CODE, MOUN : STRING);

เป็นโปรแกรมที่ใช้ในการตรวจสอบการเข้าออกของสินค้าและเพิ่มหรือลดจำนวนของสินค้า

PROCEDURE LIST;

เป็นโปรแกรมที่ใช้ในการเรียก RECORD ต่าง ๆ ออกมาแสดงบนจอภาพ

PROCEDURE SENDTX;

เป็นโปรแกรมที่ใช้ติดต่อกับ WARE HOUSE โดยการระบุรหัสต่าง ๆ

โปรแกรมย่อยที่ใช้ในโปรแกรม PULLDR.PAS

PROCEDURE INVERSE;

เป็นโปรแกรมย่อยในการทำให้สีของพื้น และสีของตัวอักษรสลับกัน

PROCEDURE NORMAL;

เป็นโปรแกรมย่อยที่ทำให้สีพื้นและสีของตัวอักษรกลับเป็นเหมือนเดิม มักจะใช้ร่วมกับ โปรแกรมย่อย INVERSE และมักจะใช้เกี่ยวกับแถบแสง

PROCEDURE QUIT;

เป็นโปรแกรมย่อยที่ใช้ในการออกจากระบบฐานข้อมูล

PROCEDURE OPEN\_FILE;

เป็นโปรแกรมย่อยที่เรียกใช้โปรแกรมย่อย OPEN\_F

PROCEDURE SAVE\_FILE;

เป็นโปรแกรมย่อยที่เรียกโปรแกรมย่อย WRITEF(I) เพื่อที่จะเก็บข้อมูลที่แก้ไขแล้วลงสู่แผ่นดิสก์

PROCEDURE LIST1;

เป็นโปรแกรมย่อยที่ใช้เรียกโปรแกรมย่อย LIST เพื่อใช้แสดงข้อมูลในฐานข้อมูล

PROCEDURE PULLDOWN\_MENU(X1,Y1,Z : INTEGER;C2:D);

เป็นโปรแกรมย่อยที่ใช้ในการสร้าง PULLDOWN MENU ย่อยของโปรแกรม ซึ่งจะสะดวกมากในการเรียกใช้ ถ้าต้องการใช้ MENU ตรงส่วนใดของจอก็สามารถระบุและทำขึ้นมาได้โดยทันที เมื่อเรียกใช้โปรแกรมนี้อย่างที่

X1 = COLUME ที่มุมบนของกรอบที่จะสร้าง

Y1 = ROW ที่บรรทัดแรกของกรอบที่จะสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Z = จำนวนตัวเลือกของคำสั่ง

C2 = ชื่อคำสั่งต่าง ๆ ซึ่งจองเป็น ARRAY เก็บเอาไว้

PROCEDURE EDIT2;

เป็นโปรแกรมย่อยที่ใช้เรียกโปรแกรมย่อย EDIT1

PROCEDURE CREATE2;

เป็นโปรแกรมย่อยที่ใช้ในการสร้างคำสั่งย่อยขึ้นมาอีกซึ่งจะสร้างในลักษณะ PULLDOWN MENU ซึ่งจะเรียกใช้โปรแกรมย่อยของ PULLDOWN\_MENU โปรแกรมย่อยของ CREATE2 จะมีดังนี้

1. DATA BASE จะเป็นโปรแกรม CREATE1 จะถูกเรียกเมื่อเราเลือกโปรแกรม CRATE2 เพื่อที่จะสร้างฐานข้อมูลใหม่และมี WARE HOUSE เพิ่มขึ้นอีก
2. APPEND จะเป็นโปรแกรมที่ใช้ในการเพิ่มข้อมูลเมื่อมีสินค้าตัวใหม่
3. QUIT เป็นการเลิกออกจากโปรแกรมนี้

PROCEDURE EDIT2;

เป็นโปรแกรมย่อยที่ใช้ในการเรียกใช้โปรแกรมย่อย EDIT1

PROCEDURE LIST2;

เป็นโปรแกรมย่อยที่ใช้ในการคำสั่งและเลือกใช้ได้ด้วย PULLDOWN MENU มีคำสั่งดังนี้

1. LIST เรียกใช้ โปรแกรมย่อย LIST1
2. FIND เรียกใช้ โปรแกรมย่อย FIND1
3. REPORT เรียกใช้ โปรแกรมย่อย PRINT
4. QUIT เป็นโปรแกรมยกเลิกการทำงาน

PROCEDURE SUB1\_MENU;

เป็นโปรแกรมย่อยที่ใช้ในการสร้างคำสั่งย่อย โดยสามารถเรียกใช้

โดยผ่านทาง PULLDOWN MENU ซึ่งจะมีคำสั่งย่อยดังนี้  
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1.OPEN จะทำการเรียกใช้โปรแกรมย่อย OPEN\_FILE
- 2.SAVE จะทำการเรียกใช้โปรแกรมย่อย SAVE\_FILE
- 3.CLOSE จะทำการเรียกใช้โปรแกรมย่อย CLOSEF
- 4.QUIT จะทำการเรียกใช้โปรแกรมย่อย QUIT

MAINPROGRAM TES

เป็นโปรแกรมหลักที่เรียกใช้โปรแกรมย่อยเหล่านี้

- 1.FILE จะทำการเรียกใช้โปรแกรมย่อย SUB1\_MENU
- 2.UPDATE จะทำการเรียกใช้โปรแกรมย่อย CREATE2
- 3.EDIT จะทำการเรียกใช้โปรแกรมย่อย EDIT1
- 4.LIST จะทำการเรียกใช้โปรแกรมย่อย LIST2
- 5.SEND จะทำการเรียกใช้โปรแกรมย่อย SENDTX1

ซึ่งคำสั่งต่าง ๆ จะถูกเลือกใช้ได้โดยการเลื่อนแถบแสง

โปรแกรมย่อยที่ใช้ในโปรแกรม ASYNC.PAS

```
PROCEDURE ASYNCINIT(PORTNO:BYTE;BAUDRATE:WORD;PARITY:CHAR  
;STOPBIT:BYTE;DATASIZE:BYTE);
```

เป็นโปรแกรมย่อยในการ SET อัตราการส่งข้อมูล ในโปรแกรมนี้จะ  
ใช้ PORT NUMBER, COM1, 9600, E, 1, 8

```
PROCEDURE SEND(DATA : BYTE);
```

เป็นโปรแกรมย่อยที่ใช้ในการส่งข้อมูลไปยังสายสื่อสารจะใช้ส่งครั้งละ  
8 BIT หรือ 1 BYTE จะส่งไปในลักษณะรหัส ASCII ซึ่งถูกแปลงแล้วข้อมูลจะ  
เป็นตัวเลข

```
PROCEDURE SENDTXT(TXT : STRI);
```

เป็นโปรแกรมย่อยที่ใช้ในการส่งข้อมูลครั้งละหลาย ๆ ตัวอักษรซึ่งจะ  
เรียกใช้ SEND อีกครั้งในการส่ง

```
PROCEDURE COMM(COMMAN : STRI ; CODE : STRI ; MOUN : STRI);
```

เป็นโปรแกรมย่อยที่ทำการสื่อสารกับ WAREHOUSE  
COMMAND จะเป็นรหัสในการที่จะบอกว่าส่งหรือรับข้อมูล(IP=รับ,XP=ส่ง)  
CODE จะเป็นรหัสสินค้า  
MOUN จะเป็นจำนวนสินค้า  
โดยการส่งรหัสต้องใช้เครื่องหมาย \* นำหน้าและ # ตามหลัง  
รหัสที่จะส่ง

```
FUNCTION ASYCREC : BYTE;
```

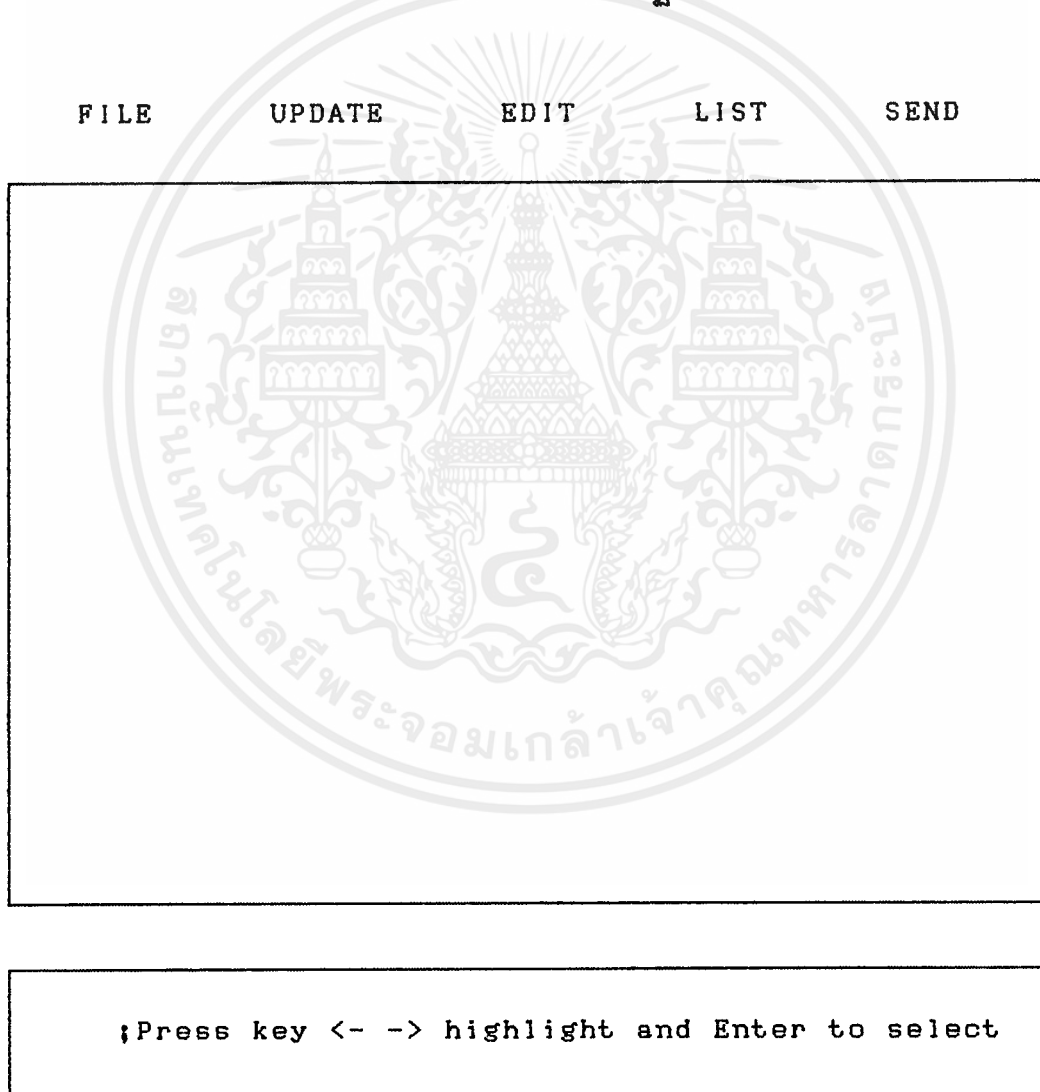
เป็นโปรแกรมย่อยที่ใช้ในการรับข้อมูลจากสายสื่อสารของ WAREHOUSE  
โดยจะรับมาครั้งละ 1 BYTE

บทที่ 4

การใช้งานโปรแกรม การจัดเก็บสินค้าด้วยคอมพิวเตอร์

ลักษณะของโปรแกรม การจัดเก็บสินค้าด้วยคอมพิวเตอร์ นั้นจะเป็นลักษณะของ PULLDOWN MENU ทั้งหมด เพราะเราจะเน้นความสะดวกในการใช้งาน โดยจะขออธิบายการใช้งานดังนี้

ไฟล์ที่ถูกทำเป็น ระบบการจัดการฐานข้อมูลชื่อ WARE\_HOUSE.EXE เมื่อทำการโหลดโปรแกรมมาใช้งานแล้วจะปรากฏที่หน้าจอดังนี้



รูปที่ 1 เมนูหลัก

เมนูหลัก จะประกอบด้วย เมนูย่อย 5 เมนูได้แก่ FILE, UPDATE, EDIT, LIST, SEND โดยแต่ละเมนูจะทำงานดังนี้  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. FILE ให้เลื่อนแถบแสงไปยัง FILE แล้วกด Enter จะปรากฏรายการดังรูปที่ 2

FILE UPDATE EDIT LIST SEND



;Press key <- -> highlight and Enter to select

รูปที่ 2 เมื่อเลือกเมนู FILE

จากรูปที่ 2 จะปรากฏรายการ 4 รายการคือ OPEN, SAVE, CLOSE, EXIT โดยแต่ละรายการจะถูกใช้งานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1.OPEN เมื่อเลื่อนแถบแสงไปยัง OPEN แล้วกด Enter จะปรากฏ หน้าจอตั้งรูปที่ 3

FILE UPDATE EDIT LIST SEND

<u>OPEN</u> SAVE CLOSE EXIT	ENTER OPEN FILE NAME : _____
--------------------------------------	------------------------------

;Press key <- -> highlight and Enter to select

รูปที่ 3 เมื่อเลือกรายการ OPEN

ให้ใส่ชื่อไฟล์ที่ต้องการเปิดเพิ่มข้อมูล โดยไฟล์นั้นต้องเป็นไฟล์ที่ ถูกสร้างไว้แล้วโดยโปรแกรมนี้และ ถ้าต้องการออกจากรายการ OPEN นี้ให้ กด ESC

1.2.SAVE เมื่อเลื่อนแถบแสงไปที่รายการ SAVE แล้วกด Enter เพื่อเป็นการเก็บข้อมูล เมื่อมีการแก้ไขข้อมูลหรือเลิกใช้โปรแกรม

1.3.CLOSE เมื่อเลื่อนแถบแสงไปที่รายการ CLOSE แล้วกด Enter เพื่อทำการปิดเพิ่มข้อมูล เมื่อต้องการเลิกใช้ไฟล์นั้น

1.4.EXIT เมื่อเลื่อนแถบแสงไปที่รายการ EXIT แล้วกด Enter เมื่อ ต้องการออกจากโปรแกรม

2.UPDATE ให้เลื่อนแถบแสงไปที่เมนู UPDATE แล้วกด Enter จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งาน เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ปรากฏรายการขึ้นที่หน้าจอตั้งรูปที่ 4  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FILE      UPDATE      EDIT      LIST      SEND

<table border="1" style="margin: auto;"> <tr><td style="text-align: center;">DATABASE</td></tr> <tr><td style="text-align: center;">APPEND</td></tr> <tr><td style="text-align: center;">QUIT</td></tr> </table>	DATABASE	APPEND	QUIT
DATABASE			
APPEND			
QUIT			

;Press key <- -> highlight and Enter to select

รูปที่ 4 เมื่อเลือกเมนู UPDATE

ซึ่งจะมีรายการดังนี้ DATABASE, APPEND, QUIT โดยแต่ละรายการจะถูกนำมาใช้งานดังนี้

2.1. DATABASE ให้เลื่อนแถบแสงไปที่รายการ DATABASE แล้วกด Enter จะปรากฏกรอบข้อความดังรูปที่ 5

FILE      UPDATE      EDIT      LIST      SEND

<table border="1" style="margin: auto;"> <tr><td style="text-align: center;"><u>DATABASE</u></td></tr> <tr><td style="text-align: center;">APPEND</td></tr> <tr><td style="text-align: center;">QUIT</td></tr> </table>	<u>DATABASE</u>	APPEND	QUIT	<table border="1" style="margin: auto;"> <tr><td style="text-align: center;">CREATE FILE NAME = _____</td></tr> </table>	CREATE FILE NAME = _____
<u>DATABASE</u>					
APPEND					
QUIT					
CREATE FILE NAME = _____					

