

## สัปดาห์ที่ 1

**P หัวข้อ 1** ทำความเข้าใจเนื้อหา และข้อตกลงต่าง ๆ ของวิชา และวิธีการเรียนอย่างเป็นระบบ

**2 สรุปคำบรรยาย :**

- .. งานทุกชิ้นที่สั่ง ไม่ใช่งานใหญ่ ไม่ว่าจะขึ้นต่อสัปดาห์ต้อง ส่งต้นชั่วโมง ในห้องเรียนของสัปดาห์ต่อไป
- .. Clipper เป็นภาษาตัวแปลภาษาขนาดเล็ก ที่เหมาะกับการศึกษาการเขียนโปรแกรมเพื่อพัฒนาฐานข้อมูล
- .. Compiler คือตัวแปรภาษา ที่แปลภาษาต้นฉบับทั้งหมด ไม่เหมือน Interpreter ที่แปลทีละคำสั่ง
- .. ตัวแปลภาษาของ Clipper ถ้าเลือกเฉพาะแฟ้มที่สำคัญสามารถจัดเก็บในแผ่น Diskette 1.44 แผ่นเดียวได้

**2 สรุปกิจกรรม :**

แจกเอกสาร course syllabus เพื่อให้ทราบเงื่อนไข การให้คะแนน และการส่งงาน

**2 รายละเอียดกิจกรรม :**

- แจกเอกสาร course syllabus
- อธิบายคะแนนที่อยู่ใน course syllabus ถึงเกณฑ์การตัดเกรด
 

คะแนนมากกว่า หรือเท่ากับ 90		ได้เกรด A
คะแนนมากกว่าหรือเท่ากับ 85	แต่น้อยกว่า 90	ได้เกรด B+
คะแนนมากกว่าหรือเท่ากับ 80	แต่น้อยกว่า 85	ได้เกรด B
คะแนนมากกว่าหรือเท่ากับ 75	แต่น้อยกว่า 80	ได้เกรด C+
คะแนนมากกว่าหรือเท่ากับ 70	แต่น้อยกว่า 75	ได้เกรด C
คะแนนมากกว่าหรือเท่ากับ 65	แต่น้อยกว่า 70	ได้เกรด D+
คะแนนมากกว่าหรือเท่ากับ 60	แต่น้อยกว่า 65	ได้เกรด D
คะแนนน้อยกว่า 60		ได้เกรด F
- อธิบายเกณฑ์การให้คะแนนเก็บ คะแนนเข้าห้อง + สอบย่อย + ส่งงาน
 

ทุกคนจะได้คะแนนเต็ม 15 % ทุกคน

แต่ถ้าวันใดไม่เข้าเรียน จะหักคะแนน 1 %

แต่ถ้าวันใดขาดสอบย่อย หรือไม่ตั้งใจทำสอบย่อยอย่างจริงจัง จะหักคะแนน 1 %

แต่ถ้าวันใดนัดส่งงานแล้วไม่ส่ง จะหัก 1 %

สำหรับชิ้นงานย่อย ขึ้นใดที่นักศึกษา คัดลอกมาส่ง หรือให้เพื่อนลอก จะหัก 1%

ถ้าหากหักไปเกิน 15% จะไม่ไปหักส่วนอื่น จะถือว่าหักคะแนนเพียง 15% เท่านั้น
- อธิบายเหตุผลที่เลือกใช้ Clipper เป็นภาษาสำหรับสอนการเขียนโปรแกรมระบบจัดการฐานข้อมูล
- อธิบายเหตุผลที่ไม่เลือกใช้ภาษาประเภท visual เช่น Visual basic หรือ Delphi หรือ Oracle เป็นต้น
- เปิดให้มีการซักถาม และเล่าประสบการณ์การเขียนโปรแกรมของนักศึกษา
 

เช่น จบมาจากที่ไหน, รู้อะไรพิเศษ ๆ มาบ้าง, ทำงานอะไรมาบ้าง, ใช้คอมพิวเตอร์แบบไหนกัน เป็นต้น

**2 เขียนแผ่นใส :**

- เหตุผลที่เลือกใช้ Clipper
  1. รูปแบบภาษาใกล้เคียงกับ FOXPRO หรือ DBASE
  2. โปรแกรมสำหรับแปลภาษามีขนาดเล็ก(จุใน DISK แผ่นเดียวได้)
  3. วิชานี้เป็นวิชาพื้นฐานสำหรับการศึกษาระดับปริญญาตรี
- เหตุผลที่ไม่เลือกใช้ภาษาประเภท Visual
  1. หากเข้าใจ Structure programming และ Database concept การไปเขียนลักษณะอื่น จะไม่ยาก
  2. ไม่เหมาะกับนักศึกษา ที่เน้นด้าน Structure programming ที่ใช้ 3 หลักการในการพัฒนาโปรแกรม
  3. การบังคับให้มี Visual object compiler ที่บ้านนักศึกษานั้นไม่จ่าย หลายคนมีเครื่องรุ่นเก่า

**P หัวข้อ 2** ศึกษาการเขียนโปรแกรมด้วย clipper, การแปลภาษา และการประมวลผล ทำความเข้าใจการเขียนโปรแกรมสำหรับทำซ้ำ ด้วยคำสั่ง for และ for ซ้อน for

## 2 สรุปคำบรรยาย :

- .. Autoexec.bat คือแฟ้มที่ใช้สั่งงานโปรแกรมต่าง ๆ ในตอนเปิดเครื่อง และกำหนดค่าตามต้องการ
- .. Pulldown menu คือเมนูแบบดึงลง คล้าย Menu bar ของ windows ที่เราใช้อยู่
- .. DBF คือแฟ้มเก็บข้อมูลในลักษณะของ Field และ Record เป็นมาตรฐานเดียวกับแฟ้มข้อมูลที่ถูกรสร้างโดยโปรแกรม Dbase III plus ทั้งข้อมูลที่เกิดใน DBF (Database file) และโปรแกรมที่มีนามสกุลเป็น PRG บางโปรแกรมสามารถนำมาใช้ใน Clipper ได้ทันทีโดยไม่ต้องแก้ไขอะไรเลย
- .. DBU (Database Utility) คือโปรแกรมที่ช่วยในการสร้าง หรือปรับปรุงแก้ไขโครงสร้างแฟ้ม DBF เพราะ Clipper เป็นตัวแปรภาษา จึงต้องอาศัย DBU ช่วยสร้างแฟ้มข้อมูลหรือ แฟ้มดัชนี สำหรับการดำเนินงานในเบื้องต้น
- .. For คือคำสั่งทำซ้ำ ถ้าสั่งให้ for ทำซ้ำ 10 ครั้ง แล้วใช้ครอบ for ที่ทำซ้ำ 10 ครั้ง อีกตัวหนึ่ง จะทำให้เกิดการทำซ้ำถึง 100 ครั้ง โดย for นั้นจะต้องปิดด้วยคำสั่ง next เสมอ

## 2 สรุปกิจกรรม :

นำคอมพิวเตอร์พร้อม projector ไปแสดงการ compile และประมวลผล ไปแสดงให้ดู

นำตัวอย่างโปรแกรมของรุ่นพี่ หรือโปรแกรมที่น่าสนใจไปให้ดูเป็นตัวอย่าง

ทำความเข้าใจลักษณะของหนังสือที่ใช้ประกอบการเรียน

ดูหลักการแปล และเริ่มเขียนโปรแกรม ในโครงสร้างง่าย ๆ จากบทที่ 1 ทั้งบท

ศึกษาการเขียนโปรแกรมเบื้องต้น บทที่ 4 ในหัวข้อ 4.1

มอบหมาย โดยให้แผ่น Clipper compiler 10 ชุดให้นักศึกษาไป install ที่บ้าน

## 2 รายละเอียดกิจกรรม :

- นำโปรแกรม Clipper compiler จำนวน 10 แผ่น แจกจ่ายให้ผู้นำนักศึกษา 10 คน นำไปคัดลอก แล้วส่งต่อไปยังคนต่อไปจนครบทุกคน แต่ให้กระทำหลังจากเรียนเสร็จ เนื่องจากต้องนำไปใช้กับคอมพิวเตอร์ส่วนตัวของที่บ้าน และนักศึกษาสามารถ copy ไว้ใช้ได้ เวลาเครื่องมีปัญหา ดังนั้นต้นฉบับที่ให้ copy จะทวีจำนวนมากขึ้น และให้นักศึกษานำต้นฉบับของอาจารย์กลับมาคืนในห้องเรียน ในวันต่อไปที่มีการเรียนการสอน
- นำตัวอย่างโปรแกรมที่อาจารย์ได้เขียน Present ผ่าน projector ให้ดูว่านักศึกษาจะต้องเขียนโปรแกรมในรูปแบบใดต่อไป ซึ่งโปรแกรมที่นำมาเสนอ ประกอบด้วย Menu แบบ Pulldown menu และ โปรแกรมเช่า และคืนหนังสือ
- อธิบายให้นักศึกษาเข้าใจว่า หนังสือที่ประกอบการสอนประกอบด้วยบทเรียน 6 บท
- อธิบายการ Compile โดยพิมพ์ให้ดูทางคอมพิวเตอร์ออก Projector โดยมีขั้นตอนดังนี้
  1. ต้องแก้ไขใน Autoexec.bat สำหรับครั้งแรก ที่จะทำให้เครื่องนั้น Compile ภาษา Clipper ได้
 

```
Path = c:\c:\clipper5
Include=c:\clipper5
Lib=c:\clipper5
Obj=c:\clipper5
```
  2. Copy โปรแกรมสำหรับ compile ทั้งหมดในห้อง c:\clipper5 ที่ถูกสร้างขึ้น
  3. ให้เขียนโปรแกรมเช่น X.PRГ เก็บในห้อง c:\clipper5 แล้วพิมพ์คำว่า ? 5 เพื่อสั่งให้พิมพ์เลข 5 ทางจอภาพ
  4. เข้าห้อง c:\clipper5 แล้วสั่งแปลโปรแกรมด้วย 2 บรรทัดข้างล่างนี้
 

```
c:\clipper5\clipper x
c:\clipper5\rtlink file x
```
- เริ่มสอนเขียนโปรแกรมในหัวข้อ 4.1 ตามหนังสือหน้า 178

## 2 เขียนแผนผัง :

- หนังสือที่ประกอบการสอนประกอบด้วยบทเรียน 6 บท
  1. สิ่งที่ต้องรู้เกี่ยวกับ Clipper
  2. คำสั่ง (Command) คือ สั่งให้โปรแกรมทำงาน 103 คำสั่ง
  3. ฟังก์ชัน (Function) คือ คำสั่งที่มีการส่งค่าเป็นหลัก 194 ฟังก์ชัน
  4. หลักการพื้นฐานในการเขียนโปรแกรม
  5. กรณีศึกษาระบบงานทะเบียน
  6. กรณีศึกษาระบบงานขาย และสินค้าคงคลัง

## สัปดาห์ที่ 2

### P หัวข้อ 3 ศึกษา และทบทวนการใช้คำสั่ง if ร่วมกับการทำซ้ำด้วยคำสั่ง for

#### 2 สรุปคำบรรยาย :

- .. if คือคำสั่ง เพื่อใช้เลือกเงื่อนไข เป็นหลักการหนึ่งที่สำคัญในเรื่องของ Structure programming
- .. for คือการทำซ้ำ ซึ่งกำหนดจำนวนครั้ง จากจุดเริ่มต้น ถึงจุดสิ้นสุด เป็นเงื่อนไข
- .. while คือการทำซ้ำ แต่จะตรวจสอบเงื่อนไขทุกครั้งที่เราเริ่ม ทำซ้ำกระบวนการใหม่ เพื่อตัดสินที่จะทำต่อหรือไม่

#### 2 สรุปกิจกรรม :

ฟังบรรยายการเขียนโปรแกรมเพื่อทำซ้ำ และเลือกเงื่อนไขเพิ่มเติม 4.2

ฟังบรรยาย ลักษณะฟังก์ชัน และคำสั่งที่สำคัญจากบทที่ 2 และ 3

ทดสอบทำโปรแกรมพีระมิดอย่างง่ายใน ชั้นเรียน

#### 2 รายละเอียดกิจกรรม :

- อธิบายคำสั่ง ? และ ?? หน้า 14 ทำหน้าที่แสดงผล โดยกำหนด row และ col ไม่ได้
- อธิบายคำสั่ง @ .. say หน้า 18 ทำหน้าที่แสดงผล โดยกำหนด row และ col ได้
- อธิบายว่าการทำซ้ำที่นิยม และใช้กันมากมี 2 คำสั่งคือ for และ while  
ตัวอย่างการใช้ for และ while ดูได้จากหน้า 178
- อธิบายฟังก์ชัน mod หน้า 142 ทำหน้าที่หารโดยหาเฉพาะเศษที่เหลือ
- ดูตัวอย่างพีระมิดอย่างง่ายพร้อมฟังคำอธิบาย ตามตัวอย่างในตัวอย่าง 4.1 ถึง 4.10
- ให้นักศึกษาทดลองทำในห้อง พร้อมส่งงานท้ายชั่วโมง โดยให้นักศึกษาเลือกได้ 2 ใน 3  
ให้เขียนโปรแกรมโดยมีโจทย์ต่อไปนี้

121	หรือ	*51	หรือ	551551
232		**40		442440
343		***31		333331
454		****20		224220
565		*****11		115111

## 2 งานมอบหมาย ครั้งที่ 1

1. ให้เขียนโปรแกรมพิมพ์ 1 ถึง 10
2. ให้เขียนโปรแกรมพีรามิดอย่างง่าย อย่างใดก็ได้ 10 โปรแกรม

## 2 เขียนแผ่นใส :

ตัวอย่างการทำซ้ำเพื่อพิมพ์เลข 1 ถึง 10 ด้วย for

```
for I = 1 to 10
```

```
? I
```

```
Next
```

ตัวอย่างการทำซ้ำเพื่อพิมพ์เลข 1 ถึง 10 ด้วย while

```
I = 0
```

```
I = 1
```

ผลลัพธ์

```
while I < 5
```

หรือ

```
while I <= 5
```

```
I ++
```

```
? I
```

```
? I
```

```
I ++
```

```
end
```

```
end
```

ตัวอย่างการใช้ mod

```
? mod(5,2)
```

ได้ผลเป็น 1

```
? mod(100, 30)
```

ได้ผลเป็น 10

## P หัวข้อ 4 ศึกษา และเปรียบเทียบการใช้คำสั่ง while หรือ do while แทนคำสั่ง for

### 2 สรุปคำบรรยาย :

- while คือการทำซ้ำ สามารถใช้คู่กับ end ถ้าใช้ do while จะใช้คู่กับ enddo ซึ่งความแตกต่างของ while ก็คือทำซ้ำไปเรื่อย ๆ จนกระทั่งเงื่อนไขเป็นจริง แต่ for นั้นจะกำหนดการทำซ้ำไว้ชัดเจน
- พีระมิด ของตัวเลข คือโจทย์ที่ใช้ฝึกนักศึกษานักคิด และวางแผน ทำให้นักศึกษารู้จักการควบคุมว่าจะเริ่มอย่างไร ประมวลผลกี่ครั้ง และหยุดเมื่อใด

### 2 สรุปกิจกรรม :

ฟังบรรยายการใช้คำสั่ง ที่แตกต่าง และเหมือนกันของ while และ for

ทดสอบทำโปรแกรมพีรามิด โดยใช้คำสั่งที่ต่างกันของ while และ for

ศึกษาการทำซ้ำที่ซับซ้อนเพิ่มขึ้นโดยใช้ พีรามิดเป็น กรณีศึกษา

### 2 รายละเอียดกิจกรรม :

- อธิบาย while ว่าเป็นการทำซ้ำที่มีการควบคุมโดยอิสระ เพราะต้องกำหนดเงื่อนไขเอง ต่างกับการใช้คำสั่ง for ซึ่งเป็นการกำหนดแบบคงที่ ทำให้เข้าใจง่าย แต่นำไปใช้ได้ไม่บ่อยเท่ากับ while เพราะยืดหยุ่นกว่า
- while หรือ do while นั้นไม่ต่างกัน เพียงแต่ do while เป็นการเขียนในรูปแบบที่เคยใช้กับ dbase III ดังนั้น Clipper จึงปรับคำสั่งนี้ให้สั้นลงโดยเปลี่ยนเป็น while เท่านั้น

- อธิบายการเขียน พีระมิดแบบขีดขวา ตามตัวอย่าง 4.11 ถึง 4.18 ซึ่งเป็นหลักการเขียนตัวเลขในรูปทรงต่าง ๆ ที่สัมพันธ์กัน
- ให้นักศึกษาไปทำการบ้านโดยให้นักศึกษาไปคิดโจทย์ แต่ให้ตัวอย่างโจทย์ที่น่าสนใจของพีระมิด

2 งานมอบหมาย ครั้งที่ 2

1. ให้ทำโปรแกรมพีระมิดที่ซับซ้อนกว่าเดิมอีก 10 โปรแกรม โดยให้นักศึกษาไปคิดโจทย์ขึ้นมา

2 เขียนแผ่นใส :

ตัวอย่างการใช้ while และ for ซึ่งให้ผลที่ต่างกัน

ผลลัพธ์

for I = 1 to 10	หรือ	I = 0	หรือ	I = 1	1
? I		while I < 5		while I <= 5	2
next		I ++		? I	3
		? I		I ++	4
		end		end	5

ตัวอย่างแสดงความแตกต่างของ while และ do while

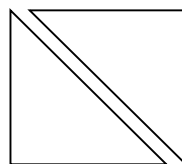
I = 0	และ	I = 0
while I < 5		do while I < 5
I ++		I ++
? I		? I
end		enddo

อธิบายความสัมพันธ์ของพีระมิดโดยใช้ภาพ

พิมพ์รวมดาดจะได้ดังนี้

1\*\*\*\*\*  
12\*\*\*\*\*  
123\*\*\*\*\*  
1234\*\*\*\*\*  
12345\*\*\*\*\*

ซึ่งแยกเป็น 2 ส่วน



1 และ \*\*\*\*\*  
12 \*\*\*\*\*  
123 \*\*\*\*\*  
1234 \*\*\*\*\*  
12345 \*\*\*\*\*

ชี้แจงให้นักศึกษา : เรื่อง พีระมิด เพื่อให้เข้าใจการทำซ้ำด้วย while และ for

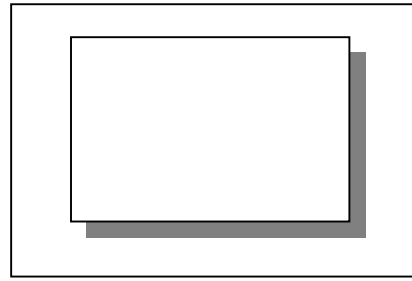
1. 11*****11 22****22 33***33 4444	2. 123*** 234**** 345***** 456*****	3. 15***** 26***** 37***** 48*****	4. *1 **212 ***32123 ****4321234
5. 1 2*2 3***3 4*****4 3***3 2*2 1	6. 101010101010 1 10101010101 101010101 1010101 10101 101 1	7. 111 12221 1233321 123444321 12345554321 1234566654321 1234567776543 21	8. 123** 234*** 345**** 456***** 567***** 678***** 789*****
9. 111111111 2222222 33333 444 5	10. 151 12421 1233321 123424321 12345154321	11. 1 22 3 3 4 4 5 5	12. 123456789 2345678 34567 456 5
13. 987654321 7654321 54321 321 1	14. 12 21 23 32 34 43 45 54 5665	15. 11111 11111 2222 2222 333 333 44 44 5 5	16. 1 1 21 ** 321 *** 4321 **** 54321*****
17. 19191919 28282828 37373737 46464646 55555555	18. 1*1*****9 1**2*****8 3***3***7 4****4**6 5*****5*5	19. 1*****5 2*****4 3*****3 4*****2 5*****1	20. 1*1 22**22 333***333 444***4444 5555*****5555 5
21. *****1*7 *****2**6 *****3***5 ****4****4 ***5*****3	22. *111111111* * 2222222 * * 33333 * * 444 * * 5 *	23. 1*****1 2*****2 3****3 4***4 5**5	24. 1*111111111 2***2222222 3*****33333 4*****444 5*****5
25. ****5***** ***444**** **33333*** *2222222** 111111111*	26. 11111* 2222** 333*** 44**** 5*****	27. 1 * 2 ** 3 *** 4 **** 5 *****	28. 54321*12345 5432***2345 543*****345 54*****45 5*****5
29. 1*3*** 2*4**** 3*5***** 4*6***** 5*7*****	30. *****1 *****21 *****321 ***4321 *54321	31. 11111***** 2222 **** 333 *** 44 ** 5 *	32. 1234567***3 123456****4 12345*****5 1234*****6 123*****7



```

ตัวอย่างการสร้าง shadow
set color to "/w"
cls
set color to "/n"
@ 5,10 clear to 22,70
set color to "w/b"
@ 4,8 clear to 21,68