;Press key <- -> highlight and Enter to select

รายการ DATABASE นี้เป็นรายการที่ใช้สร้างไฟล์ข้อมูลขึ้นใหม่ โดยใส่ชื่อไฟล์ที่ต้องการสร้างแล้วกด Enter จะปรากฏจอภาพดังรูปที่ 6

FILE            UPDATE            EDIT            LIST            SEND

CREATE STOCK DATABASE

FILE NAME = \_\_\_\_\_

RECORD = 0

PART_NUMBER : _____	ITEM_CLASS : _____
UNIT_PRICE : _____	UNIT_ON_HAND : _____

TOTAL\_UNIT\_PRICE : \_\_\_\_\_

;Press key move highlight and Enter to select

รูปที่ 6 เมื่อป้อนชื่อไฟล์ข้อมูลและกด Enter ซึ่งแต่ละช่องรายการจะรับค่าข้อมูลต่างกันดังนี้

- PART\_NUMBER : \_\_\_\_\_ จะรับค่ารหัสสินค้า
- ITEM\_CLASS : \_\_\_\_\_ จะรับชื่อสินค้า
- UNIT\_PRICE : \_\_\_\_\_ จะรับค่าราคาสินค้า (ต่อชิ้น)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนช่องรายการดังนี้จะเป็นส่วนที่แสดงค่าต่าง ๆ ซึ่งจะขอกว่าในหัวข้อต่อ ๆ ไป

UNIT\_ON\_HAND : \_\_\_\_\_ จะเป็นจำนวนของสินค้าชนิดนั้น

TOTAL\_UNIT\_PRICE : \_\_\_\_\_ จะเป็นค่าผลรวมของสินค้าทั้งหมด

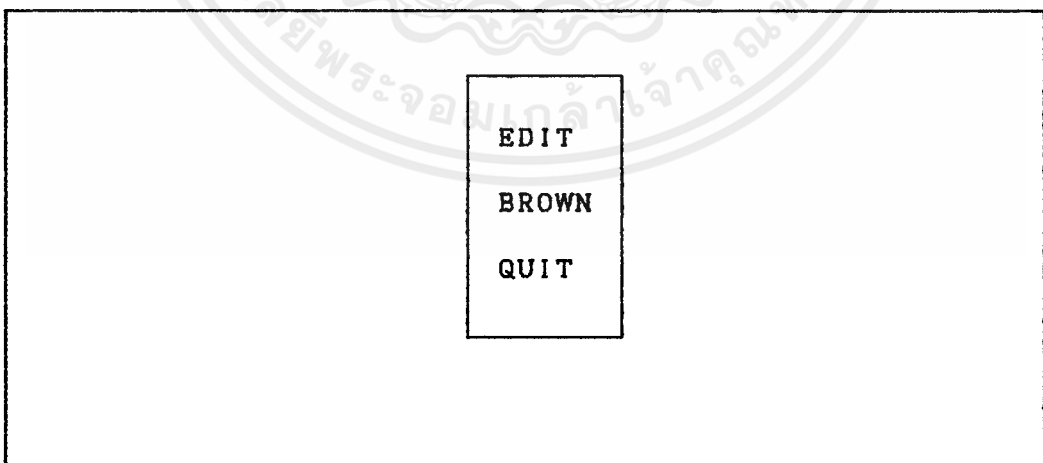
เมื่อเสร็จจากการป้อนข้อมูลหรือต้องการจะออกไปสู่เมนู UPDATE ให้กด Esc

2.2. APPEND จากรูปที่ 4 เมื่อเลือกรายการ APPEND แล้วกด Enter จะปรากฏช่องข้อมูลดังรูปที่ 6 แต่จะเป็นช่องข้อมูลใน RECORD สุดท้ายซึ่งไม่มีข้อมูลอยู่ เพื่อเป็นการเพิ่มเติมข้อมูลในกรณีที่ต้องการเพิ่มชนิดของสินค้าในไฟล์ที่ถูกสร้างขึ้นแล้วและยังเปิดอยู่ และเมื่อเสร็จจากการป้อนข้อมูลหรือต้องการจะออกไปสู่เมนู UPDATE ให้กด Esc

2.3 QUIT ใช้เมื่อต้องการออกจากเมนู UPDATE

3.EDIT เมื่อเลื่อนแถบแสงมายังเมนู EDIT แล้วกด Enter จะปรากฏรายการดังรูปที่ 7

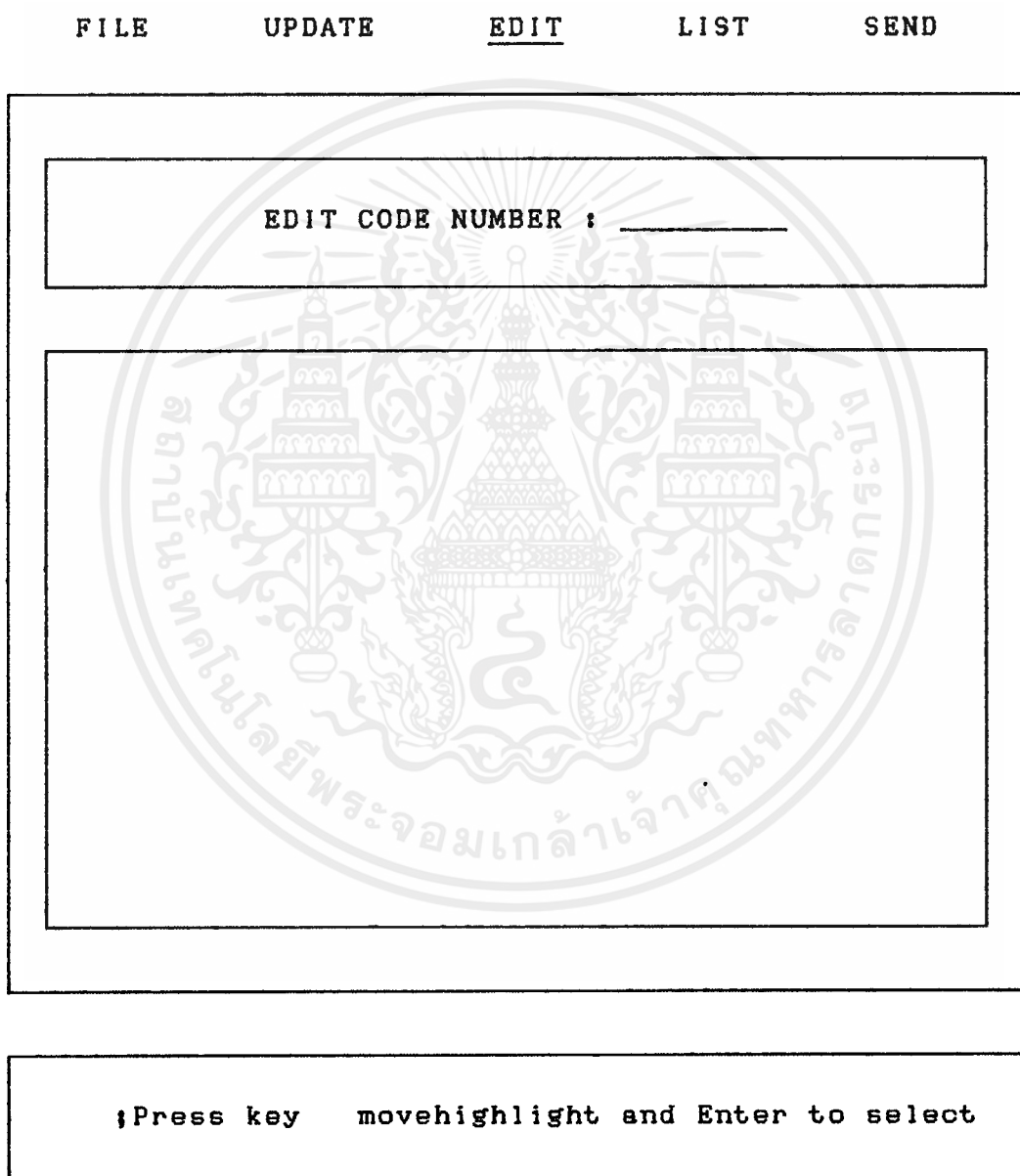
FILE UPDATE EDIT LIST SEND



;Press key movehighlight and Enter to select

ซึ่งจะมีรายการต่าง ๆ ดังนี้ EDIT, BROWN, QUIT ซึ่งจะมีลักษณะการทำงานดังนี้

3.1. EDIT ให้เลื่อนแถบแสงไปที่รายการ EDIT แล้วกด Enter จะปรากฏจอภาพดังรูปที่ 8



รูปที่ 8 เมื่อเลื่อนแถบสว่างมาที่รายการ EDIT แล้วกด Enter

ในรายการ EDIT นี้เป็นการแก้ไขข้อมูลใน RECORD ที่ต้องการโดยใส่รหัสของสินค้าที่ต้องการแก้ไขแล้วกด Enter จะปรากฏดังรูปที่ 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FILE UPDATE EDIT LIST SEND

EDIT CODE NUMBER : _____	
PART_NUMBER	: _____
ITEM_CLASS	: _____
UNIT_PRICE	: _____
UNIT_ON_HAND	: _____
TOTAL_UNIT_PRICE	: _____

;Press key movehighlight and Enter to select

รูปที่ 9 จอภาพจะแสดงรายการใน RECORD ที่ต้องการแก้ไข  
เมื่อเสร็จจากการแก้ไขข้อมูลหรือต้องการจะออกไปสู่รายการ EDIT  
ให้กด Esc

3.2.BROWN จากรูปที่ 7 เมื่อเลื่อนแถบแสงไปที่รายการ BROWN  
แล้วกด Enter จะเป็นการแก้ไขข้อมูลทุก RECORD ดังรูปที่ 10

FILE UPDATE EDIT LIST SEND

#REC_ON	CODE	NAME	PRICE	MOUNT	TOTAL
1.	_____	_____	_____	_____	_____
2.	_____	_____	_____	_____	_____
3.	_____	_____	_____	_____	_____
4.	_____	_____	_____	_____	_____
5.	_____	_____	_____	_____	_____

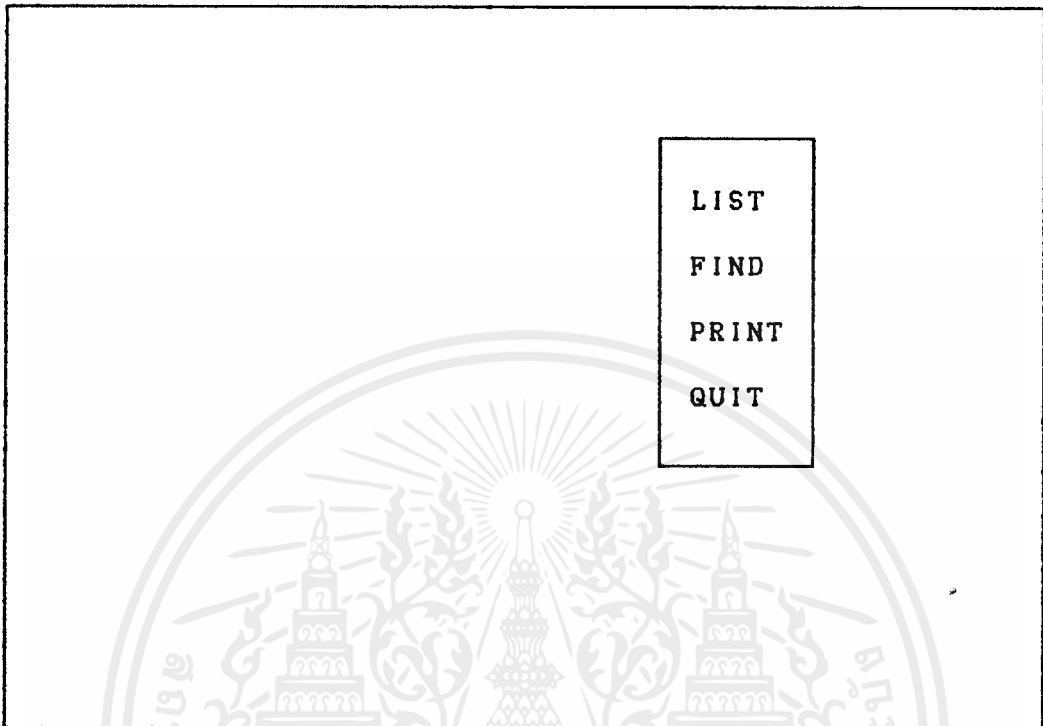
;Press key movehighlight and Enter to select

รูปที่ 11 แสดงการแก้ไขข้อมูลทุก ๆ RECORD ในไฟล์ข้อมูล  
 สามารถที่จะเลือก RECORD ที่จะแก้ไขได้โดยใช้แถบแสงเลื่อนไปยัง  
 RECORD ที่ต้องการ

3.3.QUIT เมื่อต้องการออกจากรายการ BROWN สู่เมนู EDIT

4. LIST ให้เลื่อนแถบแสงไปที่เมนู LIST แล้วกด Enter จะ  
 ปรากฏในจอภาพดังรูปที่ 12

FILE UPDATE EDIT LIST SEND



; Press key <- -> highlight and Enter to select

รูปที่ 12 เมื่อเลื่อนแถบสว่างมายังเมนู LIST แล้วกด Enter

เมื่อเลื่อนแถบสว่างมายังเมนู LIST แล้วกด Enter จะปรากฏรายการต่าง ๆ ดังนี้ LIST, FIND, PRINT, QUIT ซึ่งมีลักษณะการใช้งานดังนี้

4.1.LIST เมื่อเลื่อนแถบแสงไปที่รายการ LIST แล้วกด Enter จะปรากฏภาพบนจอตั้งรูปที่ 13 ซึ่งรายการ LIST นี้จะใช้ในการขอข้อมูลทุก RECORD ในไฟล์ข้อมูลซึ่งถูกเปิดอยู่

FILE UPDATE EDIT LIST SEND

#REC_ON	CODE	NAME	PRICE	MOUNT	TOTAL
1.	_____	_____	_____	_____	_____
2.	_____	_____	_____	_____	_____
3.	_____	_____	_____	_____	_____
4.	_____	_____	_____	_____	_____
5.	_____	_____	_____	_____	_____
6.	_____	_____	_____	_____	_____
7.	_____	_____	_____	_____	_____

;Press key movehighlight and Enter to select

รูปที่ 13 แสดงการขอดูข้อมูลทุก ๆ RECORD ในไฟล์ที่เปิดอยู่

4.2.FIND เมื่อเลื่อนแถบแสงไปที่รายการ FIND แล้วกด Enter จะปรากฏภาพบนจอจดังรูปที่ 14

FILE UPDATE EDIT LIST SEND

WHAT DO YOU WANT TO FIND PART\_NUMBER : \_\_\_\_\_

;Press key move highlight and Enter to select

รูปที่ 14 จอภาพแสดงการรหัสข้อมูลที่ต้องการค้นหา  
 เมื่อป้อนรหัสของข้อมูลที่ต้องการค้นหาแล้วกด Enter จอภาพจะ  
 ปรากฏดังรูปที่ 15

FILE UPDATE EDIT LIST SEND

WHAT DO YOU WANT TO FIND PART\_NUMBER : \_\_\_\_\_

ITEM\_CLASS = \_\_\_\_\_ UNIT\_PRICE = \_\_\_\_\_

UNIT\_ON\_HAND= \_\_\_\_\_ TOTAL\_UNIT\_PRICE= \_\_\_\_\_

;Press key move highlight and Enter to select

เมื่อเสร็จจากการค้นหาข้อมูลหรือต้องการจะออกไปสู่เมนู LIST ให้

กด Esc

4.3.PRINT ให้เลื่อนแถบแสงไปที่รายการ PRINT แล้วกด Enter PRINTER จะทำการพิมพ์ข้อมูลของไฟล์ที่เปิดอยู่