```



## P หัวข้อ 6 ศึกษาการใช้คำสั่งเพื่อการคำนวณ และการนำข้อมูลจากเพิ่มมาคำนวณ และการใช้ if กำหนดเงื่อนไข

### 2 สรุปคำบรรยาย :

.. เครื่องหมาย \* สำหรับคอมพิวเตอร์หมายถึงการคูณ แต่ถ้าเป็น + กับ - และ / จะเหมือนกับที่ใช้ในเครื่องหมายคณิตศาสตร์ทั่วไป

.. เครื่องหมาย % หมายถึง mod คือการหาเศษที่ได้จากการหาร เช่น  $7 \% 2$  จะได้ผลเป็น 1

### 2 สรุปกิจกรรม :

ศึกษากรณีการคำนวณในแบบต่าง ๆ

ทดสอบการอ่านข้อมูลมาประมวลผล เช่นระบบเงินเดือน ระบบทะเบียนนักศึกษา ระบบบุคลากร

### 2 รายละเอียดกิจกรรม :

- เครื่องหมายที่ตัวอย่างการใช้ในหนังสือหน้า 8 ซึ่งเครื่องหมายที่ใช้มี 6 เครื่องหมาย คือ

\*\* หรือ ^ เป็นการยกกำลัง เช่น  $2^{**}3$  เท่ากับ 8

\* เป็นการคูณ เช่น  $2 * 3$  เท่ากับ 6

/ เป็นการหาร เช่น  $4 / 2$  เท่ากับ 2

% เป็นการหาเศษของการหาร เช่น  $20 \% 3$  เท่ากับ 2

+ เป็นการบวก เช่น  $3 + 2$  เท่ากับ 5

- เป็นการลบ เช่น  $3 - 2$  เท่ากับ 1

- ศึกษาโปรแกรมคำนวณตามตัวอย่าง 4.33 และ 4.34 หน้า 192 และ 193

- ให้นักศึกษาเขียนโปรแกรมคำนวณ โดยเลือก 2 ใน 4 ของโปรแกรมต่อไปนี้ แบบ close book

สำหรับโครงสร้างเพิ่ม และรายละเอียดอื่น ๆ ให้กำหนดขึ้นตามความเหมาะสม

1. คำนวณเกรด 2. คำนวณค่านายหน้า 3. คำนวณภาษี 4. คำนวณรายได้หลังหักวันลา

### 2 งานมอบหมาย ครั้งที่ 4

1. ให้เขียนโปรแกรมประมวลผลมา 3 แบบ ๆ ละโปรแกรม โดยให้คิดโครงสร้างเพิ่มขึ้นเอง

### 2 เขียนแผ่นใส :

ตัวอย่างโปรแกรมคำนวณเกรดนักเรียน

```
use student
```

```
while !eof()
```

```
grade = "F"
```

```
if score >=50 ; grade = "D" ; endif
```

```
if score >=60 ; grade = "C" ; endif
```

```
if score >=70 ; grade = "B" ; endif
```

```
if score >=80 ; grade = "A" ; endif
```

```
? sid, score, grade
```



```

        skip
    end
    ตัวอย่างเลือกเฉพาะพนักงานที่มีสถานภาพเป็น A
    use empl
    l = 1
    while !eof()
        if status = "A"
            ? l, emplid, name, status
            l ++
        endif
    skip
    end
end

```

**P หัวข้อ 7** ศึกษาการเลือกที่จะกระทำในลักษณะเมนู โดยใช้คำสั่ง `accept` รับค่า และใช้ `if` เลือกกระทำ

**2** สรุปคำบรรยาย :

- .. คำสั่ง `accept` เป็นคำสั่งรับค่าจากแป้นพิมพ์อย่างง่าย ๆ เมื่อรับเข้ามาแล้วสามารถใช้คำสั่ง `case` เพื่อทำการเลือกเงื่อนไข ซึ่ง `case` นั้นจะเข้าใจง่ายกว่า `if` ในกรณีที่เงื่อนไขที่ต้องการตรวจสอบมีมาก แต่ถ้าจะใช้ `if` มักใช้กับเงื่อนไขเดียว หรือ 2 เงื่อนไขที่เป็นจริงกับเท็จเท่านั้น แต่บางท่านอาจใช้ `if` ซ้อนกัน จนมีหน้าที่เหมือน `case` ได้
- .. `wait` นั้นจะรับรหัสจากแป้นพิมพ์แล้ว `inkey()` สามารถรับค่ามาเป็นรหัสตั้งแต่ 0 ถึง 255 ได้
- .. `text` เป็นอีกคำสั่งหนึ่งที่ใช้แสดงผลพร้อมได้อย่างง่าย ๆ ซึ่ง ? นั้นก็เป็นการแสดงผลในบรรทัดที่ต้องการ แต่ถ้าใช้ ?? จะเป็นการแสดงค่าโดยไม่สนใจเรื่องการปิดบรรทัด

**2** สรุปกิจกรรม :

ทำความเข้าใจหลักการเขียนโปรแกรมเมนู และหน้าที่  
ทำความเข้าใจการเลือกเงื่อนไขที่แตกต่างกันของ `case` และ `if`

**2** รายละเอียดกิจกรรม :

- ศึกษาเขียนเมนู แสดงตัวเลือกด้วย `text` หรือ ? แล้วรอรับค่าด้วย `accept` หรือ `wait` ตามตัวอย่างหน้า 195
- ศึกษาความแตกต่างของการใช้ `if` และ `case` ตามตัวอย่าง 4.37 ถึง 4.39
- ศึกษาหน้าที่ของ ฟังก์ชัน `inkey()` หน้า 127 ซึ่งรอรับค่าแบบไม่แสดงผลบนจอภาพ
- ศึกษาหน้าที่ของ ฟังก์ชัน `wait()` หน้า 71 ซึ่งรอรับค่าอักษร แล้วส่งเป็นรหัส ASCII

**2** งานมอบหมาย ครั้งที่ 5

1. ให้เขียนโปรแกรมรับตัวเลข แล้วเลือกด้วย `case` กระทำการบางอย่าง ตามสมควร
2. ให้เขียนโปรแกรมรับตัวอักษร แล้วเลือกด้วย `if` กระทำการบางอย่าง ตามสมควร

**2** เขียนแผ่นใส :

ตัวอย่างเมนูด้วย `text` และ `case`

`text`

1. พิมพ์เลข 5
2. พิมพ์คำว่า "YONOK"
3. เลิกการทำงาน

```

endtext
accept to opt
do case
case opt = "1" ; ? 5
case opt = "2" ; ? "YONOK"
case opt = "3" ; exit
endcase

```

ตัวอย่างเมนูด้วย ? และ if

```

while .t.
? "1. พิมพ์เลข 5"
? "2. พิมพ์คำว่า YONOK"
? "3. เลิกการทำงาน"
wait "รอรับค่า"
opt = lastkey() - 48
if opt = 1 ; ? 5 ; endif
if opt = 2 ; ? "YONOK" ; endif
if opt = 3 ; exit ; endif
wait
end

```

## สัปดาห์ที่ 4

**P หัวข้อ 8** ศึกษาการสร้างเมนู โดยใช้ function ของภาษา ด้วย prompt และ achoice

**2** สรุปคำบรรยาย :

- .. get เป็นคำสั่งรับค่าจากเพิ่มพิมพ์ที่นิยมและมีประสิทธิภาพ เพราะรับได้ทั้งตัวเลข และตัวอักษรได้ยาว
  - .. prompt และ achoice เป็นคำสั่งที่ใช้แสดงเมนูให้เลือกซึ่งต่างก็มีคุณสมบัติเด่นของตนเอง ที่ไม่ยิ่งหย่อนไปกว่ากัน
- นักศึกษาจึงจำเป็นต้องเรียนรู้การทำเมนูทั้ง prompt และ achoice

**2** สรุปกิจกรรม :

ศึกษาการเลือกตัวเลือกที่แตกต่างกันของเมนูแบบใช้ get, prompt และ achoice

ทดสอบการเขียนเมนูอย่างง่ายเพื่อเลือกกระทำอย่างใดอย่างหนึ่ง

**2** รายละเอียดกิจกรรม :

- ศึกษาการใช้คำสั่ง get หน้า 16 เพื่อรอรับค่าได้ในหลายหลายรูปแบบ และยืดหยุ่นมาก
  - ศึกษาการสร้างตัวเลือกเป็นเมนูด้วย prompt หน้า 18 เพื่อให้เลือกได้สะดวก
  - ศึกษาการสร้างตัวเลือกเป็นเมนูด้วย achoice หน้า 75 ซึ่งกำหนดตัวเลือกมาได้จาก array
  - ให้นักศึกษาเขียนโปรแกรมเมนูทั้งแบบ prompt และ achoice ในห้องเรียน โดยมีตัวเลือกดังนี้
- เมนูทั้ง 2 โปรแกรมจะต้องทำซ้ำ และเลิกการทำงานต่อเมื่อกด 4
1. พิมพ์สูตรคูณแม่ 2
  2. พิมพ์คำว่า YONOK 100 บรรทัด
  3. พิมพ์ชื่อนักศึกษา
  4. เลิกการทำงาน