4.4.QUIT เมื่อต้องการออกจากเมนู LIST สู่เมนูหลัก

5.SEND ให้เลื่อนแถบแสงไปที่รายการ SEND แล้วกด Enter จอภาพจะปรากฏภาพดังรูปที่ 16

FILE UPDATE EDIT LIST SEND

```
WARE HOUSE
DO YOU WANT IMPORT OR EXPORT (IP/XP)? ___
CODE STROCK ? _____
HOW MANY STROCK MOUNT? _____
```

;Press key movehighlight and Enter to select

รูปที่ 16 จอภาพแสดงการสื่อสารข้อมูลกับ WAREHOUSE

เมื่อต้องการนำสินค้าเข้าให้ใส่รหัส IP และเมื่อต้องการนำสินค้าออกให้ใส่รหัส XP จากนั้นให้ใส่รหัสและจำนวนของสินค้าที่ต้องการนำเข้าหรือออก

โปรแกรมที่ใช้ เขียนโดย TURBO PASCAL

```
PROGRAM MAIN_WAREHOUSE;  
USES CRT,DOS,ASYNC,STORKW,WIN;  
CONST  
    COM1 = $00;  
    COM2 = $01;  
TYPE D = ARRAY[1..5] OF STRING;  
VAR  
    FNAME : STRING[14];  
    G : BOOLEAN;  
    C,C1 : D;  
    B : CHAR;  
    I : INTEGER;  
    I1,I2,I3 : BYTE;  
  
{ |-----| }  
{ |          EXIT TO DOS          | }  
{ |-----| }  
  
PROCEDURE EXIT1;  
  
    BEGIN  
  
        G := TRUE;  
  
        SENDTXT('H');  
  
        CLRSCR;  
  
    END;  
  
PROCEDURE OPEN_FILE;
```

BEGIN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
TABLE(13,4,49,6);  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
OPEN_F;  
CLEARWIN(13,4,49,6);
```

```
END;
```

```
{ | ----- | }  
{ | FROM MAIN PROGRAM CALL TO PROCEDURE LIST | }  
{ | ----- | }
```

```
PROCEDURE LIST1;
```

```
VAR F1 : CHAR;
```

```
BEGIN
```

```
CLEARWIN(47,4,57,9);
```

```
TABLE(2,4,78,6);
```

```
TABLE(2,7,78,21);
```

```
GOTOXY(4,5);
```

```
WRITE('#REC_NO      CODE      NAME  
      PRICE      MOUNT      TOTAL ');
```

```
LIST;
```

```
F1 := READKEY;
```

```
CLEARWIN(2,4,78,21);
```

```
END;
```

```
{ | ----- | }  
{ | FROM MAIN PROGRAM CALL TO PROCEDURE WROWSE | }  
{ |----- | }
```

```
PROCEDURE BROWSE1;
```

```
VAR F1 : CHAR;
```

```
BEGIN
```

```
CLEARWIN(34,4,48,8);
```

```
TABLE(2,4,78,6);
```

```
TABLE(2,7,78,21);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ข้อมูลนี้ทางสื่อออนไลน์ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
GOTOXY(4,5);WRITE('#REC_NO      CODE      NAME
                        PRICE      MOUNT      TOTAL ');
BROWSE;
F1 := READKEY;
CLEARWIN(2,4,78,21);
END;
```

```
{-----}
{ CREATE PULLDOWN MENU }
{ X1 = COLUMN ,Y1 = ROW }
{ Z = MOUNT OF CHOICE }
{ C2 = NAME OF CHOICE }
{-----}
```

```
PROCEDURE PULLDOWN_MENU(X1,Y1,Z : INTEGER;C2:D);
VAR Y2,I4 : INTEGER; G1 : CHAR;
BEGIN
  Y2 := Y1;
  NORMAL;
  FOR I := 1 TO Z DO
  BEGIN
    GOTOXY(X1,Y2);
    WRITE(C2[I]);
    Y2 := Y2+1;
  END;
  Y2 := Y2-1;
  I3 := 1;
  I4 := Y1;
  INVERSE;
  GOTOXY(X1,I4);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
WRITE(C2[I3]);  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
NORMAL;
GOTOXY(17,24);
WRITE('Press key ',CHR(24),CHR(25),' highlight and Enter
to select and ESC to menu . ');
G := FALSE;
WHILE G = FALSE DO
BEGIN
G1 := READKEY;
IF ORD(G1) = 0 THEN
G1:= READKEY;
IF G1 = 'P' THEN
BEGIN
NORMAL;
GOTOXY(X1,14);
WRITE(C2[I3]);
I4 := I4+1;
I3 := I3+1;
IF I3 > Z THEN BEGIN I3 := 1;
I4 := Y1;
END;
INVERSE;
GOTOXY(X1,14);
WRITE(C2[I3]);
NORMAL;
END;
IF G1 = 'H' THEN
BEGIN
NORMAL ;
GOTOXY(X1,14);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

WRITE(C2[I3]);
I4 := WHEREY-1;
I3 := I3-1;
IF I3 < 1 THEN
BEGIN I3 := Z; I4 := Y2; END;
INVERSE ; GOTOXY(X1,I4);
WRITE(C2[I3]);

NORMAL
END;

IF ORD(G1) = 13 THEN
EXIT;
IF ORD(G1) = 27 THEN
BEGIN
CLEARWIN(X1-3,Y1-1,X1+10,Y1+Z+1);
I3 := 0;
EXIT;
END;
END;
END;

```

```

PROCEDURE SENDTX1;
VAR F1 : CHAR;
BEGIN
TABLE(29,5,73,14);
SENDTX;
F1 := READKEY;
END;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PROCEDURE EDIT2;  
    BEGIN  
        EDIT;  
        NORMAL;  
        CLEARWIN(10,5,70,20);  
    END;
```

```
PROCEDURE FIND1;  
    BEGIN  
        FIND;  
        CLEARWIN(10,5,70,20);  
    END;
```

```
PROCEDURE APPEND1;  
    VAR F1 : CHAR;  
    BEGIN  
        APPEND;  
    END;
```

```
PROCEDURE CREATE2;  
    VAR X1,Y1,Y2,Z : INTEGER;  
        C1 : D;  
    BEGIN  
        TABLE(17,4,28,8);  
        C1[1] := ' DATABASE '  
        C1[2] := ' APPEND '  
        C1[3] := ' QUIT '
```

```
X1:=18;Y1:=5;Z := 3;  
PULLDOWN_MENU(X1,Y1,Z,C1);  
CASE 13 OF  
    1 : CREATE1;  
    2 : APPEND1;  
END;  
CLEARWIN(17,4,28,8);  
END;
```

```
PROCEDURE EDIT1;  
VAR X1,Y1,Y2,Z : INTEGER;  
    C1 : D;  
BEGIN  
    TABLE(34,4,43,8);  
    C1[1] := ' EDIT  ' ;  
    C1[2] := ' BROWSE ' ;  
    C1[3] := ' QUIT  ' ;  
    X1:=35;Y1:=5;Z := 3;  
    PULLDOWN_MENU(X1,Y1,Z,C1);  
CASE 13 OF  
    1 : EDIT2;  
    2 : BROWSE1;  
END;  
CLEARWIN(34,4,43,8);  
END;
```

```
PROCEDURE LIST2;  
  VAR X1,Y1,Y2,Z : INTEGER;  
      C1 : D;  
  BEGIN  
    TABLE(47,4,57,9);  
    C1[1] := ' LIST  ' ;  
    C1[2] := ' FIND  ' ;  
    C1[3] := ' PRINT ' ;  
    C1[4] := ' QUIT  ' ;  
    X1:=49;Y1:=5;Z := 4;  
    PULLDOWN_MENU(X1,Y1,Z,C1);  
    CASE 13 OF  
      1 : LIST1;  
      2 : FIND1;  
      3 : PRINT;  
    END;  
    CLEARWIN(47,4,57,9);  
  END;
```

```
PROCEDURE SUB1_MENU;  
  VAR C1 : D;  
      G1 : CHAR;  
      X1,Y1,Z : INTEGER;  
  BEGIN  
    TABLE(4,4,12,9);  
    C1[1] := ' OPEN  ' ;  
    C1[2] := ' SAVE  ' ;  
    C1[3] := ' CLOSE ' ;  
    C1[4] := ' EXIT  ' ;
```

```
X1:=5;Y1 := 5;Z:=4;
PULLDOWN_MENU(X1,Y1,Z,C1);
CASE 13 OF
    1 : OPEN_FILE ;
    2 : SAVE_FILE ;
    3 : CLOSE_F;
    4 : EXIT1;
END;
CLEARWIN(4,4,12,9);
END;
VAR Y1,M,DATE,DAY :WORD;
BEGIN
    CLRSCR;
    ASYNCINIT(COM1,4800,'N',1,8);
    GETDATE(Y1,M,DATE,DAY);
    GOTOXY(70,1);
    WRITE(DATE,'/',M,'/',Y1);
    TABLE(1,3,79,22);
    TABLE(1,23,79,25);
    C[1] := ' FILE ' ;
    C[2] := 'UPDATE';
    C[3] := ' EDIT ' ;
    C[4] := ' LIST ' ;
    C[5] := ' SEND ' ;
    NORMAL;
    GOTOXY(7,2);
    WRITELN(C[1]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
GOTOXY(21,2);
WRITELN(C[2]);
GOTOXY(35,2);
WRITELN(C[3]);
GOTOXY(49,2);
WRITELN(C[4]);
GOTOXY(63,2);
WRITELN(C[5]);
INVERSE;
I1 := 1;
I2 := 7;
GOTOXY(I2,2);
WRITE(C[I1]);
NORMAL;
GOTOXY(15,24);
WRITE('! Press key ',CHR(27),CHR(26),'
      highlight and Enter to select . ');
G := FALSE;
WHILE G = FALSE DO
  BEGIN
    B := READKEY;
    IF ORD(B) = 0 THEN
      B:=READKEY;
    IF B = 'M' THEN
      BEGIN
        NORMAL;
        GOTOXY(I2,2);
        WRITE(C[I1]);
        I2:= WHEREX+8;
```

-48-

```
I1 := I1+1;
```

```
IF I1 > 5 THEN
```

```
BEGIN
```

```
I1 := 1;
```

```
I2 := 7;
```

```
END;
```

```
INVERSE;
```

```
GOTOXY(I2,2);
```

```
WRITE(CI I1);
```

```
NORMAL;
```

```
END;
```

```
IF B = 'K' THEN
```

```
BEGIN
```

```
NORMAL ;
```

```
GOTOXY(I2,2);
```

```
WRITE(CI I1);
```

```
I2 := WHEREX-20;
```

```
I1 := I1 - 1;
```

```
IF I1 < 1 THEN
```

```
BEGIN
```

```
I1 := 5;
```

```
I2 := 63;
```

```
END;
```

```
INVERSE;
```

```
GOTOXY(I2,2);
```

```
WRITE(CI I1);
```

```
NORMAL;
```

```
END;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
IF ORD(B) = 13 THEN
```

```
  CASE 11 OF
```

```
    1 : SUB1_MENU;
```

```
    2 : CREATE2;{CREATE1;}
```

```
    3 : EDIT1;{EDIT;}
```

```
    4 : LIST2;{LIST1;}
```

```
    5 : SENDTX1;
```

```
  END;
```

```
GOTOXY(15,24);
```

```
WRITE('! Press key ',CHR(27),CHR(26),'  
highlight and Enter to select  
' );
```

```
END;
```

```
END.
```

```
UNIT ASYNC;
INTERFACE
USES DOS,CRT;
    TYPE
        STRI = STRING;
PROCEDURE ASYNCINIT(PORTNO:BYTE;BOUDRATE:WORD;
    PARITY:CHAR;STOPBIT:BYTE;DATASIZE:BYTE);
PROCEDURE SEND(DATA : BYTE);
PROCEDURE SENDTXT(TXT : STRI);
FUNCTION ASYCREC : BYTE;
FUNCTION RXCHAR : STRI;
IMPLEMENTATION
CONST
    COM1 = $00;
    COM2 = $01;
VAR
    COMMAN, CODE, MOUN : STRI;
    COMPORT : BYTE;
    DATA, DA : BYTE;
    TXT : STRI;
    CH : CHAR;
```

```
{-----}
{   SET   PORT,BOUDRATE,PARITY,STOPBIT,DATA   }
{-----}
```

```
PROCEDURE ASYNCINIT(PORTNO:BYTE;BOUDRATE:WORD;
                    PARITY:CHAR;STOPBIT:BYTE;DATASIZE:BYTE);
```

```
VAR
```

```
    REGS : REGISTERS ;
```

```
BEGIN
```

```
    REGS.AL := $00;
```

```
    CASE BOUDRATE OF
```

```
        110 : REGS.AL := REGS.AL OR $00;
```

```
        150 : REGS.AL := REGS.AL OR $20;
```

```
        300 : REGS.AL := REGS.AL OR $40;
```

```
        600 : REGS.AL := REGS.AL OR $60;
```

```
        1200 : REGS.AL := REGS.AL OR $80;
```

```
        2400 : REGS.AL := REGS.AL OR $0A0;
```

```
        4800 : REGS.AL := REGS.AL OR $0C0;
```

```
        9600 : REGS.AL := REGS.AL OR $0E0;
```

```
END;
```

```
    CASE UPCASE(PARITY) OF
```

```
        'N' : REGS.AL := REGS.AL OR $00;
```

```
        'O' : REGS.AL := REGS.AL OR $08;
```

```
        'E' : REGS.AL := REGS.AL OR $18;
```

```
END;
```

```
    CASE STOPBIT OF
```

```
        1 : REGS.AL := REGS.AL OR $00;
```

```
        2 : REGS.AL := REGS.AL OR $04;
```

```
END;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CASE DATASIZE OF

7 : REGS.AL := REGS.AL OR \$02;

8 : REGS.AL := REGS.AL OR \$03;

END;

CASE PORTNO OF

COM1 : REGS.DX := \$00;

COM2 : REGS.DX := \$01;

END;

COMPORT := REGS.DX ;

REGS.AH := \$00;

INTR(\$14,REGS);

END;

PROCEDURE SEND(DATA : BYTE);

VAR

REGS : REGISTERS;

BEGIN

REGS.AH := \$01;

REGS.AL := DATA;

REGS.DX := \$00;

INTR(\$14,REGS);

END;

```
{-----}  
{ SEND STRING }  
{-----}
```

```
PROCEDURE SENDTXT(TXT : STR1);
```

```
VAR I : INTEGER;
```

```
G : CHAR;
```

```
BEGIN
```

```
GOTOXY(16,24);WRITE(' Sending '  
,TXT,') ;
```

```
FOR I := 1 TO LENGTH(TXT) DO
```

```
BEGIN
```

```
DELAY(50);
```

```
SEND(ORD(TXT[I]));
```

```
END;
```

```
END;
```

```
{-----}  
{ RECEIVE FROM SERI }  
{-----}
```

```
FUNCTION ASYCREC : BYTE;
```

```
VAR
```

```
REGS : REGISTERS;
```

```
BEGIN
```

```
REGS.AH := #02;
```

```
REGS.DX := COM1;
```

```
INTR(#14,REGS);
```

```
ASYCREC := REGS.AL;
```

```
END;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{-----}  
{ RECEIVE STRING }  
{-----}
```

```
FUNCTION RXCHAR : STRI;
```

```
VAR
```

```
DATA : BYTE;
```

```
CH : CHAR;
```

```
TXT : STRI;
```

```
BEGIN
```

```
CH := ' ';
```

```
WHILE CH <> '#' DO
```

```
BEGIN
```

```
DATA := ASYCREC;
```

```
CH := CHR(DATA);
```

```
IF CH = '*' THEN
```

```
BEGIN
```

```
TXT := CH;
```

```
REPEAT
```

```
DATA := ASYCREC;
```

```
CH := CHR(DATA);
```

```
TXT := TXT+CH;
```

```
UNTIL CH = '#';
```

```
END;
```

```
END;
```

```
RXCHAR := TXT;
```

```
END;
```

```
BEGIN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
END.  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
UN, STORKW;  
INTERFACE  
USES TDS, CRT, ASYNC, PRINTER;  
PROCEDURE INVERSE;  
PROCEDURE NORMAL;  
FUNCTION LOOP: STRING;  
FUNCTION LOOP2: STRING;  
PROCEDURE LOOP1;  
PROCEDURE FOPEN;  
PROCEDURE INP_DAT(VAR I : INTEGER);  
PROCEDURE WRITEF(VAR I : INTEGER);  
PROCEDURE CLOSE_F;  
PROCEDURE CREATE1;  
PROCEDURE CREATE(VAR I : INTEGER);  
PROCEDURE CHECK_LENGTH_FILE;  
PROCEDURE APPEND;  
PROCEDURE EDIT;  
PROCEDURE BROWSE;  
PROCEDURE FIND;  
PROCEDURE OPEN F;  
PROCEDURE SAVE_FILE;  
PROCEDURE STORK IN_OUT(COMM, CODE, MOUN : STRING);  
PROCEDURE LIST;  
FUNCTION CHECK_FILE_OPEN: BOOLEAN;  
PROCEDURE SENDTX;  
FUNCTION CHANG_C(VAR CODE : STRING): STRING;  
FUNCTION CHECK_WH(VAR CODE : STRING): STRING;  
FUNCTION SIZE_FILE: INTEGER;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PROCEDURE CLEARWIN(D1,D2,D3,D4: INTEGER);  
PROCEDURE TABLE(C1,R1,C2,R2: INTEGER);  
PROCEDURE TABLE1(C1,R1,C2,R2: INTEGER);  
PROCEDURE LOAD_H;  
PROCEDURE PRINT;
```