## 2 งานมอบหมาย ครั้งที่ 6

1. ให้เขียนโปรแกรมเมนู ทั้งแบบ `get`, `prompt` และ `achoice` อย่างละโปรแกรม โดยมีรูปแบบที่สวยงาม

### .2 เขียนแผ่นใส :

ตัวอย่างโปรแกรมเมนูด้วย `prompt`

`while .t.`

`@ 5,10 prompt "1. พิมพ์เลข 5"`

`@ 6,10 prompt "2. พิมพ์คำว่า YONOK"`

`@ 7,10 prompt "3. เลิกการทำงาน"`

`menu to opt`

`if opt = 1 ; ? 5 ; endif`

`if opt = 2 ; ? "YONOK" ; endif`

`if opt = 3 ; exit ; endif`

`wait`

`end`

ตัวอย่างโปรแกรมเมนูด้วย `achoice`

`choice = { "1. พิมพ์เลข 5", "2. พิมพ์คำว่า YONOK", "3. เลิกการทำงาน" }`

`while .t.`

`opt = achoice (5,5,7,60,choice)`

`do case`

`case opt = 1 ; ? 5`

`case opt = 2 ; ? "YONOK"`

`case opt = 3 ; exit`

`endcase`

`wait`

`end`

## P หัวข้อ 9 ศึกษาการสร้างเมนูหลัก และเมนูย่อย โดยประยุกต์ใช้ `prompt` และ `achoice` ซ้อนกัน

### 2 สรุปคำบรรยาย :

- เมนูอย่างง่าย ๆ นั้นใครก็ทำได้ แต่เมนูที่สมบูรณ์จะต้องวางแผน สร้างระบบเครือข่าย ว่าจะมีเมนูหลัก หรือเมนูย่อยอย่างไร และในแต่ละเมนูย่อยนั้น มีงานอะไรบ้าง ดังนั้นการสร้างเมนูที่สมบูรณ์ปัญหาหลักคือการสร้างตัวเลือก หรืองานจากการเลือกตัวเลือก ไม่ใช่ตัวเมนู
- โปรแกรมย่อยคือโปรแกรมที่ถูกเรียกให้มาทำงาน ในบทนี้จะไม่กล่าวถึง แต่เกริ่นนำให้ทราบว่า โปรแกรมย่อยที่เมนูไปเรียกมาทำงานนั้น สามารถเขียนได้ทั้งในโปรแกรมเมนู หรือเป็นโปรแกรมอิสระข้างนอกก็ได้

## 2 สรุปกิจกรรม :

ศึกษาการเขียนเมนูย่อย ที่ซับซ้อนขึ้น

ทดสอบการเขียนเมนูย่อย โดยเลือกได้ระหว่าง prompt และ achoice

## 2 รายละเอียดกิจกรรม :

- ทบทวนเมนู prompt และ achoice ตามหน้า 18 และ 75
- ศึกษาการทำเมนูซ้อนเมนู จากแผ่นใส
- ให้นักศึกษาทำโปรแกรมเมนูซ้อนเมนู โดยเขียนให้แตกต่างไปจากตัวอย่าง แบบ close book ในห้องเรียน เช่น การเลือกเพิ่ม และเลือกที่จะคำนวณภาษี หรือผลรวม หรือคำนวณหน้า เป็นต้น

## 2 เขียนแผ่นใส :

ตัวอย่างโปรแกรมเมนูด้วย achoice และ prompt ร่วมกัน

```
choice = { "1. พิมพ์เลข","2. พิมพ์อักษร","3. เลิกการทำงาน" }
```

```
while .t.
```

```
    opt = achoice (5,5,7,60,choice)
```

```
    do case
```

```
    case opt = 1
```

```
        @ 5,10 prompt "1. พิมพ์เลข 5"
```

```
        @ 6,10 prompt "2. พิมพ์เลข 6"
```

```
        @ 7,10 prompt "3. เลิกการทำงาน"
```

```
        menu to optx
```

```
        if optx = 1 ; ? "Five" ; endif
```

```
        if optx = 2 ; ? "Six" ; endif
```

```
        if optx = 3 ; exit ; endif
```

```
    case opt = 2
```

```
        @ 5,10 prompt "1. พิมพ์เลข A"
```

```
        @ 6,10 prompt "2. พิมพ์เลข B"
```

```
        @ 7,10 prompt "3. เลิกการทำงาน"
```

```
        menu to optx
```

```
        if optx = 1 ; ? "This is A" ; endif
```

```
        if optx = 2 ; ? "This is B" ; endif
```

```
        if optx = 3 ; exit ; endif
```

```
    case opt = 3 ; exit
```

```
    endcase
```

```
    wait
```

```
end
```

**P หัวข้อ 10** ศึกษา และทบทวน การเขียนโปรแกรมอ่านข้อมูลมาพิมพ์ คำนวณ คำนวณผลรวม การแยกหน้า และแสดงผลบนพื้นสี

**2** สรุปคำบรรยาย :

- .. นักศึกษาเริ่มชินกับการเขียนโปรแกรม ในหัวข้อนี้จะฝึกให้นักศึกษารู้จักการคำนวณอย่างง่าย ๆ การทำงานในส่วนทำซ้ำ การแสดงผลแบบแยกหน้า และการแสดงสี เพื่อให้ผลลัพธ์สวยงาม
- .. สีสี่หลายสี ซึ่งในหลักสูตรนี้จะใช้เลขสีไม่เกิน 16 สี เพราะสามารถแสดงผลให้เห็นความแตกต่างได้ในระบบ DOS
- .. ให้กลับไปทบทวนหัวข้อ 9 และเปิดหนังสือเพื่อทำความเข้าใจตัวอย่างโปรแกรมหน้า 188 ถึง 193

**2** สรุปกิจกรรม :

ทบทวนโปรแกรมอ่านข้อมูลมาพิมพ์ที่สมบูรณ์ยิ่งขึ้น

ทดสอบการเขียนโปรแกรมรายงานแยกหน้า

**2** รายละเอียดกิจกรรม :

- ทบทวนโปรแกรมอ่านข้อมูลมาพิมพ์จาก 4.25 ถึง 4.34 หน้า 188 ถึง 193
- ให้นักศึกษาทำงานในห้องเรียน โดยเขียนโปรแกรมอ่านข้อมูลมาพิมพ์โดยทำรายงานให้สมบูรณ์ทั้งการมีลำดับคำนวณ และเลือกเงื่อนไข โดยใช้โครงสร้างแฟ้มดังนี้ รหัสพนักงาน ยอดขาย เงินเดือน สถานภาพ แผนก โดยเขียนมา 2 โปรแกรม

**2** งานมอบหมาย ครั้งที่ 7

1. ให้เขียนโปรแกรมรายงาน ที่มีลักษณะที่แตกต่างกัน มา 3 แบบ ๆ ละโปรแกรม

## สัปดาห์ที่ 5

**P หัวข้อ 11** ศึกษากรณีตัวอย่าง การอ่านข้อมูลมาแสดง โดยคำนวณคะแนน แล้วแสดงเกรดที่เหมาะสม ด้วย do case

**2** สรุปคำบรรยาย :

- .. นักศึกษาทุกทราบบความหมายของคำว่าผลการเรียน ปกติคะแนนจะเก็บเป็นตัวเลข แต่หลังปฏิบัติการศึกษา ผลการเรียนแสดงออกมาเป็นเกรดเช่น A B C D และ F หมายถึงตก ในบทนี้ต้องการแสดงให้เห็นความสามารถของ do case เพื่อเลือกข้อมูลมาแสดงอย่างเหมาะสม เพราะแฟ้มข้อมูลเก็บคะแนน แต่ผลลัพธ์บนจอภาพจะแสดงเป็นเกรด
- .. นักศึกษาอาจคุ้นเคยกับคำว่า ค่านายหน้า เช่นขายได้มาก ก็ได้เงินมาก ขายได้น้อยก็ได้ค่านายหน้าน้อย ถ้านักศึกษารู้จักประยุกต์โปรแกรมเกรด ไปเขียนโปรแกรมค่านายหน้าได้ ก็นับว่ายอด

**2** สรุปกิจกรรม :

ศึกษาการเลือกเงื่อนไขเช่นการใช้ do case ตัดเกรดนักเรียน และคำนวณที่ซับซ้อน

ทดสอบการเขียนโปรแกรมตัดเกรด หรือหาค่านายหน้า (Commission)

**2** รายละเอียดกิจกรรม :

- ทบทวนการใช้ do case ตัดเกรด โดยศึกษาจากแผ่นใส
- ให้นักศึกษาเขียนโปรแกรมตัดเกรดตามตัวอย่างแบบ close book แต่ให้เพิ่มการเลือกสถานภาพในห้องเรียน และให้เขียนโปรแกรมคำนวณค่านายหน้า โดยได้ยอดที่แตกต่างไปตามยอดขายที่เพิ่มขึ้น (สิ่งที่ไม่ได้กำหนด ให้กำหนดขึ้นเอง)

## 2 งานมอบหมาย ครั้งที่ 8

1. ให้เขียนโปรแกรมที่มีลักษณะเป็นโปรแกรมตัดเกรด จาก คะแนนเก็บ กลางภาค ปลายภาค
2. ให้เขียนโปรแกรมที่มีลักษณะเป็นโปรแกรมคำนวณค่านายหน้า(Commission)

## 2 เขียนแผ่นใส :

ตัวอย่างโปรแกรมตัดเกรด

```
use student
while !eof()
```

```
score = keep + midterm + final
do case
```

```
case score >=80 ; grade = "A" ; A++
case score >=70 ; grade = "B" ; B++
case score >= 60 ; grade = "C" ; C++
case score >=50 ; grade = "D" ; D++
otherwise ; grade = "F" ; F++
```

```
endcase
```

```
? sid, keep, midterm, final, score, grade
```

```
skip
```

```
end
```

```
? "Have A ", A
```

```
? "Have B ", B
```

```
? "Have C ", C
```

```
? "Have D ", D
```