#### IMPLEMENTATION

```
TYPE  
  TPRECD = RECORD  
    B,C,N_W : STRING[10];  
    D,E,G : REAL;  
  END;  
  SORT_ARY = TPRECD;  
  SORT_FILE = FILE OF TPRECD ;  
VAR  
  N_W H, CODE_D : ARRAY[1..21] OF STRING;  
  FNAME : STRING[14];  
  F : SORT_FILE;  
  SA : ARRAY[1..21] OF TPRECD;  
  S : STRING[10];  
  P,M : REAL;  
  K, LEN, I, X, Y, Z, V : INTEGER;  
  K2, K3 : STRING;  
  K1 : REAL;  
  CH : CHAR;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PROCEDURE INVERSE;
```

```
  BEGIN
```

```
    TEXTBACKGROUND(7);
```

```
    TEXTCOLOR(0);
```

```
  END;
```

```
PROCEDURE NORMAL;
```

```
  BEGIN
```

```
    TEXTBACKGROUND(0);
```

```
    TEXTCOLOR(7);
```

```
  END;
```

```
{ |-----| }  
{ | CREATE DOUBLE LINE | }  
{ |-----| }
```

```
PROCEDURE TABLE(C1,R1,C2,R2:INTEGER);
```

```
  VAR I: INTEGER;
```

```
  BEGIN
```

```
    TEXTCOLOR(3);
```

```
    GOTOXY(C1,R1);
```

```
    WRITE(CHR(201));
```

```
    GOTOXY(C2,R1);
```

```
    WRITE(CHR(187));
```

```
    GOTOXY(C1,R2);
```

```
    WRITE(CHR(200));
```

```
    GOTOXY(C2,R2);
```

```
    WRITE(CHR(188));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FOR I := C1+1 TO C2-1 DO
```

```
  BEGIN
```

```
    GOTOXY(I,R1);WRITE(CHR(205));
```

```
    GOTOXY(I,R2);WRITE(CHR(205));
```

```
  END;
```

```
    FOR I := R1+1 TO R2-1 DO
```

```
      BEGIN
```

```
        GOTOXY(C1,I);WRITE(CHR(186));
```

```
        GOTOXY(C2,I);WRITE(CHR(186));
```

```
      END;
```

```
TEXTCOLOR(7);
```

```
END;
```

```
{ |-----| }  
{ |   CREATE   SINGLE   LINE   | }  
{ |-----| }  
  { |-----| }  
  { |-----| }
```

```
PROCEDURE TABLE1(C1,R1,C2,R2 : INTEGER);
```

```
  VAR I : INTEGER;
```

```
  BEGIN
```

```
    TEXTCOLOR(3);
```

```
    GOTOXY(C1,R1);
```

```
    WRITE(CHR(218));
```

```
    GOTOXY(C2,R1);
```

```
    WRITE(CHR(191));
```

```
    GOTOXY(C1,R2);
```

```
    WRITE(CHR(192));
```

```
    GOTOXY(C2,R2);
```

```
    WRITE(CHR(217));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FOR I := C1+1 TO C2-1 DO
BEGIN
GOTOXY(I,R1);WRITE(CHR(196));
GOTOXY(I,R2);WRITE(CHR(196));
END;
FOR I := R1+1 TO R2-1 DO
BEGIN
GOTOXY(C1,I);WRITE(CHR(179));
GOTOXY(C2,I);WRITE(CHR(179));
END;
TEXTCOLOR(7);
END;

{ |-----| }
{ |   D1 = COLUMN 1 , D2 = ROW 1   | }
{ |   D2 = COLUMN 2 , D3   ROW 2   | }
{ |-----| }

PROCEDURE CLEARWIN(D1,D2,D3,D4: INTEGER);
BEGIN
WINDOW(D1,D2,D3,D4);
CLRSCR;
WINDOW(1,1,80,25);
END;
```

```
{ |-----| }  
{ | READ CHARECTER AND CHECK MOUNT | }  
{ | CHECK KEY ESC,ENTER AND BACKSPACE | }  
{ |-----| }
```

FUNCTION LOOP:STRING;

VAR J : INTEGER;

BEGIN

CH := ' ';

S := '';

LOOP := '';

V := 0;

J := 1;

WHILE V <= Z DO

BEGIN

GOTOXY(Y+V,X);

CH := READKEY;

IF ORD(CH) = 13 THEN

BEGIN

LOOP := S;

EXIT;

END;

IF ORD(CH) = 27 THEN

BEGIN V := Z+1;

EXIT;

END;

GOTOXY(Y+V,X);

WRITE(CH);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
IF ORD(CH) <> 0 THEN
IF ORD(CH) = 8 THEN
BEGIN
  IF V<> 0 THEN
    BEGIN
      CH := ' ';
      V := V-1; GOTOXY(Y+V,X); WRITE(CH);
      DELETE(S,LENGTH(S),1); END;
    END
  ELSE
    BEGIN
      S := S+CH;
      V := V+1;
    END
  ELSE
    CH := READKEY;
  END;
  LOOP := S;
END;
```

```
FUNCTION LOOP2:STRING;
```

```
  VAR J : INTEGER;
```

```
  BEGIN
```

```
    CH := ' ';
```

```
    S := '';
```

```
    LOOP2 :='';
```

```
    V : 0;
```

```
  WHILE V <= Z DO
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
BEGIN
    GOTOXY(Y+V,X);
    CH := READKEY;
    IF ORD(CH) = 13 THEN
BEGIN
    LOOP2 := S;
EXIT;
END;
    IF ORD(CH) = 27 THEN
        BEGIN V := Z+1;EXIT;END;
        GOTOXY(Y+V,X);
        WRITE(CH);
    IF ORD(CH) <> 0 THEN
    IF ORD(CH) = 8 THEN
BEGIN
    IF V <> 0 THEN
        BEGIN
            CH := ' ';
            V := V-1;
            GOTOXY(Y+V,X);
            WRITE(CH);
            DELETE(S,LENGTH(S),1); END;
        END
    ELSE
        BEGIN
            S := S+UPCASE(CH);
            V := V+1;
        END
    END
END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ELSE
    CH := READKEY;
END;
    LOOP2 := S;
END;
```

```
PROCEDURE LOOP1;
```

```
    BEGIN
        GOTOXY(Y,X);
        READ(K1);
        GOTOXY(Y,X);
    END;
```

```
PROCEDURE CHECK_LENGTH_FILE;
```

```
    VAR I : INTEGER;
        ST : STRING;
    BEGIN
        ST := '';
        IF POS('.',FNAME) = 0 THEN
            IF LENGTH(FNAME) > 8 THEN
                FNAME := COPY(FNAME, 1,8)
```

```
            ELSE
                FNAME := FNAME+'.WHP'
```

```
            ELSE
```

```
        BEGIN
```

```
            ST := COPY(FNAME, POS('.',FNAME), 4);
```

```
            DELETE(FNAME, POS('.',FNAME), 4);
```

```
            I := LENGTH(FNAME);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

IF I > 8 THEN
    FNAME := COPY(FNAME,1,8)+ST
ELSE
    FNAME := FNAME+ST;
END;
END;

```

```

PROCEDURE OPEN_F;
VAR I : INTEGER;
BEGIN
    GOTOXY(16,24);
    WRITE('          Press write file name at open .
GOTOXY(15,5);
    WRITE('ENTER OPEN FILE NAME : ');
    Y : 38; X:=5; Z := 14;
    FNAME := ''; FNAME := LOOP2;
    IF LENGTH(FNAME) = 0 THEN EXIT;
    CHECK_LENGTH_FILE;

    {-----}
    { READLN; }
    {-----}

    ASSIGN(F,FNAME);
    {$I } RESET(F); {$I+}

    IF IOresult <> 0 THEN
        BEGIN

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
GOTOXY(10,24);  
WRITE('                ');  
GOTOXY(24,24);  
WRITE(FNAME,' FILE NOT ARE OPEN IN DATABASE WAREHOUSE');  
READLN;  
EXIT;
```

END

ELSE

```
FOR I := 1 TO FILESIZE(F) DO  
  READ(F,SAC11);  
  GOTOXY(4,24);  
  WRITE(FNAME);  
  LOAD_H;
```

END;

FUNCTION SIZE\_FILE:INTEGER;

BEGIN

```
  SIZE_FILE := FILESIZE(F);
```

END;

FUNCTION CHECK\_FILE\_OPEN:BOOLEAN;

BEGIN

```
  IF LENGTH(FNAME) = 0 THEN
```

BEGIN

```
  GOTOXY(17,24);
```

```
  WRITE('                FILES ARE NOT OPEN. ;ANY KEY TO MENU.
```

```
  CHECK_FILE_OPEN := FALSE;
```

END;

END;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PROCEDURE FOPEN;
```

```
  BEGIN
```

```
    ASSIGN (F,FNAME);
```

```
    REWRITE(F);
```

```
  END;
```

```
PROCEDURE SAVE_FILE;
```

```
  VAR I : INTEGER;
```

```
  BEGIN
```

```
    IF FNAME <> '' THEN
```

```
      BEGIN
```

```
        SEEK(F,0);
```

```
        FOR I : 1 TO SIZE_FILE DO
```

```
          WRITE(F,I);
```

```
        END
```

```
      ELSE
```

```
        BEGIN
```

```
          GOTOXY(10,24);WRITE(' ');
```

```
          GOTOXY(35,24);
```

```
          WRITE('FILE NOT OPEN');
```

```
        END;
```

```
  END;
```

```
PROCEDURE INP_DAT(VAR I : INTEGER);
```

```
  VAR K2 : STRING;
```

```
  VA1 : STRING;
```

```
  VA : REAL;
```

```
  RES,COUNT : INTEGER;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
CHECK : STRING;  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
BEGIN

    K1:=0;

    M :=0;

    NORMAL;

    GOTOXY(35,9);WRITE('record = ',i);

    GOTOXY(5,13);WRITE('PART_NUMBER      : ');

    INVERSE;WRITE(SACIJ.B);

    NORMAL;

    GOTOXY(43,13);

    WRITE('ITEM_CLASS      : ');

    INVERSE;WRITE(SACIJ.C);

    NORMAL;

    GOTOXY(5,15);

    WRITE('UNITS_PRICE      : ');

    INVERSE;WRITE(SACIJ.D:5:0);

    NORMAL;

    GOTOXY(43,15);

    WRITE('UNIT_ON_HAND      : ');

    INVERSE;WRITE(SACIJ.E:6:0);

    NORMAL;

    GOTOXY(23,20);

    WRITE('TOTAL_UNIT_PRICE : ');

    INVERSE;WRITE(SACIJ.G:8:0);

    Y:=25;X:=13;Z:=2;K3 := SACIJ.B;

    CHECK := 'F';

    WHILE CHECK = 'F' DO

        BEGIN
```

```
            Y:=25;X:=13;Z:=2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            K3 := LOOP;
```

```
IF LENGTH(K3) = 0 THEN BEGIN CHECK := 'T';END;
IF ORD(CH) = 27 THEN BEGIN I := 22;EXIT; END;
COUNT := 1;
RES := 1;
WHILE RES < FILESIZE(F)+1 DO
    BEGIN
        IF K3 = SACRES].B THEN
            BEGIN
                GOTOXY(25,13);WRITE(' ');
                RES := FILESIZE(F)+1;
            END
        ELSE
            BEGIN
                COUNT := COUNT + 1;
            END;
            RES := RES +1;
        END;
        IF COUNT > FILESIZE(F) THEN
            BEGIN
                SACI].B := K3;
                CHECK := 'T';
            END;
    END;

END;

Y:=63;X:=13;Z:=6;K3 := SACI].C;
SACI].C := LOOP;
IF LENGTH(SACI].C) = 0 THEN SACI].C := K3;
```

END;

PROCEDURE WRITEF(VAR I : INTEGER);

BEGIN

WRITE(F,SA[ I ]);

END;

PROCEDURE CLOSE\_F;

VAR

RE : CHAR;

BEGIN

IF FNAME <> '' THEN

BEGIN

CLOSE(F);

FNAME := '';

GOTOXY(4,24);WRITE(' ');

END

ELSE

BEGIN

GOTOXY(10,24);WRITE('

GOTOXY(24,24);

WRITE(FNAME, ' FILE NOT ARE OPEN IN  
DATABASE WAREHOUSE');

READLN;

EXIT;

END;

END;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PROCEDURE CREATE1;
  VAR I : INTEGER;
  BEGIN
    FNAME := '';
    GOTOXY(16,24);
    WRITE('          Press write file name.
          ');
    TABLE(35,7,70,9);
    GOTOXY(36,8);
    WRITE(' CREATE FILE NAME = ');
    Y := 56; X := 8; Z := 14; FNAME := LOOP2;
    IF LENGTH(FNAME) = 0 THEN BEGIN CLEARWIN(17,4,70,11);
    EXIT; END;
    CHECK_LENGTH_FILE;
    FOPEN;
    CLEARWIN(13,4,75,11);
    GOTOXY(4,24);WRITE(FNAME);
    TABLE1(3,4,78,7);TABLE1(13,19,67,21);
    TABLE1(3,11,39,18);TABLE1(41,11,78,18);
    GOTOXY(15,5);
    WRITE('          CREATE STOCK DATABASE WAREHOUSE');
    GOTOXY(28,6);WRITE(' FILE NAME = ',FNAME);
    GOTOXY(16,24);
    WRITE('          Press key ESC to menu .');
    I := 1 ;
    CREATE(I);
    CLEARWIN(3,4,78,21);
    WINDOW(1,1,80,25);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
FNT ;  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PROCEDURE APPEND;  
  VAR I : INTEGER;  
  BEGIN  
    IF CHECK_FILE_OPEN = FALSE THEN BEGIN  
      CLEARWIN(13,4,60,10);EXIT;END;  
    CLEARWIN(13,4,60,10);  
    GOTOXY(4,24);WRITE(FNAME);  
    TABLE1(3,4,78,7);  
    TABLE1(13,19,67,21);  
    TABLE1(3,11,39,18);  
    TABLE1(41,11,78,18);  
    GOTOXY(15,5);  
    WRITE('      APPEND STOCK DATABASE WAREHOUSE');  
    GOTOXY(30,6);  
    WRITE(' FILE NAME = ',FNAME);  
    SEEK(F,FILESIZE(F));  
    I := FILEPOS(F)+1;  
    CREATE(I);  
    CLEARWIN(3,4,78,21);  
    WINDOW(1,1,80,25);  
  END;
```