## P หัวข้อ 12 ศึกษากรณีตัวอย่าง การอ่านข้อมูลพนักงานขาย ไปคำนวณค่านายหน้า และหาผลรวมค่านายหน้า

### 2 สรุปคำบรรยาย :

- .. ค่านายหน้าภาษาอังกฤษคือ **commission** ปัจจุบันจะมี ธุรกิจที่ชื่อว่า **work at home** ซึ่งออกมาว่าเป็นธุรกิจที่รวยง่ายเนื่องจากขาย ของบำรุงร่างกายแล้วได้ค่านายหน้าไม่ถึงกึ่งหนึ่งของราคาขาย มีชั้นเหมือน **Amway** จึงมีการคำนวณยอดขาย และได้ผลเป็นค่านายหน้าแบ่งให้ผู้ขาย
- .. **Close book** หมายถึงปิดหนังสือ แล้วทำงาน หรือถ้าเป็นสอบบางวิชาจะ **Open book** ซึ่งหมายถึงให้เปิดหนังสือหรือเอกสารขณะสอบได้

### 2 สรุปกิจกรรม :

ศึกษาการคำนวณในหลาย ๆ รูปแบบทั้งการตรวจสอบเงื่อนไข การทำซ้ำ และการหาผลคำนวณรวม  
ทดสอบการคำนวณหลายรูปแบบ และหลายวิธีการ จากที่ได้ศึกษามา

### 2 รายละเอียดกิจกรรม :

- ทบทวนการอ่านข้อมูลพนักงานขายไปคำนวณค่านายหน้า และหาผลรวมจากแผ่นใส
- ให้นักศึกษาทำงานในห้องเรียนแบบ **close book** เพื่อคำนวณค่านายหน้า รวมทั้งให้มี **shadow** และแยกหน้า

## 2 เขียนแผ่นใส :

ตัวอย่างโปรแกรมคำนวณค่านายหน้า

```
set color to "/w"
cls
set color to "/n"
@ 5,10 clear to 22,70
set color to "w/b"
@ 4,8 clear to 21,68
use commision
l = 1 ; sincome = 0 ; tincome = 0
while !eof()
    do case
        case sale >= 100000 ;    comm = sale * 0.2
        case sale >= 20000 ;    comm = sale * 0.1
        case sale >= 10000 ;    comm = sale * 0.08
        otherwise ;            comm = sale * 0.05
    endcase
    income = sale + salary - (salary * 0.07)
    if status = "A"
        ? l , sid , name , salary , sale , income
        l ++
        sincome = sincome + income
    endif
    tincome = tincome + income
    skip
end
? "Total income = " , tincome
? "Total income for Status is A = " , sincome
```

## P หัวข้อ 13 ศึกษาการเขียนโปรแกรมปรับปรุงเพิ่มข้อมูลโดยสามารถ เพิ่ม ลบ และแก้ไขค่าในแต่ละระเบียน

### 2 สรุปคำบรรยาย :

- .. เพิ่มข้อมูลมีคำสั่งสำคัญคือ **append blank** เป็นคำสั่งเพิ่มระเบียนเปล่า จากนั้นจะทำการ **replace** ค่าที่มี อาจนำมาจากตัวแปรที่รับจากแป้นพิมพ์ก็ได้
- .. ลบ ภาษาอังกฤษ คือ **delete** ซึ่งเป็นคำสั่งที่ **clipper** ใช้ลบระเบียน แต่คำสั่งนี้จะเป็นการทำเครื่องหมายเท่านั้น ถ้าต้องการให้ ระเบียนที่ถูกลบ หายไปแบบเรียกคืนไม่ได้ต้องใช้คำสั่ง **pack** ในทางกลับกัน ถ้าเรียกคืนก็ใช้คำสั่ง **recall**

- การแก้ไข จะต้องให้ตัวชี้ไปที่ระเบียนนั้น แล้วใช้คำสั่ง **replace** เขียนทับลงไปใ้ระเบียนเดิม
- **update()** เป็นฟังก์ชันใช้ตรวจว่าในขณะที่รับค่าเข้ามาใน **get** มีการเปลี่ยนค่าเดิมหรือไม่ ถ้าไม่เปลี่ยนค่าจาก **function update()** จะมีค่าเป็นเท็จ ถ้าเปลี่ยนก็จะมีค่าเป็นจริง

## 2 สรุปกิจกรรม :

ทำความเข้าใจ การปรับปรุงข้อมูลอย่างง่าย  
ทดสอบการเพิ่ม ลบ แก้ไข อย่างง่าย

## 2 รายละเอียดกิจกรรม :

- ศึกษาการเพิ่ม ลบ และแก้ไขข้อมูลจากหน้า 222 และ 223
- ศึกษาฟังก์ชัน **updated()** จากหน้า 174
- ให้นักศึกษาได้เขียนโปรแกรม เพิ่ม ลบ และแก้ไข แต่ให้ทำซ้ำได้ แบบ **open book** ในห้องเรียน

## 2 งานมอบหมาย ครั้งที่ 9

1. ให้เขียนโปรแกรม เพิ่ม ลบ แก้ไข อย่างง่าย จากแฟ้มที่นักศึกษาคุ้นเคย

## สัปดาห์ที่ 6

### P หัวข้อ 14 ศึกษาการเขียนโปรแกรมปรับปรุงแฟ้มข้อมูล สามารถตรวจสอบความถูกต้องก่อน เพิ่ม ลบ และแก้ไข

## 2 สรุปคำบรรยาย :

- การเพิ่ม ลบ แก้ไขที่ดีต้องมีการถามย้ำก่อน ดังนั้นทุกครั้งที่จะกระทำการต้องมีคำสั่งขึ้นมาถามว่า ท่านแน่ใจหรือไม่ ซึ่ง **alert** เป็น **pop up question** อย่างง่าย ๆ เพื่อถาม **yes no** หรือ **cancel** หรืออะไรที่ท่านจะระบุไปเพื่อตรวจสอบ
- **pop up question** คือคำถามที่ขึ้นมากลายจอภาพกะทันหัน เมื่อหายไปจะไม่ทำให้ผลบนจอภาพเดิมเสียไป ปกติจะใช้ฟังก์ชัน **alert** เพราะ **clipper** เตรียมมาให้ใช้งานได้ง่าย แต่มีข้อจำกัดหลายอย่าง บางท่านอาจพัฒนาขึ้นใช้เอง

## 2 สรุปกิจกรรม :

ทำความเข้าใจ หลักการตรวจสอบข้อมูล การถามซ้ำ และการจำลอง **buffer** ในการปรับปรุงข้อมูล  
ทดสอบการปรับปรุงข้อมูล โดยตรวจสอบก่อน ปรับปรุงจริง

## 2 รายละเอียดกิจกรรม :

- ศึกษาฟังก์ชัน **Alert** เพื่อสอบถามเฉพาะจุดได้อย่างมีประสิทธิภาพ หน้า 82
- ศึกษาโปรแกรมปรับปรุงข้อมูลที่มีการถามย้ำก่อนปรับปรุงจริง จากแผ่นใส
- ให้นักศึกษาปรับปรุงโปรแกรมให้สมบูรณ์ และใช้งานกับแฟ้มที่นักศึกษาคิดขึ้นมา ส่งในชั้นเรียนแบบ **open book** และให้เขียนทั้งโปรแกรม เพิ่ม ลบ และแก้ไขข้อมูล

## 2 เขียนแผ่นใส :

ตัวอย่างโปรแกรมเพิ่มข้อมูล

use empl

while .t.

@ 5,10 say "Add id " get \_emplid

@ 6,10 say "Add name " get \_name

read



```

asksure = alert("Are you sure?",{ "yes","no" }
if asksure = 1
    append blank
    replace emplid with _emplid, name with _name
endif
doagain = alert("Add again ?",{ "yes","no" }
if doagain != 1 ; exit ; endif
end

```

## P หัวข้อ 15 ศึกษาการสร้าง เรียกใช้ และข้อแตกต่างของ procedure และ function

### 2 สรุปคำบรรยาย :

- .. procedure คือโปรแกรมย่อยที่เน้นการทำงาน เพื่อให้ได้ผลอย่างหนึ่ง เช่นทำหน้าที่เพิ่ม ลบ หรือแก้ไขข้อมูลเป็นต้น หรืออาจทำหน้าที่แสดงรายงาน รับค่าจากแป้นพิมพ์ โดยปกติจะใช้คำสั่ง do ในการเรียก procedure มาทำงาน
- .. function คือ โปรแกรมย่อยที่เน้นการส่งค่า เช่น  $x = aa()$  โดย aa เป็นฟังก์ชันที่ไปหาค่าอะไรมา แล้วส่งค่าให้ X บางท่านอาจเขียนฟังก์ชันให้ทำงานเหมือน procedure ก็ได้ แต่จะเขียน procedure ให้ทำงานเหมือน ฟังก์ชันไม่ได้ ซึ่งปกติฟังก์ชันจะเหมือนตัวแปรตัวหนึ่ง บางครั้งอาจเรียกใช้โดยเขียนเฉพาะ function ก็ได้ เช่น aa เท่านั้น ไม่เหมือน procedure ที่ต้องมีคำว่า do เช่น do bb

### 2 สรุปกิจกรรม :

ศึกษาความแตกต่าง และหน้าที่ของ procedure และ function  
ทดสอบการใช้งาน procedure และ function ตามหน้าที่ที่เหมาะสม

### 2 รายละเอียดกิจกรรม :

- ศึกษาการเขียน procedure จากตัวอย่างหน้า 147
- ศึกษาการเขียน function gradechr จากตัวอย่างหน้า 243 , 266
- ศึกษาความแตกต่างของ function และ procedure บนแผ่นใส

### 2 งานมอบหมาย ครั้งที่ 10

1. ให้เขียนโปรแกรมปรับปรุงข้อมูลโดยใช้งานร่วมกับ procedure และ function ได้.