```
PROCEDURE CREATE(VAR I : INTEGER);
```

```
  VAR LEN : INTEGER;
```

```
  BEGIN
```

```
    LEN:=0;
```

```
    WHILE I <= 21 DO
```

```
      BEGIN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
SAC13.B := ' ' ;  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SACIJ.C := '      ';
```

```
SACIJ.N W := '0';
```

```
SACIJ.D := 0;
```

```
SACIJ.E := 0;
```

```
SACIJ.G := 0;
```

```
INP_DAT(I);
```

```
IF LENGTH(SACIJ.B) = 0 THEN
```

```
  BEGIN
```

```
    NORMAL;
```

```
    WINDOW(3,4,77,19);
```

```
    CLRSCR;
```

```
    WINDOW(1,1,80,25);
```

```
    EXIT;
```

```
  END
```

```
ELSE
```

```
  WRITEF(I);
```

```
  I := I+1;
```

```
END;
```

```
  NORMAL;
```

```
  WINDOW(3,4,77,19);
```

```
  CLRSCR;
```

```
  WINDOW(1,1,80,25);
```

```
END;
```

```
PROCEDURE EDIT;
```

```
  VAR G1 : CHAR;
```

```
  CODE : STRING[3];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
VA : REAL;
CHECK : STRING;
COUNT : INTEGER;
RES : INTEGER;
BEGIN
    CLEARWIN(34,4,43,8);
    IF CHECK_FILE_OPEN = FALSE THEN
BEGIN
    G1 := READKEY;
    EXIT;
    END;
    GOTOXY(16,24);
    WRITE('          Press key ESC to menu .
TABLE1(10,5,70,7);TABLE(15,8,65,20);
GOTOXY(32,6);
WRITE('EDIT CODE NUMBER : ');
Y := 61;X := 6;Z := 2;
CODE := LOOP;
IF ORD(CH) = 27 THEN BEGIN I := 22;EXIT; END;
LEN:=0;
GOTOXY(17,24);
WRITE(' ENTER TO EDIT AND ESC TO MENU. ');
SEEK(F,0);
I := FILEPOS(F);
WHILE I <= FILESIZE(F) DO
BEGIN
SEEK(F,I);
IF CODE = SAC[FILEPOS(F)].B THEN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
BEGIN
  GOTOXY(20,10);
  WRITE('PART_NUMBER      : ');INVERSE;WRITE(' ');
  GOTOXY(40,10);WRITE(SA[1].B);
  NORMAL;
  GOTOXY(20,12);WRITE('ITEM_CLASS      : ');
  INVERSE;WRITE(' ');
  GOTOXY(40,12);WRITE(SA[1].C);
  NORMAL;
  GOTOXY(20,14);WRITE('UNITS_PRICE      : ');
  INVERSE;WRITE(' ');
  GOTOXY(40,14);WRITE(SA[1].D:5:2);
  NORMAL;
  GOTOXY(20,16);WRITE('UNIT_ON_HAND      : ');
  INVERSE;WRITE(' ');
  GOTOXY(40,16);WRITE(SA[1].E:6:0);
  NORMAL;
  GOTOXY(20,18);WRITE('TOTAL_UNIT_PRICE : ');
  INVERSE;WRITE(' ');
  GOTOXY(40,18);WRITE(SA[1].G:8:2);
  Y:=40;X:=10;Z:=2;K3:= SA[1].B;
  CHECK := 'F';
  WHILE CHECK = 'F' DO
    BEGIN
      Y:=40;X:=10;Z:=2;
      K3 := LOOP;
      IF LENGTH(K3) = 0 THEN BEGIN CHECK := 'T';END;
      IF ORD(CH) = 27 THEN BEGIN I := 22;EXIT; END;
      COUNT := 1;
    END
  END
  ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
```

```
RES := 1;
WHILE RES < FILESIZE(F)+1 DO
  BEGIN
    IF K3 = SA[RES].B THEN
      BEGIN
        GOTOXY(40,10);WRITE(K3);
        RES := FILESIZE(F)+1;
      END
    ELSE
      BEGIN
        COUNT := COUNT + 1;
        END;
        RES := RES + 1;
        END;
        IF COUNT > FILESIZE(F) THEN
          BEGIN
            SA[1].B := K3;
            CHECK := 'T';
            END;
          END;
        END;
        Y:=40;X:=12;Z:=6;K3 := SA[1].C;
        SA[1].C := LOOP;
        IF LENGTH(SA[1].C) = 0 THEN SA[1].C := K3;
        IF ORD(CH) = 27 THEN BEGIN SA[1].C := K3;J := 22;EXIT; END;
        VA1 := 'P';
        WHILE VA1 = 'P' DO
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
BEGIN
  Y:=40;X:=14;Z:=7;
  K2 := LOOP;
  IF LENGTH(K2) <> 0 THEN
    BEGIN
      VAL(K2,VA,RES);
      IF RES = 0 THEN BEGIN
        SACIJ.D := VA ;VA1 := 'H'; END
      ELSE
        BEGIN
          GOTOXY(40,14);WRITE(SACIJ.D:5:2);
          END;
        END
      ELSE
        VA1 := 'H';
      END;
      GOTOXY(40,14);WRITE(SACIJ.D:5:2);
      CH := READKEY;
      READ(CH);
      LEN :=1;
      M := M+ SACIJ.G;
      NORMAL;
    END;
    I := I+1;
  END;
  NORMAL;
  WINDOW(3,4,77,20);
  CLRSCR;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุที่แบ่ส่งเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

END;

PROCEDURE FIND;

VAR I : INTEGER;

CODE : STRING[3];

CH1 : CHAR;

BEGIN

CLEARWIN(47,4,57,9);

IF CHECK\_FILE\_OPEN = FALSE THEN EXIT;

TABLE1(10,5,70,7);TABLE1(10,8,70,16);

GOTOXY(20,6);

WRITE('WHAT DO YOU WANT TO FIND PART\_NUMBER : ');

Y := 59; X := 6; Z := 2;

CODE := LOOP;

SEEK(F,0);

FOR I := 0 TO FILESIZE(F) DO

BEGIN

SEEK(F,I);

IF CODE=SAC[FILEPOS(F)].B THEN

BEGIN

GOTOXY(13,10);

WRITE('ITEM\_CLESS = ',SAC[FILEPOS(F)].C);

GOTOXY(42,10);

WRITE('UNITS\_PRICE = ',SAC[FILEPOS(F)].D:3:2);

GOTOXY(13,12);

WRITE('UNITS ON\_HAND = ',SAC[FILEPOS(F)].E:3:0);

GOTOXY(42,12);

WRITE('TOTAL\_UNIT\_PRICE = ',SAC[FILEPOS(F)].G:5:2);

GOTOXY(29,14);

```
WRITE('NUMBER_WARE_HOUSE = ',SACFILEPOS(F)J.N_W);
```

```
CH1 : READKEY;
```

```
EXIT;
```

```
END;
```

```
END;
```

```
EXIT
```

```
END;
```

```
PROCEDURE STORK IN_OUT(COMM, CODE, MOUN : STRING);
```

```
VAR V1, V2 : INTEGER;
```

```
BEGIN
```

```
IF COMM = 'IP' THEN
```

```
FOR I := 1 TO 21 DO
```

```
IF SACIJ.B = CODE THEN
```

```
BEGIN
```

```
VAL(MOUN, V1, V2);
```

```
SACIJ.E := SACIJ.E + V1;
```

```
SACIJ.G := SACIJ.D * SACIJ.E;
```

```
END;
```

```
IF COMM = 'XP' THEN
```

```
FOR I := 1 TO 21 DO
```

```
IF CODE = SACIJ.B THEN
```

```
BEGIN
```

```
VAL(MOUN, V1, V2);
```

```
SACIJ.E := SACIJ.E - V1;
```

```
SACIJ.G := SACIJ.D * SACIJ.E;
```

```
END;
```

```
END;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PROCEDURE LIST3(VAR C : INTEGER);  
  VAR I : LONGINT;  
  Q1 : INTEGER;  
  BEGIN  
    Q1 := 0-C;  
    GOTOXY(1,1);WRITE(FILESIZE(F),' ',C);  
    FOR I := C TO FILESIZE(F) DO  
      BEGIN  
        GOTOXY(5,1+8+Q1);WRITELN(''  
        GOTOXY(6,1+8+Q1);WRITELN(I);  
        GOTOXY(17,1+8+Q1);WRITE(SACIJ.B);  
        GOTOXY(30,1+8+Q1);WRITE(SACIJ.C);  
        GOTOXY(41,1+8+Q1);WRITE(SACIJ.D:5:2);  
        GOTOXY(54,1+8+Q1);WRITE(SACIJ.E:4:0);  
        GOTOXY(65,1+8+Q1);WRITE(SACIJ.G:7:2);  
        GOTOXY(76,1+8+Q1);WRITE(SACIJ.N_W);  
        IF SACIJ.B = ' ' THEN  
          EXIT  
        ELSE  
          IF I = 12+C THEN  
            EXIT;  
          SEEK(F,I);  
          END;  
        END;  
      END;  
END;
```

```
PROCEDURE LIST;
```

```
  VAR G1 : CHAR;
```

```
  C : INTEGER;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
BEGIN

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
IF CHECK_FILE_OPEN = FALSE THEN EXIT;
    SEEK(F,1);
    WHILE NOT EOF(F) DO
        BEGIN
            C := FILEPOS(F);
            WRITE(C);
            LIST3(C);
            IF FILESIZE(F) = FILEPOS(F) THEN BEGIN
                G1 := READKEY ;
                WINDOW(3,8,77,20);CLRSCR;WINDOW(1,1,80,25);
                EXIT;END
        ELSE
            BEGIN
                G1 := READKEY;
                IF ORD(G1) = 27 THEN
                    EXIT
                ELSE
                    BEGIN
                        WINDOW(3,8,77,20);
                        CLRSCR;WINDOW(1,1,80,25);
                    END;
            END;
        END;
    END;
END;
END;
```

```
PROCEDURE BROWSE;
```

```
VAR C,Q,I,RES : INTEGER;
```

```
G1 : CHAR;
```

```
G : BOOLEAN;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

COUNT : INTEGER;
CHECK : STRING;
VA1,K2 : STRING;
VA : REAL;

BEGIN

  C := FALSE;

  IF CHECK_FILE_OPEN = FALSE THEN EXIT;

  SEEK(F,1);

  C := FILEPOS(F);

  LIST3(C);

  SEEK(F,1);

  Q := 8;

  GOTOXY(16,24);

  WRITE('          Press key ESC to Menu ');

  WHILE FILEPOS(F) < FILESIZE(F) DO
    BEGIN
      I := FILEPOS(F);

      WHILE G = FALSE DO BEGIN

        I := FILEPOS(F);

        GOTOXY(6,Q); INVERSE;WRITELN(I);NORMAL;

        GOTOXY(17,Q); INVERSE;WRITE(' ');

        GOTOXY(17,Q);WRITE(SAC I J.B);NORMAL;

        GOTOXY(30,Q); INVERSE;WRITE(' ');

        GOTOXY(30,Q);WRITE(SAC I J.C);NORMAL;

        GOTOXY(41,Q); INVERSE;WRITE(' ');

        GOTOXY(41,Q);WRITE(SAC I J.D:5:2);NORMAL;

        GOTOXY(54,Q); INVERSE;WRITE(' ');

        GOTOXY(54,Q);WRITE(SAC I J.E:4:0);NORMAL;

        GOTOXY(65,Q); INVERSE;WRITE(' ');

```

```
GOTOXY(65,Q);WRITE(SALIJ,G:7:2);
G1 := READKEY;
IF ORD(G1) = 27 THEN BEGIN
    G := TRUE;SEEK(F,FILESIZE(F));NORMAL;EXIT;END;
IF ORD(G1) = 0 THEN BEGIN NORMAL;
G1 := READKEY;
GOTOXY(6,Q);WRITE(' ');GOTOXY(6,Q);WRITELN(I);
GOTOXY(17,Q);WRITE(' ');GOTOXY(17,Q);WRITE(SALIJ.B);
GOTOXY(30,Q);WRITE(' ');GOTOXY(30,Q);WRITE(SALIJ.C);
GOTOXY(41,Q);WRITE(' ');GOTOXY(41,Q);
WRITE(SALIJ.D:5:2);
GOTOXY(54,Q);WRITE(' ');GOTOXY(54,Q);
WRITE(SALIJ.E:4:0);
GOTOXY(65,Q);WRITE(' ');GOTOXY(65,Q);
WRITE(SALIJ.G:7:2);
IF G1 = 'H' THEN BEGIN { Arrow Key UP-}
    I := I-1;
    Q := Q-1;SEEK(F,I);
    IF Q < 8 THEN BEGIN
        IF I >= 1 THEN BEGIN
            C:=FILEPOS(F);
            LIST3(C);
            SEEK(F,FILEPOS(F)-11);Q :=8;END
        ELSE
            BEGIN
                SEEK(F,1);Q := 8;I:=FILEPOS(F);END;
            END;
    END;
END;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
IF G1 = 'P' THEN { Arror Key DROWN-3}
  IF Q = 20 THEN
    IF I < FILESIZE(F) THEN BEGIN
      Q:=20;SEEK(F,FILEPOS(F)-11);
      C := FILEPOS(F);

      LIST3(C); I:=I+1;SEEK(F,I);END
    ELSE
      BEGIN
        I := FILESIZE(F);Q:=20;END
      ELSE
        IF I <> FILESIZE(F) THEN BEGIN
          I := I+1;
          SEEK(F,I);
          Q := Q+1; END;
        END;
      IF ORD(G1) = 13 THEN
        BEGIN
          K1 := 0;
          Y:=17;X:=Q;Z:=2;K3 := SA[FILEPOS(F)].B;

          CHECK : 'F';
          WHILE CHECK = 'F' DO
            BEGIN
              Y:=17;X:=Q;Z:=2;
              K3 := LOOP;

              IF LENGTH(K3) = 0 THEN BEGIN CHECK := 'T';END;

```

```
COUNT := 1;
RES := 1;
WHILE RES < FILESIZE(F)+1 DO
  BEGIN
    IF K3 = SA[RES].B THEN
      BEGIN
        GOTOXY(17,Q);WRITE(SA[FILEPOS(F)].B);
        RES := FILESIZE(F)+1;
      END
    ELSE
      BEGIN
        COUNT := COUNT + 1;
      END;
      RES := RES + 1;
    END;
    IF COUNT > FILESIZE(F) THEN
      BEGIN
        SA[FILEPOS(F)].B := K3;
        CHECK := 'T';
      END;
  END;
END;

Y:=30;X:=Q;Z:=6;K3 := SA[FILEPOS(F)].C;
SA[FILEPOS(F)].C := LOOP;
IF LENGTH(SA[FILEPOS(F)].C) = 0 THEN
  SA[FILEPOS(F)].C := K3;
IF ORD(CH) = 27 THEN BEGIN SA[FILEPOS(F)].C := K3;
I := 22;EXIT; END;
VA1 : 'P';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
WHILE VA1 = 'P' DO
BEGIN
    Y:=41;X:=Q;Z:=7;

    K2 := LOOP;

    IF LENGTH(K2) <> 0 THEN

        BEGIN

            VAL(K2,VA,RES);

            IF RES = 0 THEN BEGIN

                SAC11.D := VA ;VA1 := 'H'; END

            ELSE

                BEGIN

                    GOTOXY(41,Q);WRITE(SAC11.D:5:2);

                    END;

                END

            ELSE

                VA1 := 'H';

                END;

                GOTOXY(41,Q);WRITE(SAC11.D:5:2);

                CH := READKEY;

                READ(CH);

                LEN :=1;

                M := M+ SACFILEPOS(F)J.G;

                NORMAL;

            END;

        END;

    END;

    END;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{-----}
{ Load Number Warehouse-3
{-----}
```