### 2 เขียนแผ่นใส :

ตัวอย่างการเขียน procedure และใช้งาน

```

do x          หรือ      do y
procedure x          procedure y
? "12345"          for l = 1 to 10 ; ? l ; next
return          return

```

ตัวอย่างการเขียน function และใช้งาน

```

? aa(5)          หรือ      ? xx(10)
function aa(xx)          function xx(zzz)
ZZ = XX + XX          ZZZ = ZZZ * 2
return (zz)          return (zzz)

```

ตัวอย่างการเขียน **procedure** แบบส่งค่า

do x with 5	หรือ	do yy with 10
procedure x		procedure yy
param y		param xxx
? "===" , y * y, "==="		a = xxx + 10
return		? a
		return

ตัวอย่างการเขียน **procedure** แบบส่งค่า

do x	หรือ	do xx	หรือ	do xxx
do y		do yy		? aa(5)
procedure x		procedure xx		function aa(y)
? 5		do zz		z = y + 5
do z		do yy		return (z)
return		return		procedure xxx
procedure y		procedure yy		do yyy
? 20		? 10		? 55
return		return		return
procedure z		procedure zz		procedure yyy
? 10		? 100		? 44
return		return		return

**P หัวข้อ 16** ศึกษาเทคนิคการเขียนโปรแกรม เพื่อสร้างข้อสอบ ให้ผู้ใช้ได้ทดสอบทำด้วยตนเอง

**2** สรุปคำบรรยาย :

- .. ข้อสอบก็คือการแสดงคำถาม และตัวเลือก เมื่อผู้ใช้เลือกตัวใดตัวหนึ่ง ก็จะแสดงผลลัพธ์ว่า ถูกหรือผิดอย่างไร
- .. **Midterm project** มีก็เพื่อฝึกให้นักศึกษาได้รู้ถึงการเขียนโปรแกรมเป็นระบบ เป็นขั้นตอน โดยกำหนดโจทย์อย่างกว้าง ๆ ให้ทำผังแผนไส ซึ่งมีรายละเอียดนอกเหนือจากการทำโปรแกรมข้อสอบเช่น การทำรายงานแบบต่าง ๆ แนวการเก็บข้อมูล และการเรียกใช้จากแฟ้มที่มีการ **link** กันอยู่

**2** สรุปกิจกรรม :

แลกเปลี่ยนประสบการณ์การทำข้อสอบ **online** จากที่เคยพบเห็น  
 ศึกษาแนวคิดการทำโปรแกรมข้อสอบ เพื่อสร้าง **midterm project**  
 สิ่งงาน **midterm project** เกี่ยวกับการเขียนโปรแกรมข้อสอบ

**2** รายละเอียดกิจกรรม :

- ให้นักศึกษาแสดงความคิดเห็นเมื่อได้ยินคำว่าข้อสอบ **online** โดยให้ส่งตัวแทนมาเล่าให้ฟัง
- เล่าหลักการของโปรแกรมข้อสอบให้ฟังคือ
  1. อ่านข้อมูลจากแฟ้มข้อสอบไปแสดงผลบนจอภาพ
  2. ให้ผู้ทำข้อสอบเลือกคำตอบ ตามโจทย์บนจอภาพ
  3. นำคำตอบที่ถูกเลือก และที่ถูกไปเปรียบเทียบกัน หากถูกให้แสดงคำว่าถูก หากผิดให้แสดงคำว่าผิด
  4. เริ่มอ่านข้อสอบข้อต่อไปมาให้ทำจนกว่าจะหมดข้อสอบ หรือหมดเงื่อนไข

- ศึกษาตัวอย่างโปรแกรมข้อสอบ จากหน้า 216 และ 217
- โครงสร้างข้อสอบให้นักศึกษาฝึกทำอย่างง่ายคือ  
เลขข้อสอบ      คำถาม    ตัวเลือก 1 ถึง 4    เลขคำตอบที่ถูกต้อง    รวมเป็น 7 เขตข้อมูล ต่อระเบียน
- ให้นักศึกษาไปเตรียมหาหนังสือข้อสอบเพื่อเตรียมมาป้อนข้อมูลข้อสอบให้ได้ 270 ข้อ

**2 งานมอบหมาย ครั้งที่ 11**

1. ให้สร้างฐานข้อมูลข้อสอบ และป้อนข้อมูลตัวอย่างสัก 10 ระเบียน และพิมพ์ให้ดู
2. ให้เขียนโปรแกรมให้ทำข้อสอบอย่างง่าย

**สั่งงาน midterm project ครั้งที่ 1 :** ให้นักศึกษาเริ่มพิมพ์ข้อมูล และเข้าใจสิ่งที่จะทำทั้งหมดสำหรับงานนี้

**2 เขียนแผนผัง :**

รูปแบบของโปรแกรมข้อสอบพื้นฐาน

เลขข้อ	คำถาม	เช่น	ข้อ 5	ดาวบริวารของโลกคือข้อใด
ตัวเลือก 1				1. ดาวพุธ
ตัวเลือก 2				2. ดาวศุกร์
ตัวเลือก 3				3. ดวงจันทร์
ตัวเลือก 4				4. ดวงอังคาร

ท่านเลือกข้อใด

ท่านเลือกข้อ 3

ผลของการเลือกคือ ถูก หรือ ผิด

ผลของการเลือกคือ ถูก

โจทย์ระบบข้อสอบสำหรับ Midterm project ซึ่งแสดงไว้ในรูปของเมนู

1. ข้อสอบ	2. ปรับปรุง	3.ระบบ
1.1 ข้อสอบทั้งหมดตามลำดับ 90 ข้อ ชุด 1	2.1 เพิ่ม ...	3.1 อธิบายแหล่งข้อมูลที่ได้มา
1.2 สุ่มจากข้อสอบทั้งหมด 30 ข้อ ชุด 1	2.2 ลบ ...	3.2 อธิบายลักษณะโปรแกรม
1.3 นำข้อสอบมาให้ทำ 90 ข้อ ชุด 2	2.3 แก้ไข ...	3.3 เล่าประวัติผู้เขียน
1.4 สุ่มมาให้ทำ 30 ข้อ ชุด 2	2.4 แสดง ...	3.4 แสดงประวัติผู้ที่ได้ทำข้อสอบ
1.5 นำข้อสอบมาให้ทำ 90 ข้อ ชุด 3	2.5 ออก	3.5 เลิกการทำงาน
1.6 สุ่มมาให้ทำ 30 ข้อ ชุด 3		
1.7 นำข้อสอบทั้งหมด 270 ข้อ		
1.8 สุ่มมาให้ทำ 270 ข้อ		
1.9 ออก		

**สัปดาห์ที่ 7**

**P หัวข้อ 17** ศึกษาเทคนิคการเขียนโปรแกรมสุ่มข้อสอบไม่ให้ซ้ำกัน มาให้ผู้ใช้ได้ทำ

**2 สรุปคำบรรยาย :**

- .. การสุ่มไม่ซ้ำจะใช้ เทคนิคการเก็บข้อมูลลงอาเรย์ เมื่อสุ่มมาได้ 1 ค่าก็จะไปตรวจว่ามีอยู่ในอาเรย์แล้วหรือไม่ ถ้าไม่ก็เก็บลงอาเรย์ แต่ถ้ามีแล้วต้องไปทำการสุ่งใหม่
- .. Reccount ( ) เป็นฟังก์ชันที่ใช้นับจำนวนระเบียนในแฟ้มว่ามีเท่าใด
- .. Array ( ) เป็นฟังก์ชันที่ใช้ประกาศว่า array สำหรับเตรียมรับข้อมูลอยู่เท่าใด เหมาะกับการจองค่าในหน่วยความจำ
- .. Ascan ( ) เป็นคำสั่งที่หลายภาษาไม่มี เพราะใช้ตรวจว่าค่าที่มีนั้น มีอยู่ในอาเรย์หรือไม่ ถ้าเป็นภาษาอื่น จะต้องใช้การทำซ้ำกับข้อมูลในอาเรย์ แล้วเปรียบเทียบทีละตัว จนกว่าจะหมด หรือพบ จึงแน่ใจว่ามีหรือไม่มีในอาเรย์
- .. Go เป็นคำสั่งที่ไปยังเลขระเบียนที่ต้องการ เพราะในข้อสอบแบบสุ่ม จะเลือกกระเบียนมาแสดง ด้วยการไปยังระเบียนที่ต้องการด้วยคำสั่ง go นี้เอง

## 2 รูปกิจกรรม :

ทำความเข้าใจวิธีการสุ่มตัวเลขไม่ให้ซ้ำกัน

ศึกษาการเลือกข้อสอบ จากเลขที่สุ่มมาให้ผู้ใช้ได้ฝึกทำข้อสอบ

## 2 รายละเอียดกิจกรรม :

- ศึกษาหลักการเขียนโปรแกรมสุ่มตัวเลขไม่ให้ซ้ำจากตัวอย่าง 4.59 ถึง 4.60 หน้า 217 ถึง 218
- หลักการของโปรแกรมสุ่มข้อสอบเป็นดังนี้
  1. รับจำนวนข้อสอบทั้งหมดในแฟ้ม
  2. เริ่มสุ่มทีละครั้ง ให้ได้เลขที่ไม่เกินเลขข้อสอบ
  3. ตรวจสอบที่สุ่มมาซ้ำกับที่เคยสุ่มไว้แล้วหรือไม่
  4. กลับไปทำข้อ 2 เพื่อสุ่มเลขใหม่ให้ได้จำนวนเท่าที่ต้องการ
  5. ไปเลือกข้อสอบ โดยใช้คำสั่ง GO ไปยังเลขระเบียบที่ต้องการ
  6. เข้ากระบวนการข้อสอบตามปกติ
- อธิบายฟังก์ชัน reccount() หน้า 154
- อธิบายฟังก์ชัน array() หน้า 84
- อธิบายฟังก์ชัน second() หน้า 161
- อธิบายฟังก์ชัน ascan() หน้า 86
- อธิบายคำสั่ง go หน้า 31