PROCEDURE LOAD\_H;

VAR I,Y,X : INTEGER;

BEGIN

Y := 17;

X := 11;

GOTOXY(8,8);

FOR I := 1 TO 21 DO

BEGIN

STR(X,N\_W\_H[I]);

IF X <> Y THEN X := X+1 ELSE

BEGIN

X := X+4;

Y := Y+10;

END;

CODE\_D[I] := '0';

END;

FOR X := 1 TO FILESIZE(F) DO

BEGIN

FOR I := 1 TO 21 DO

BEGIN

IF SAC[X].N\_W <> '0' THEN

IF SAC[X].N\_W = N\_W\_H[I] THEN CODE\_D[I] := SAC[X].B;

END;

END;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
END; ฝ่าฝืนใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{ --- ----- - ----}  
{-Change Code Strock To wrehouse-}  
{----- ----- ----}
```

```
FUNCTION CHANG_C(VAR CODE : STRING):STRING;
```

```
VAR CODE1 : STRING;
```

```
Y : INTEGER;
```

```
CD : STRING;
```

```
Z : INTEGER;
```

```
RES : INTEGER;
```

```
BEGIN
```

```
FOR I := 1 TO FILESIZE(F) DO
```

```
BEGIN
```

```
IF CODE = SAC[I].B THEN
```

```
IF SAC[I].N_W <> 'O' THEN BEGIN
```

```
CHANG_C := SAC[I].N_W;EXIT;
```

```
END
```

```
ELSE
```

```
FOR Y := 1 TO 7 DO
```

```
BEGIN
```

```
FOR Z := 1 TO 21 DO
```

```
BEGIN
```

```
VAL(COPY(N_W_H[Z],2,1),X,RES);
```

```
IF X = Y THEN
```

```
IF CODE_D[Z] = 'O' THEN
```

```
BEGIN
```

```
CODE_D[Z] := SAC[I].B;
```

```
SAC[I].N_W := N_W_H[Z];
```

```
CHANG_C := N_W_H[Z];EXIT;
```

```
END;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
END;
```

```
END;
```

```
END;
```

```
END;
```

```
{----- -}

```

```
{ Check Ware House Space-}

```

```
{ ----- -}

```

```
FUNCTION CHECK_WH(VAR CODE : STRING):STRING;
```

```
BEGIN
```

```
FOR I := 1 TO FILESIZE(F) DO
```

```
IF CODE = SACIJ.B THEN
```

```
IF SACIJ.E = 0 THEN BEGIN
```

```
CHECK_WH := '0'; EXIT; END
```

```
ELSE BEGIN
```

```
CHECK_WH := SACIJ.N_W;EXIT;END;
```

```
END;
```

```
{|-----|}

```

```
{| COMMUNICATION TO WAREHOUSE |}

```

```
{| COMMAN IS IP => I , XP => O |}

```

```
{| ODE NUMBER FIND TO NUMBAR WAREHOUSE |}

```

```
{| GET TO CODE1 |}

```

```
{| CODE SEND = COMMAN+CODE1 |}

```

```
{|-----|}

```

```
PROCEDURE SENDTX;
```

```
VAR COMMAN, CODE, CODE1, MOUN, TXT : STRING;
```

```
VA : REAL;
```

```
RES : INTEGER;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BEGIN

IF CHECK\_FILE\_OPEN = FALSE THEN

BEGIN CLEARWIN(29,5,73,14);EXIT;END;

GOTOXY(45,6);WRITELN('WARE HOUSE');

GOTOXY(31,8);

WRITE('DO YOU WANT IMPORT OR EXPORT (IP/XP) ? ');

Y:=8; X:=71; Z:=1; COMMAN:=LOOP;

IF ORD(CH) = 27 THEN BEGIN CLEARWIN(29,5,73,14);

EXIT; END;

GOTOXY(40,10); WRITE('CODE STROCK ? ');

Y:=10; X:=54; Z:=2; CODE:=LOOP;

IF ORD(CH) = 27 THEN BEGIN CLEARWIN(29,5,73,14);

EXIT; END;

IF COMMAN = 'IP' THEN

BEGIN

GOTOXY(35,12);

WRITE(' HOW MANY STROCK MOUNT ? ');

Y:=12; X:=60; Z:=1; MOUN:=LOOP;

IF ORD(CH) = 27 THEN BEGIN CLEARWIN(29,5,73,14);

EXIT; END;

END;

IF COMMAN = 'IP' THEN

BEGIN

COMMAN : 'I';

CODE1 := CHANG\_C(CODE);

COMMAN := COMMAN+CODE1;

IF SACI].E = 0 THEN

BEGIN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
SENDTXT('G'); {-COMMAND 'G' START LIFT GO TO HOME-3  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TXT := '';
WHILE(TXT <> '*OK#') DO TXT := RXCHAR;
DELAY(100);
SENDDTXT(COMMAN);
TXT := RXCHAR;
IF TXT = '*OK#' THEN
    BEGIN
        GOTOXY(17,24);WRITE('      OK      ');
        READLN;
    END
ELSE
    BEGIN
        GOTOXY(17,24);
        WRITE(COMMAN, '      IS ERROR');
        READLN;CLEARWIN(29,5,73,14);EXIT;
    END;
VAL(MOUN,VA,RES);
SACIJ.E := SACIJ.E+VA;
SACIJ.G := SACIJ.D*SACIJ.E;
WRITEF(I);
CLEARWIN(29,5,73,14);EXIT;
END;
END;
IF COMMAN = 'XP' THEN
    BEGIN
        COMMAN := 'O';
        CODE1 := CHECK_WH(CODE);
        COMMAN := COMMAN+CODE1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
IF SACIJ.E <> 0 THEN
BEGIN
SENDTXT(COMMAN);DELAY(100);
TXT := RXCHAR;
IF TXT = '*OK#' THEN
BEGIN
GOTOXY(20,24);
WRITE('          OK          ');
END
ELSE
BEGIN
GOTOXY(20,24);
WRITE(COMMAN,'          IS ERROR ');
CLEARWIN(29,5,73,14);EXIT;
END;
SENDTXT('S');      {-COMMAND 'S' IS OUT FROM WAREHOUSE-}
SACIJ.E := SACIJ.E-SACIJ.E;SACIJ.G := 0;
SACIJ.N_W := '0';
WRIREF(I);
FOR I := 1 TO 21 DO
IF N_W_HC[I] = CODE1 THEN
CODE_DC[I] := '0';
CLEARWIN(29,5,73,14);EXIT;
END;
END;
GOTOXY(20,24);
WRITE('          COMMAND ',COMMAN,' NOT FOUND ');
READLN;CLEARWIN(29,5,73,14);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ENE; ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PROCEDURE PRINT;
  VAR I,CC : INTEGER;
  TOTAL : REAL;
  ST : STRING;
  BEGIN
    TOTAL := 0;
    SEEK(F,1);
    FOR I := 1 TO 6 DO
      WRITELN(LST);
      WRITELN(LST, '                                WARE HOUSE ');
      WRITELN(LST);
      WRITELN(LST, '          REPORT STOCK IN
        THE DATABASE ON COMPUTER ');
      WRITELN(LST);
      WRITELN(LST, '=====
        =====');
      WRITELN(LST, ' #REC_NO      CODE      NAME
                    PRICE      MOUNT      TOTAL ');
      WRITELN(LST, '=====
        =====');
    FOR I := 1 TO FILESIZE(F) DO
      BEGIN
        SEEK(F,1);
        STR(I,ST);
        FOR CC := 1 TO 5-LENGTH(ST) DO
          WRITE(LST, ' '); WRITE(LST,ST);
          WRITE(LST, '          ',SA[FILEPOS(F)].B);
          WRITE(LST, '          ',SA[FILEPOS(F)].C);
          FOR CC := 1 TO 7-LENGTH(SA[FILEPOS(F)].C) DO
            WRITE(LST, ' ');
```

```
WRITE(LST, ' ');
STR(SACFILEPOS(F)J.D:5:2,ST);
FOR CC := 1 TO 14-LENGTH(ST) DO
    WRITE(LST, ' ');WRITE(LST,ST);
STR(SACFILEPOS(F)J.E:3:0,ST);
FOR CC := 1 TO 10-LENGTH(ST) DO
    WRITE(LST, ' '); WRITE(LST,ST);
STR(SACFILEPOS(F)J.G:6:2,ST);
FOR CC := 1 TO 17-LENGTH(ST) DO
    WRITE(LST, ' '); WRITELN(LST,ST);
TOTAL := TOTAL+SACFILEPOS(F)J.G
END;
WRITELN(LST, '===== = ===== =
===== == ');
WRITE(LST, '          TOTAL CURRENCY OF STOCK');
STR(TOTAL:5:2,ST);
FOR CC := 1 TO 35-LENGTH(ST) DO
    WRITE(LST, ' ');WRITELN(LST,ST);
WRITELN(LST, '=====
===== == ');
CLEARWIN(47,4,57,9);
END;

BEGIN
END.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาโทฉบับนี้ เสร็จสิ้นลงได้ด้วยความร่วมมือและความอนุเคราะห์จากบุคคลและกลุ่มบุคคลต่าง ๆ หลายฝ่ายทั้งในส่วนของสถาบันครูอาจารย์และเพื่อน ๆ โดยเฉพาะอย่างยิ่ง อาจารย์ภากร หุตสังภาค ที่กรุณามาเป็นอาจารย์ที่ปรึกษาปริญญาโท และช่วยให้คำแนะนำต่าง ๆ จนปริญญาโทฉบับนี้ เสร็จสิ้นลงจนสมบูรณ์ ท้ายที่สุดนี้คณะผู้จัดทำต้องขอขอบคุณอาจารย์ทุกท่านที่มีส่วนช่วยให้คณะผู้จัดทำมีโอกาสศึกษาในสถาบันอันทรงเกียรติแห่งนี้และขอขอบคุณเพื่อน ๆ พี่ ๆ น้อง ๆ ทุกคน

คณะผู้จัดทำ

หนังสืออ้างอิง

- [1]. BORLAND, TURBO PASCAL 4.0, 1987.
- [2]. STEPHEN O'BRIEN, TURBO PASCAL 6 THE COMPLETE REFERENCE, 1991.
- [3]. L.WAYNE HOPN, STRUCTURED PROGRAMMING IN TURBO PASCAL, 1990.
- [4]. ROBERT JOURDAIN, PROGRAMMER'S PROBLEM SOLVER FOR THE IBM PC.XT&AT. 1986.
- [5]. น.ต.ดร.ไพศาล สงวนหม่ม รศ.ยีน กุวัรรวณ, การสื่อสารข้อมูล และ ไมโครคอมพิวเตอร์เน็ตเวิร์ค,

คู่มือการใช้งาน  
โปรแกรมการจัดเก็บสินค้าด้วยคอมพิวเตอร์  
COMPUTERIZED WAREHOUSE CONTROL



นายกิตติ อัครปรีดี	34.162142
นายพรกิตติ สุวรรณพงษ์	34.162163
นายพฤษ ล้อมเมตตา	34.162164

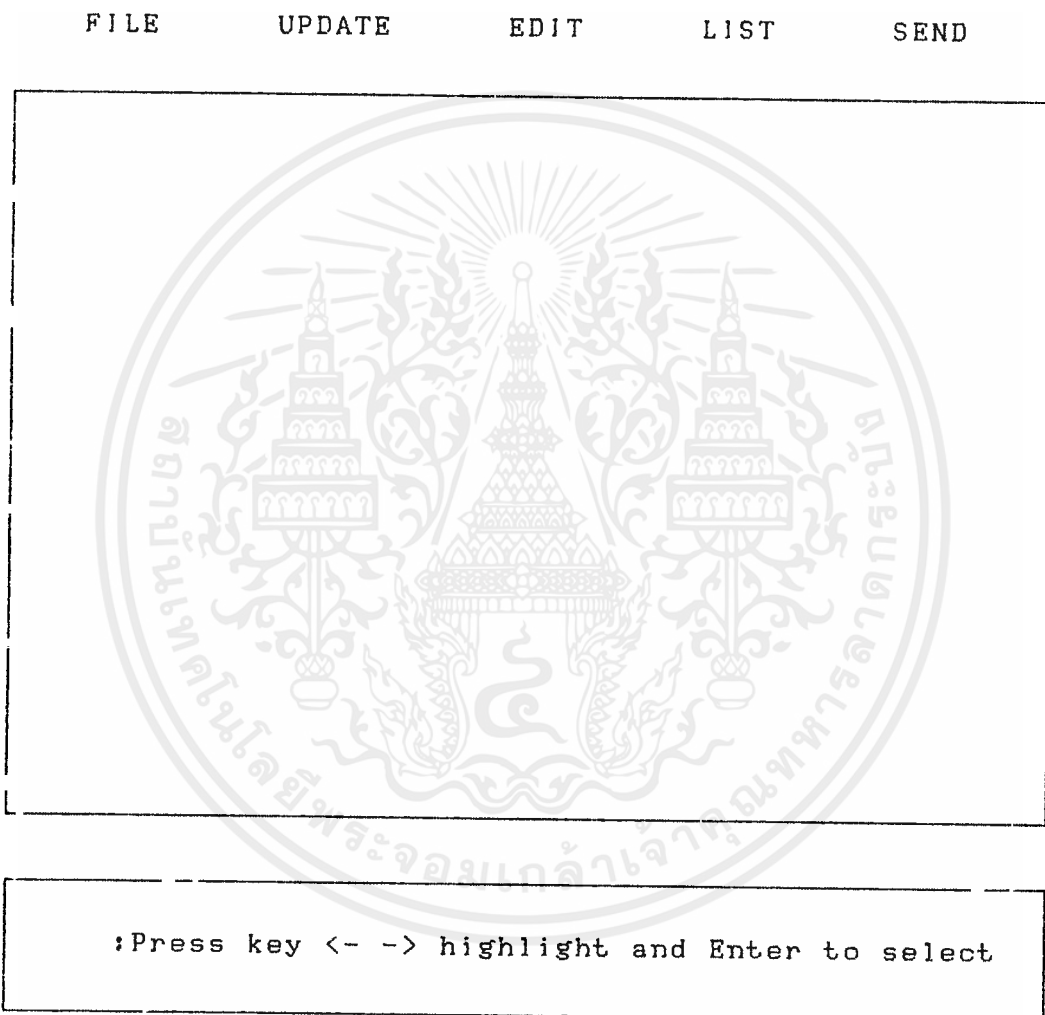
ปริญญาานิพนธ์สำหรับปริญญาอุตสาหกรรมศาสตรบัณฑิต  
สาขาวิชาเทคโนโลยีคอมพิวเตอร์อุตสาหกรรม  
ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้งานโปรแกรม การจัดเก็บสินค้าด้วยคอมพิวเตอร์

ลักษณะของโปรแกรม การจัดเก็บสินค้าด้วยคอมพิวเตอร์ นั้นจะเป็นลักษณะของ PULLDOWN MENU ทั้งหมด เพราะเราจะเน้นความสะดวกในการใช้งาน โดยจะขออธิบายการใช้งานดังนี้

เมื่อทำการโหลดโปรแกรมมาใช้งานแล้วจะปรากฏที่หน้าจอดังนี้



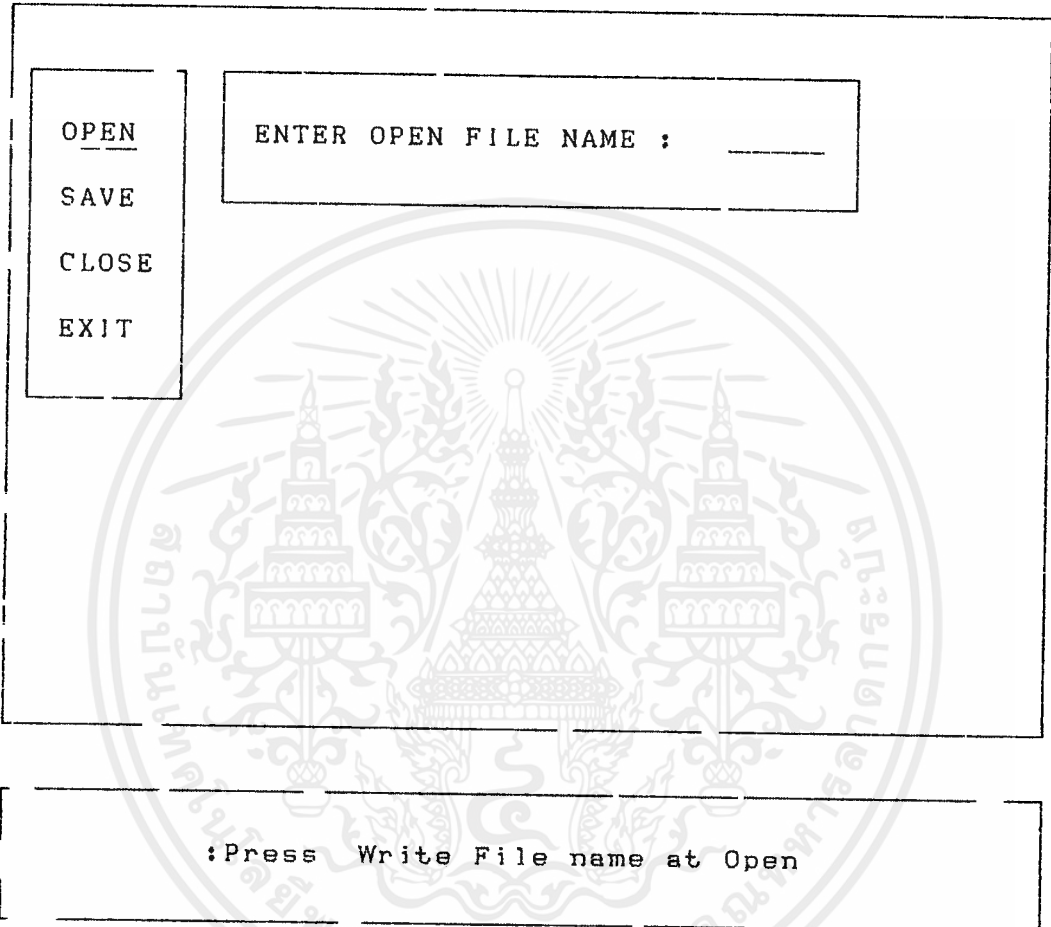
รูปที่ 1 เมนูหลัก

เมนูหลัก จะประกอบด้วย เมนูย่อย 5 เมนู ได้แก่ FILE, UPDATE, EDIT, LIST, SEND โดยแต่ละเมนูจะทำงานดังนี้



1.1.OPEN เมื่อเลื่อนแถบแสงไปยัง OPEN แล้วกด Enter จะปรากฏ หน้าจอตั้งรูปที่ 3

FILE UPDATE EDIT LIST SEND



รูปที่ 3 เมื่อเลือกรายการ OPEN

ให้ใส่ชื่อไฟล์ที่ต้องการเปิดเพิ่มข้อมูล โดยไฟล์นั้นต้องเป็นไฟล์ที่ ถูกสร้างไว้แล้วโดยโปรแกรมนี้และ ถ้าต้องการออกจากรายการ OPEN นี้ให้ กด ESC

1.2.SAVE เมื่อเลื่อนแถบแสงไปที่รายการ SAVE แล้วกด Enter เพื่อเป็นการเก็บข้อมูล เมื่อมีการแก้ไขข้อมูลหรือเลิกใช้โปรแกรม

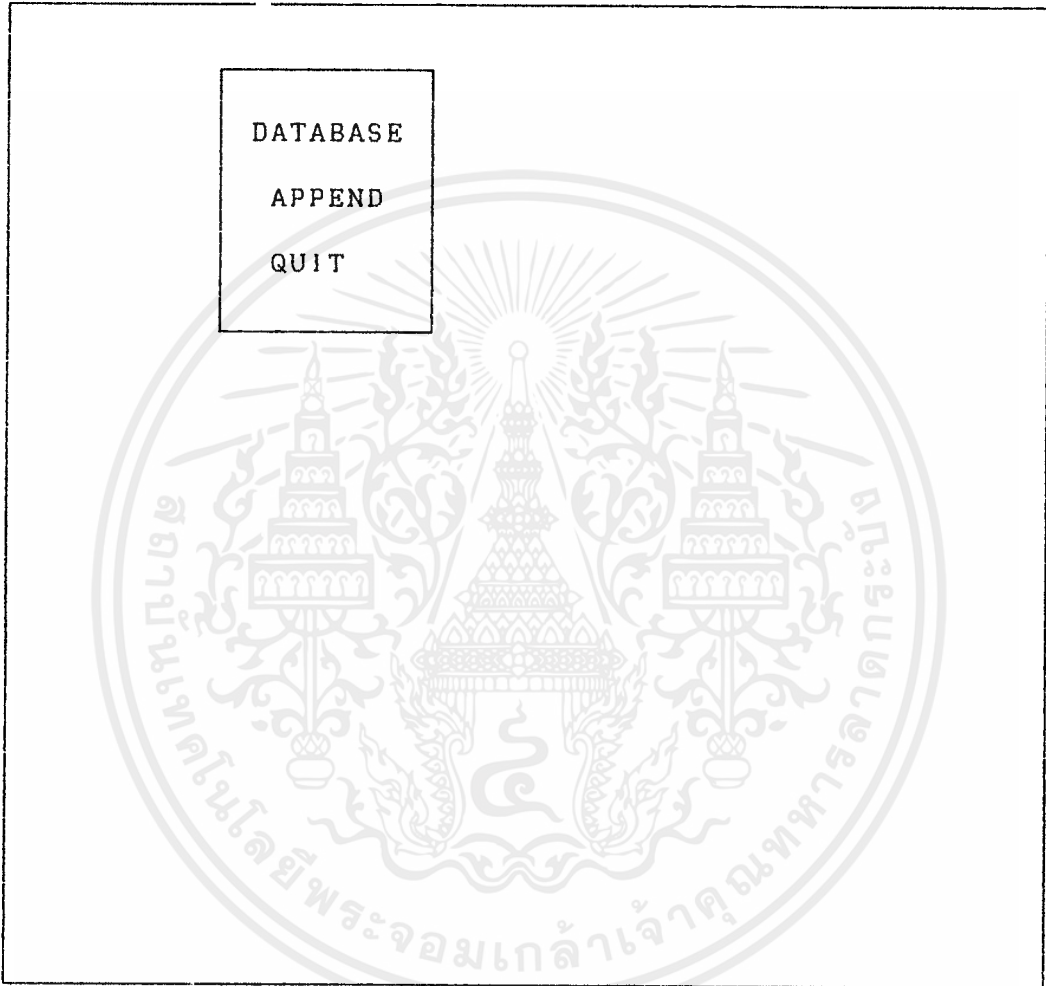
1.3.CLOSE เมื่อเลื่อนแถบแสงไปที่รายการ CLOSE แล้วกด Enter เพื่อทำการปิดแฟ้มข้อมูล เมื่อต้องการเลิกใช้ไฟล์นั้น

1.4.EXIT เมื่อเลื่อนแถบแสงไปที่รายการ EXIT แล้วกด Enter เมื่อ

ต้องการออกจากโปรแกรมการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.UPDATE ให้เลื่อนแถบแสงไปที่เมนู UPDATE แล้วกด Enter จะปรากฏรายการขึ้นที่หน้าจอดังรูปที่ 4

FILE      UPDATE      EDIT      LIST      SEND

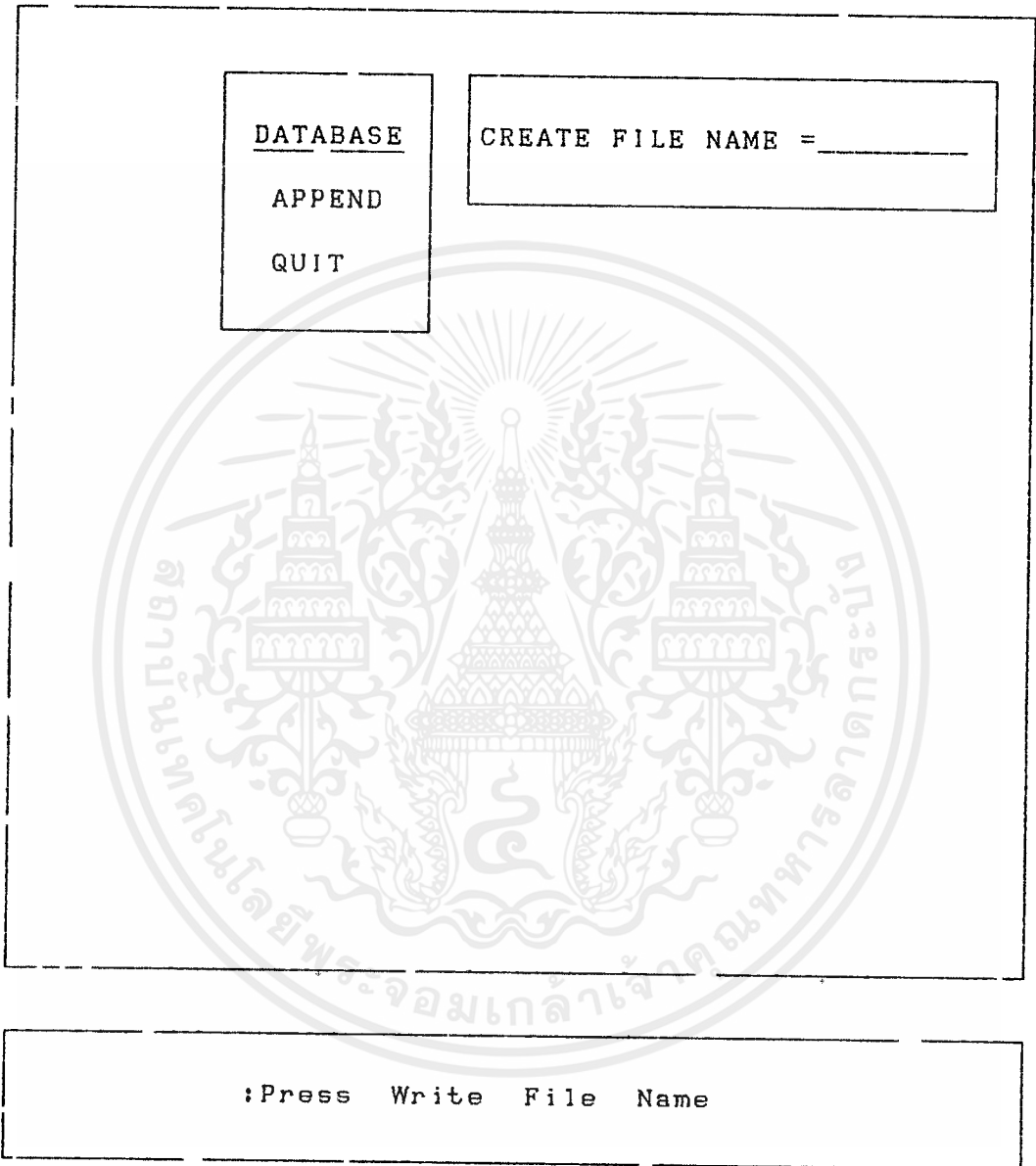


:Press key Move highlight and Enter to select

รูปที่ 4 เมื่อเลือกเมนู UPDATE  
ซึ่งจะมีรายการดังนี้ DATABASE, APPEND, QUIT โดยแต่ละรายการจะ  
ถูกนำมาใช้งานดังนี้

2.1.DATABASE ให้เลื่อนแถบแสงไปที่รายการ DATABASE แล้วกด Enter จะปรากฏกรอบข้อความดังรูปที่ 5

FILE            UPDATE            EDIT            LIST            SEND



รูปที่ 5 เมื่อเลือกใช้รายการ DATABASE

รายการ DATABASE นี้เป็นรายการที่ใช้สร้างไฟล์ข้อมูลขึ้นใหม่ โดยใส่ชื่อไฟล์ที่ต้องการสร้างแล้วกด Enter จะปรากฏจอภาพดังรูปที่ 6

FILE UPDATE EDIT LIST SEND

CREATE STOCK DATABASE	
FILE NAME = _____	
RECORD = 0	
PART NUMBER : _____	ITEM_CLASS : _____
UNIT PRICE : _____	UNIT_ON_HAND : _____
TOTAL_UNIT_PRICE : _____	
:Press Key Esc to Menu	

รูปที่ 6 เมื่อบ้อนชื่อไฟล์ข้อมูลและกด Enter

ซึ่งแต่ละช่องรายการจะรับค่าข้อมูลต่างกัันดังนี้

PART\_NUMBER : \_\_\_\_\_ จะรับค่ารหัสสินค้า

ITEM\_CLASS : \_\_\_\_\_ จะรับชื่อสินค้า

UNIT\_PRICE : \_\_\_\_\_ จะรับค่าราคาสินค้า (ต่อชิ้น)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนช่องรายการดังนี้ผู้ใช้ไม่ต้องป้อนข้อมูลเข้าไปเพราะช่องเหล่านี้จะถูกใช้งานโดยโปรแกรมซึ่งจะขออธิบายในหัวข้อต่อ ๆ ไป

UNIT\_ON\_HAND : \_\_\_ \_\_\_ จะเป็นจำนวนของสินค้าชนิดนั้น

TOTAL\_UNIT\_PRICE : \_\_\_ จะเป็นค่าผลรวมของสินค้าทั้งหมด

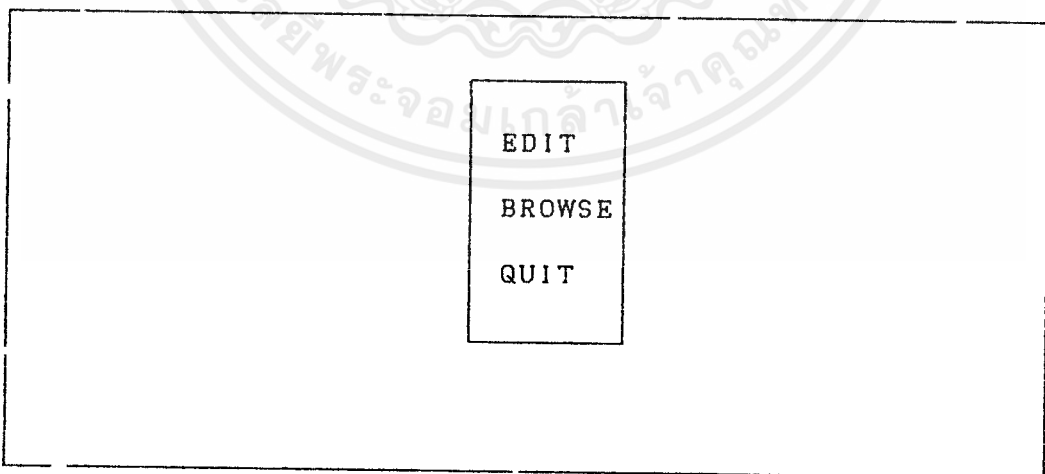
เมื่อเสร็จจากการป้อนข้อมูลหรือต้องการจะออกไปสู่เมนู UPDATE ให้กด Esc

2.2. APPEND จากรูปที่ 4 เมื่อเลือกรายการ APPEND แล้วกด Enter จะปรากฏช่องข้อมูลดังรูปที่ 6 แต่จะเป็นช่องข้อมูลใน RECORD สุดท้ายซึ่งไม่มีข้อมูลอยู่ เพื่อเป็นการเพิ่มเติมข้อมูลในกรณีที่ต้องการเพิ่มชนิดของสินค้าในไฟล์ที่ถูกสร้างขึ้นแล้วและยังเปิดอยู่ และเมื่อเสร็จจากการป้อนข้อมูลหรือต้องการจะออกไปสู่เมนู UPDATE ให้กด Esc