## 2 เขียนแผนผัง :

ตัวอย่างโปรแกรมสุ่มตัวเลขไม่ให้ซ้ำ

```
tot = 50
```

```
arnum = array(50)
```

```
for I = 1 to 50
```

```
    rnd = mod(second() * 100,50) + 1           // สุ่มเลขที่ไม่เกินเลข 50
```

```
    if ascan(arnum,rnd) = 0                   // เช็คซ้ำหรือไม่ซ้ำ
```

```
        arnum[I] = rnd
```

```
    else
```

```
        I --
```

```
    endif
```

```
next
```

```
for I = 1 to 50
```

```
    ? arnum[I]
```

```
next
```

## P หัวข้อ 18 ศึกษาเทคนิคการเขียนโปรแกรมข้อสอบ โดยสามารถเฉลยผลการทำงานข้อสอบ อย่างละเอียด พร้อมเก็บประวัติ

### 2 สรุปคำบรรยาย :

- .. การจะเลือกข้อมูล สามารถทำได้มีหลายคำสั่งที่นิยมใช้กัน เช่น **locate** หรือบางท่านอาจใช้ **set filter** โดย **set filter** นี้นิยมใช้เลือกข้อมูลเป็นกลุ่มใหญ่ เช่นเลือกข้อมูลที่เป็นเพศชาย เมื่อใช้คำสั่ง **set filter** แล้วควรตามด้วยคำสั่ง **go top** เพราะจะเกิดข้อผิดพลาดขึ้นหากไม่ใช้คำสั่งนี้
- .. **locate** ใช้ค้นหาข้อมูลตามค่าที่ต้องการ เช่นหาคนชื่อ บุรินทร์ แต่สามารถประยุกต์ให้ใช้งานได้เหมือน **set filter** โดยใช้ **locate rest** เข้าช่วย หรืออาจใช้คำสั่ง **continue** ก็ได้เช่นกัน
- .. การเก็บประวัติการทำงานข้อสอบ หมายถึงการเก็บชื่อ และคะแนนของผู้ทำ เมื่อทำเสร็จแล้ว หรืออาจจะระบุเงื่อนไขเป็นอย่างอื่น ตามแต่ความต้องการของผู้ใช้ และนักพัฒนา

### 2 สรุปกิจกรรม :

ศึกษาการ สร้างข้อสอบให้สามารถเฉลย พร้อมเก็บประวัติได้  
ทดสอบการเขียนโปรแกรมข้อสอบแบบสุ่ม อย่างง่าย

### 2 รายละเอียดกิจกรรม :

- ศึกษาโครงสร้างแฟ้มข้อสอบ 3 ชุดจากแผ่นใน
- ศึกษาการเขียนโปรแกรมแล้วใช้เฉลยตามแบบชุด 2
- ศึกษาการเชื่อม 2 แฟ้ม หน้า 208 หัวข้อ 4.6
- อธิบายคำสั่ง **locate** หน้า 34
- อธิบายคำสั่ง **set filter** หน้า 55

### 2 งานมอบหมาย ครั้งที่ 12

1. ให้เขียนโปรแกรมข้อสอบ ที่ปรับปรุงให้สมบูรณ์เท่าที่ท่านทำได้

**ส่งงาน midterm project ครั้งที่ 2 :** ให้กลับไปสร้างแฟ้ม ตามโครงสร้างในแผ่นใส และป้อนข้อมูลให้สัมพันธ์กัน และให้เขียนโปรแกรมต่าง ๆ ตามโจทย์ที่ต้องการ

### 2 เขียนแผ่นใส :

โครงสร้างแฟ้มข้อสอบชุด 1 แบบธรรมดา

แฟ้มที่ 1.1	ลำดับ	คำถาม	ตัวเลือก 1 ถึง 4	คำตอบ
เช่น	1	2+1=?	3 4 5 6	1
	2	ant แปลว่าอะไร	แมว หมา มด มอด	3

โครงสร้างแฟ้มข้อสอบชุด 2 แบบแยกคำตอบ และเฉลยได้ 2 บรรทัด

แฟ้มที่ 2.1	ลำดับ	คำถาม	ตัวเลือก 1 ถึง 4
เช่น	1	2+1=?	3 4 5 6
	2	ant แปลว่าอะไร	แมว หมา มด มอด

แฟ้มที่ 2.2

เช่น	ลำดับ	คำตอบ
	1	1
	2	3

แฟ้มที่ 2.3

เช่น	ลำดับ	เฉลย 1	เฉลย 2
	1	เพราะเลขบวกกันธรรมดา	ทำให้ 2 + 1 ได้ 3 ข้อที่ถูกจึงเป็น 1

	2	เพราะเปิด dictionary แล้วพบว่า	คำว่า ant แปลว่ามด คำตอบจึงเป็น 3
โครงสร้างเพิ่มข้อสอบชุด 3 แบบมีเฉลยได้มาก			
แฟ้มที่ 3.1	ลำดับ	คำถาม	ตัวเลือก 1 ถึง 4
เช่น	1	2+1=?	3 4 5 6
	2	ant แปลว่าอะไร	แมว หมา มด มอด
แฟ้มที่ 3.2	ลำดับ	คำตอบ	
เช่น	1	1	
	2	3	
แฟ้มที่ 3.3	ลำดับ	ลำดับเฉลย	เฉลย
	1	1	เพราะเลขบวกกันธรรมดา
	1	2	ทำให้ 2 + 1 ได้ 3 ข้อที่ถูกจึงเป็น 1
	2	1	เพราะเปิด dictionary แล้วพบว่า
	2	2	คำว่า ant แปลว่ามด คำตอบจึงเป็น 3

โครงสร้างเพิ่มเก็บประวัติ

แฟ้ม 4.1      วันที่      เวลา      ตัวเลือก      ชื่อ      จำนวนทั้งหมด      จำนวนที่ถูก

ตัวอย่างโปรแกรมข้อสอบเฉลย ตามโครงสร้างแฟ้ม ชุดที่ 2

```

use quiz
while !eof()
  cls
  ? qid,ques
  ? choice1 ; ? choice2 ; ? choice3 ; ? choice4
  opt = 0
  @ 7,20 say "Your choosing is " get opt ; read
  if opt = ans
    @ 8,20 say "Right - Explain : " + exp1 + exp2
  else
    @ 8,20 say "wrong - Explain : " + exp1 + exp2
  endif
  skip
  wait
end

```

ตัวอย่างโปรแกรมเก็บประวัติ

```

_qname = space(40)
@ 5,20 say "what is your name : " get _qname
read
_qdate = date()
_qtime = time ()
_choice = 2                    // เพื่อป้องกันข้อสอบชุดไหนจากตัวเลือกไหน เช่น 2
l = 0
while lastkey() != 27
  l ++
  ? qid, ques

```

```

? choice1 ; ? choice2 ; ? choice3 ; ? choice4
opt = 0
@ 7,20 say "Your choosing is " get opt ; read
if opt = ans ; ok + + ; endif
skip
wait

end
use hist
append blank
replace qdate with _qdate, qtime with _qtime, choice with _choice, qname with _qname ;
allq with l, score with ok

```

**P หัวข้อ 19** ศึกษาการใช้คำสั่ง `save screen` เพื่อควบคุมให้เกิดการแสดงผลร่วมกับ `menu` อย่างง่าย

**2** สรุปคำบรรยาย :

- .. การจัดเก็บจอภาพไว้ชั่วคราว มักใช้คำสั่ง `save screen` เมื่อไปทำงานอย่างใดอย่างหนึ่ง หรือเขียนจอไปแล้วสามารถจำลองที่ทำการจัดเก็บไว้ กลับมาเขียนอีกครั้งด้วย `restore screen`
- .. `Menu` มักแสดงผลเป็นตัวเลือก อาจใช้ `save screen` ทำให้การแสดงผลเมนูมีระเบียบ ให้ตรงกับความต้องการได้โดยง่าย เช่นการเลือกตัวเลือกเพื่อไปทำงานใด ๆ จะทำให้จอภาพเปลี่ยนไป เมื่อเลิกงานแล้ว ควรกลับมาที่เมนูเดิม การจะใช้โปรแกรมเขียนจอใหม่ทั้งหมด อาจไม่ใช่วิธีที่ดีที่สุด จึงใช้ `restore screen` เพื่อนำจอภาพเดิมกลับมา
- .. `Chr` คือฟังก์ชันที่ใช้ส่งค่าตัวอักษร ส่วน `asc` คือฟังก์ชันที่ใช้ส่งเลข `ASCII` เช่น `asc("A")` จะได้ 65 เป็นต้น

**2** สรุปกิจกรรม :

ศึกษาการแต่งจอภาพ ให้ใช้งานร่วมกับโปรแกรมเมนู

**2** รายละเอียดกิจกรรม :

- ศึกษาการแต่งจอภาพอย่างง่าย หน้า 193 และ 194
- อธิบายคำสั่ง `save screen` หน้า 43 ซึ่งจัดเก็บหน้าจอภาพไว้
- อธิบายคำสั่ง `restore screen` หน้า 40 เรียกหน้าจอภาพ มาแสดงผลใหม่
- อธิบายฟังก์ชัน `inkey ()` หน้า 127
- อธิบายฟังก์ชัน `lastkey ()` หน้า 131
- อธิบายฟังก์ชัน `chr ()` หน้า 91 ซึ่งตรงข้ามกับฟังก์ชัน `asc ()` หน้า 85
- อธิบายคำสั่ง `keyboard` หน้า 33

**2** เขียนแผ่นใส :

ตัวอย่างโปรแกรมตรวจแป้นพิมพ์อย่างง่าย

```

keyboard chr(0)
inkey()
while lastkey() = 0
    ? time()

```

```

inkey(1)
end
ตัวอย่างลูกศรวิ่งไป และกลับ
keyboard chr(0)
inkey()
i = 10
while lastkey() = 0
  if i <= 60
    @ 10,i say " =>"
  else
    @ 10,61 - mod(i,60) say "<="
    if i > 110; i = 9; endif
  endif
  i++
inkey(0.1)
end