2.3 QUIT ใช้เมื่อต้องการออกจากเมนู UPDATE

3. EDIT เมื่อเลื่อนแถบแสงมายังเมนู EDIT แล้วกด Enter จะปรากฏรายการดังรูปที่ 7

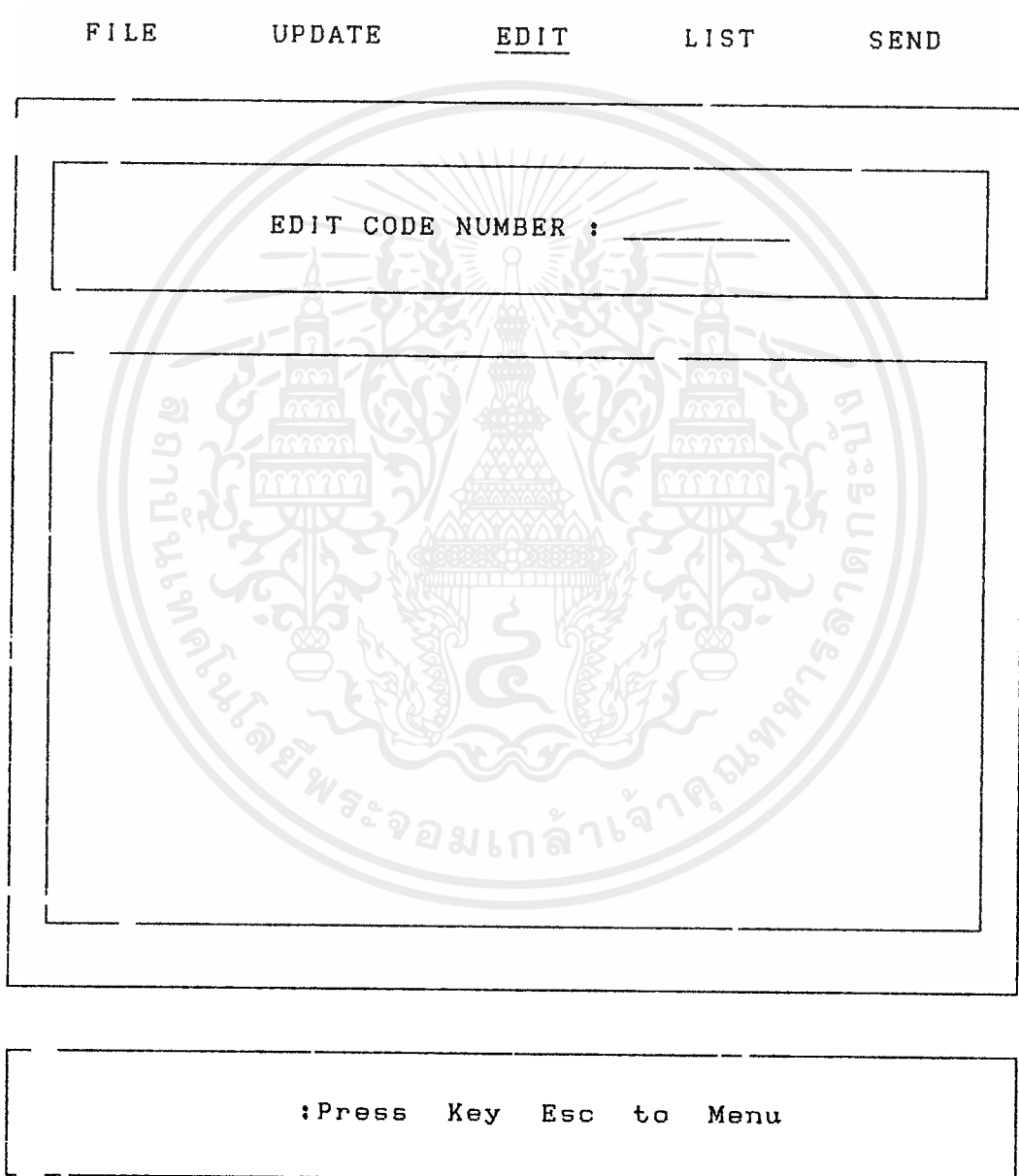
FILE UPDATE EDIT LIST SEND



: Press key move highlight and Enter to select

ซึ่งจะมีรายการต่าง ๆ ดังนี้ EDIT, BROWN, QUIT ซึ่งจะมีลักษณะการทำงานดังนี้

3.1. EDIT ให้เลื่อนแถบแสงไปที่รายการ EDIT แล้วกด Enter จะปรากฏจอภาพดังรูปที่ 8



รูปที่ 8 เมื่อเลื่อนแถบสว่างมาที่รายการ EDIT แล้วกด Enter

ในรายการ EDIT นี้เป็นการแก้ไขข้อมูลใน RECORD ที่ต้องการโดย

ใส่รหัสของสินค้าที่ต้องการแก้ไขแล้วกด Enter จะปรากฏดังรูปที่ 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FILE UPDATE EDIT LIST SEND

EDIT CODE NUMBER : \_\_\_\_\_

PART\_NUMBER : \_\_\_\_\_  
 ITEM\_CLASS : \_\_\_\_\_  
 UNIT\_PRICE : \_\_\_\_\_  
 UNIT\_ON\_HAND : \_\_\_\_\_  
 TOTAL\_UNIT\_PRICE : \_\_\_\_\_

: Press Key Esc to Menu

รูปที่ 9 จอภาพจะแสดงรายการใน RECORD ที่ต้องการแก้ไข

เมื่อเสร็จจากการแก้ไขข้อมูลหรือต้องการจะออกไปสู่รายการ EDIT ให้กด Esc

3.2.BROWSE จากรูปที่ 7 เมื่อเลื่อนแถบแสงไปที่รายการ BROWSE แล้วกด Enter จะเป็นการแก้ไขข้อมูลทุก RECORD ดังรูปที่ 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FILE UPDATE EDIT LIST SEND

#REC	ON	CODE	NAME	PRICE	MOUNT	TOTAL
1.						
2.						
3.						
4.						
5.						

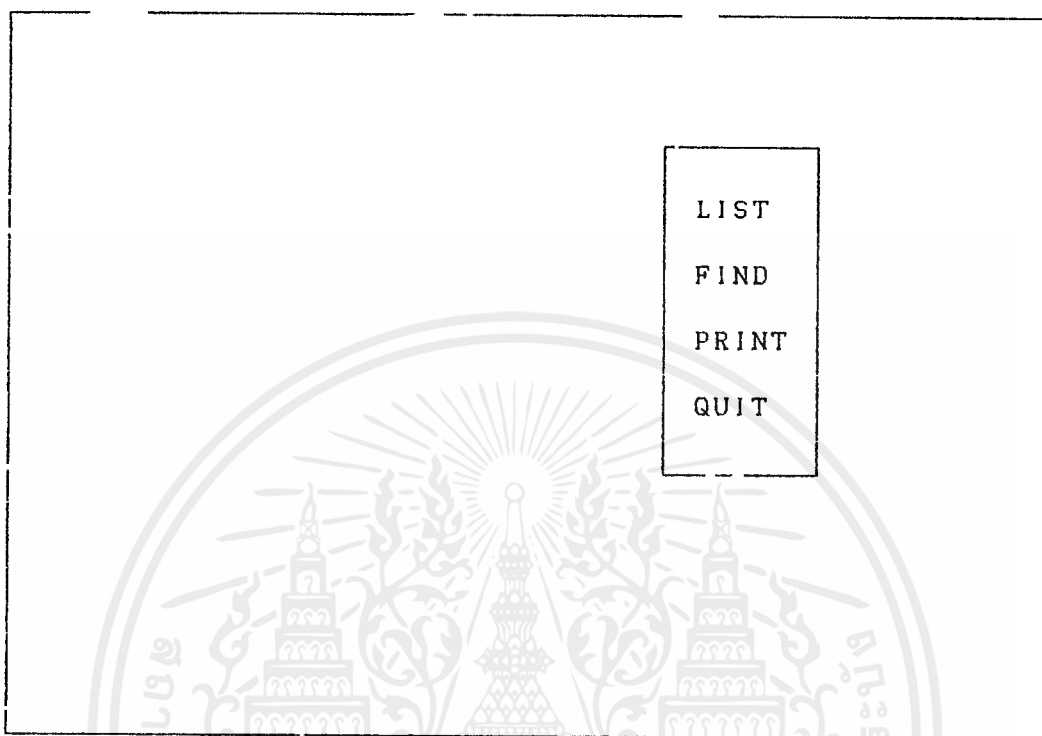
:Press Key Esc to Menu

รูปที่ 10 แสดงการแก้ไขข้อมูลทุก ๆ RECORD ในไฟล์ข้อมูล  
สามารถที่จะเลือก RECORD ที่จะแก้ไขได้โดยใช้แถบแสงเลื่อนไปยัง  
RECORD ที่ต้องการ

3.3. QUIT เมื่อต้องการออกจากรายการ BROWN สู่เมนู EDIT

4. LIST ให้เลื่อนแถบแสงไปที่เมนู LIST แล้วกด Enter จะ  
ปรากฏในจอภาพดังรูปที่ 11

FILE UPDATE EDIT LIST SEND



; Press Key Move highlight and Enter to select

รูปที่ 11 เมื่อเลื่อนแถบสว่างมายังเมนู LIST แล้วกด Enter

เมื่อเลื่อนแถบสว่างมายังเมนู LIST แล้วกด Enter จะปรากฏรายการต่าง ๆ ดังนี้ LIST, FIND, PRINT, QUIT ซึ่งมีลักษณะการใช้งานดังนี้

4.1. LIST เมื่อเลื่อนแถบแสงไปที่รายการ LIST แล้วกด Enter จะปรากฏภาพบนจอดังรูปที่ 13 ซึ่งรายการ LIST นี้จะใช้ในการขอดูข้อมูลทุก RECORD ในไฟล์ข้อมูลซึ่งถูกเปิดอยู่ โดยตัวเลขสองหลักท้ายบรรทัดจะแสดงการนำสินค้าเข้าไปเก็บใน WARE HOUSE ถ้ามีเลขเป็น 00 แสดงว่าไม่มีสินค้าใน WARE HOUSE ถ้ามีเลขอื่นๆ นอกจาก 00 แสดงว่า WARE HOUSE ได้นำสินค้าในแคว้นนั้นไปเก็บไว้ยัง WARE HOUSE ที่มีหมายเลขช่องนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FILE            UPDATE            EDIT            LIST            SEND

#REC_ON	CODE	NAME	PRICE	MOUNT	TOTAL
1.	_____	_____	_____	_____	00
2.	_____	_____	_____	_____	00
3.	_____	_____	_____	_____	00
4.	_____	_____	_____	_____	00
5.	_____	_____	_____	_____	00
6.	_____	_____	_____	_____	00
7.	_____	_____	_____	_____	00

:Press Key Esc to Menu

รูปที่ 12 แสดงการขอดูข้อมูลทุก ๆ RECORD ในไฟล์ที่เปิดอยู่

4.2.FIND เมื่อเลื่อนแถบแสงไปที่รายการ FIND แล้วกด Enter จะปรากฏภาพบนจอดังรูปที่ 13

FILE UPDATE EDIT LIST SEND

WHAT DO YOU WANT TO FIND PART\_NUMBER : \_\_\_\_\_

:Press Key Esc to Menu

รูปที่ 13 จอภาพแสดงการรหัสข้อมูลที่ต้องการค้นหา  
 เมื่อป้อนรหัสของข้อมูลที่ต้องการค้นหาแล้วกด Enter จอภาพจะ  
 ปรากฏดังรูปที่ 14

FILE UPDATE EDIT LIST SEND

WHAT DO YOU WANT TO FIND PART\_NUMBER : \_\_\_\_\_

ITEM\_CLASS = \_\_\_\_\_ UNIT\_PRICE = \_\_\_\_\_  
 UNIT\_ON\_HAND=\_\_\_\_\_ TOTAL\_UNIT\_PRICE=\_\_\_\_\_

:Press Key Esc to Menu

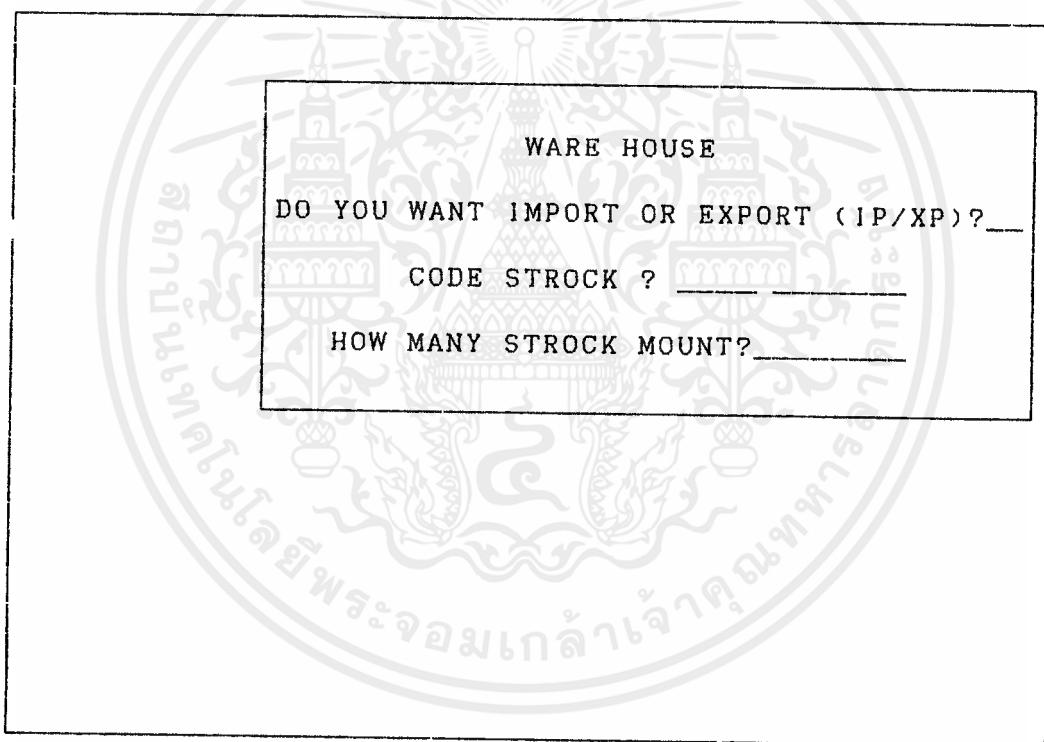
เมื่อเสร็จจากการค้นหาข้อมูลหรือต้องการจะออกไปสู่เมนู LIST ให้กด Esc

4.3.PRINT ให้เลื่อนแถบแสงไปที่รายการ PRINT แล้วกด Enter PRINTER จะทำการพิมพ์ข้อมูลของไฟล์ที่เปิดอยู่

4.4.QUIT เมื่อต้องการออกจากเมนู LIST สู่เมนูหลัก

5.SEND ให้เลื่อนแถบแสงไปที่รายการ SEND แล้วกด Enter จอภาพจะปรากฏภาพดังรูปที่ 15

FILE UPDATE EDIT LIST SEND



:Press Key Esc to Menu

รูปที่ 15 จอภาพแสดงการสื่อสารข้อมูลกับ WAREHOUSE  
เมื่อต้องการนำสินค้าเข้าให้ใส่รหัส IP และเมื่อต้องการนำสินค้าออก  
ให้ใส่รหัส XP จากนั้นให้ใส่รหัสและจำนวนของสินค้าที่ต้องการนำเข้าหรือออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## หมายเหตุ

1. โปรแกรมนี้เป็นโปรแกรมที่ใช้งานเฉพาะอย่าง
2. การสร้างฐานข้อมูลมีขนาดจำกัด
3. โปรแกรมนี้จะอยู่ในชื่อ WARE\_H.PAS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้