```

## สัปดาห์ที่ 8

**P หัวข้อ 20** ศึกษาการเขียนโปรแกรม **pulldown** เมนู โดยสร้างระบบ **interface** ขึ้นเอง อย่างง่าย

**2** สรุปคำบรรยาย :

- .. Pulldown menu สามารถสร้างได้ง่าย ๆ ด้วย **prompt** หรือ **achoice** แต่จะให้ทำงานตามต้องการนั้นไม่ได้ เพราะผมต้องการให้นำพิกัดเดินได้ และมีระบบ **screen saver** แบบที่เขียนด้วย **clipper** จึงพัฒนาโปรแกรมนี้ขึ้นมาจากแนวคิดที่มีอยู่ในภาษา **pascal**
- .. การเขียนเมนูที่ซับซ้อน จำเป็นต้องใช้ **procedure** เพราะจะทำให้พัฒนาโปรแกรมได้ง่ายขึ้น เพื่อเรียกงานต่าง ๆ ขึ้นมา และจะเห็นชัดเมื่อสร้าง **procedure** ที่อาจถูกเรียกใช้หลายครั้ง ทำให้ประหยัดเวลาเขียนซ้ำ ๆ กัน
- .. **set epoch** เป็นคำสั่งที่ใช้แก้ปัญหา **Y2K** ได้ เพราะถ้าไม่ทำการ **set epoch** เวลาแสดงปี **2001** จะเข้าใจว่าเป็นปี **1901** ได้ จึงใช้คำสั่งนี้ในทุกโปรแกรมที่เกี่ยวข้องกับการใช้วันที่

**2** สรุปกิจกรรม :

ศึกษาการตรวจสอบแป้นพิมพ์จาก **inkey()** และเลือกแสดงผลตามต้องการ

**2** รายละเอียดกิจกรรม :

- ศึกษาโปรแกรมเมนูจากตัวอย่าง 4.40 ถึง 4.42 หน้า 196 ถึง 198
- ศึกษาหน้าที่ของแต่ละ **procedure** ตามแผ่นใน
- อธิบายคำสั่ง **set color** หน้า 46 ซึ่งมีค่าของสีในหน้า 47
- อธิบายคำสั่ง **set cursor** หน้า 48
- อธิบายคำสั่ง **set epoch** หน้า 53



## 2 งานมอบหมาย ครั้งที่ 13

1. ให้นำโปรแกรมเมนูจากตัวอย่างไปเขียน และแปลให้ใช้งานได้

### 2 เขียนแผนใน:

หน้าที่ของโปรแกรม **pull-down** มิติเดียว (ต.ย. 4.42)

1. กำหนดสี และ cursor
2. กำหนดค่าให้ array เพื่อเป็นตัวเลือก
3. สั่งพิมพ์เมนูบนจอภาพ
4. เข้าสู่การวนซ้ำเพื่อทำงาน
5. พิมพ์เวลา
6. หยุดตรวจแป้นพิมพ์ทุก 0.2 วินาที
7. ตรวจ 3 แป้นแล้วทำตามหน้าที่ คือแป้นลง ขึ้น และ enter
8. สั่งพิมพ์เมนูบนจอภาพใหม่
9. สั่ง clear ค่า inkey หรือแป้นพิมพ์

สำหรับ ส่วนพิมพ์เมนูจะใช้วิธีการดังนี้

1. พิมพ์ตัวเลือกทั้งหมด ไม่ว่าจะเลือก หรือไม่ถูกเลือก
2. สั่งพิมพ์เฉพาะตัวเลือกที่ active อยู่ แต่เปลี่ยนสีที่จะพิมพ์

ตัวอย่างเมนู

1. พิมพ์ 1 ถึง 10
2. พิมพ์สูตรคูณแม่ 2
3. เลิกการทำงาน หรือปุ่ม ESC

## P หัวข้อ 21 ศึกษาการเขียนโปรแกรม pull-down เมนู และการปรับแต่งโปรแกรมจากตัวอย่าง ตรงกับความต้องการ

### 2 สรุปคำบรรยาย:

- .. ให้นักศึกษาทบทวนการเขียนโปรแกรมเมนูจากครั้งที่แล้ว และพัฒนาโปรแกรมให้ได้ตามขั้นตอนต่อไปนี้ คือ เตรียมค่าให้ array เพื่อจะนำไปใช้แสดงผลเป็นเมนูต่อไป, เริ่มกำหนดค่าเริ่มต้นต่าง ๆ ก่อนเข้าส่วนทำซ้ำ, พิมพ์เมนูครั้งแรก ก่อนเข้าส่วนทำซ้ำ, พิมพ์เวลา และหยุด 0.2 วินาที เพื่อรอรับค่าจากแป้นพิมพ์, ตรวจว่ามีกรกดปุ่ม esc ขวา ซ้าย ลง ขึ้น หรือ enter หรือไม่, ถ้ามีจะกระทำการเปลี่ยนค่า opt ตามความเหมาะสม, ถ้ามีการกด enter จะไปทำตัวเลือกที่ต้องการ, ตรวจว่าอยู่ในเมนูย่อยที่ถูกต้องหรือไม่ และปรับค่าสูงสุดต่ำสุดใหม่, เริ่มพิมพ์เมนูใหม่, ทำการ clear ค่าของ keyboard, กลับไปเริ่มกระบวนการทำซ้ำใหม่
- .. นักศึกษาต้องไม่ลืมการทำ midterm project ซึ่งควรมีสิ่งต่อไปนี้ ปกนอก ปกใน คำนำ สารบัญ สรุป ประเมินการ สอน, ปกต้องมีรูป email และ homepage ที่เก็บข้อมูลของ project ไว้แล้ว, มีเลขหน้า อธิบายโครงสร้างแฟ้ม และ อธิบายโปรแกรม

## 2 สรุปกิจกรรม :

ศึกษาการเขียนโปรแกรมเมนู โดยศึกษาส่วนต่าง ๆ เพื่อสามารถปรับปรุงไปใช้ได้

## 2 รายละเอียดกิจกรรม :

- ศึกษาโปรแกรมเมนูจากตัวอย่าง 4.43 ถึง 4.45
- ทำความเข้าใจหน้าที่ของส่วนต่าง ๆ ในโปรแกรม 4.45 หน้า 204 ตามแผ่นใส

## 2 เขียนแผ่นใส :

หน้าที่ของโปรแกรมตัวอย่าง 4.45

1. เตรียมค่าให้ **array** เพื่อจะนำไปใช้แสดงผลเป็นเมนูต่อไป
2. เริ่มกำหนดค่าเริ่มต้นต่าง ๆ ก่อนเข้าส่วนทำซ้ำ
3. พิมพ์เมนูครั้งแรกก่อนเข้าส่วนทำซ้ำ
4. พิมพ์เวลา และหยุด 0.2 วินาที เพื่อรอรับค่าจากแป้นพิมพ์
5. ตรวจสอบว่าการกดปุ่ม **esc** ขวา ซ้าย ลง ขึ้น หรือ **enter** หรือไม่  
ถ้ามีจะกระทำการเปลี่ยนค่า **opt** ตามความเหมาะสม  
ถ้ามีการกด **enter** จะไปทำตัวเลือกที่ต้องการ
6. ตรวจสอบว่าอยู่ในเมนูย่อยที่ถูกต้องหรือไม่ และปรับค่าสูงสุดต่ำสุดใหม่
7. เริ่มพิมพ์เมนูใหม่
8. ทำการ **clear** ค่าของ **keyboard**
9. กลับไปเริ่มกระบวนการทำซ้ำใหม่

งานของ **procedure writemenu** มีดังนี้

1. สั่งพิมพ์เมนูหลักบนจอภาพ
2. เลือกพิมพ์เมนูย่อยปัจจุบันใหม่ ไม่ได้พิมพ์เมนูย่อยทั้งหมด
3. กำหนดสีใหม่ให้ต่างไปจากที่พิมพ์เมนูย่อย
4. พิมพ์เฉพาะตัวเลือกปัจจุบัน ใหม่ให้ดูแตกต่าง

งานหลัก	ปรับปรุง	รายงาน	รายงานสรุป	ระบบ
		รายงานขายประจำเดือน		
		...		
		รายงานสินค้า ...		
		รายงานลูกหนี้ ...		
		รายงานเจ้าหนี้ ...		
		กลับเมนูหลัก		

งาน **midterm project** ต้องมีสิ่งต่อไปนี้

1. ปกนอก ปกใน คำนำ สารบัญ สรุป ประเมินการสอน
2. ปกต้องมีรูป **email** และ **homepage** ที่เก็บข้อมูลของ **project** ไว้แล้ว
3. มีเลขหน้า อธิบายโครงสร้างแฟ้ม และอธิบายโปรแกรม

## สัปดาห์ที่ 9

### ๑ สอบกลางภาค

## สัปดาห์ที่ 10

### **P** หัวข้อ 22 ทบทวน และเฉลย จุดที่นักศึกษาผิดพลาดกันมาก จากการสอบกลางภาค

#### 2 สรุปคำบรรยาย :

- .. นักศึกษาพลาดในเรื่องการหาค่าผลรวมกันมาก และการพิมพ์ข้อมูลแบบแยกหน้า ให้ทบทวนให้ดี แม้จะผ่านไปแล้ว แต่หลักการต่าง ๆ ต้องนำไปประยุกต์ เพื่อนำไปทำ **final project** ที่สมบูรณ์
- .. โปรแกรมเพิ่ม ลบ แก้ไขที่ออกสอบไป แม้จะง่าย แต่ก็มีนักศึกษาพลาด เช่นการตรวจค่าก่อนจัดเก็บ ข้อให้กลับไปทบทวนอีกครั้ง เพราะเป็นเรื่องสำคัญ
- .. เมนูแบบ **prompt** และ **achoice** ที่เขียนมามีความสับสน เพราะเขียนสลับกันเช่น **prompt** ไปเขียนแบบ **achoice** บ้างทำให้รูปแบบโปรแกรมผิดไปจาก **syntax** ที่ถูกต้อง

#### 2 สรุปกิจกรรม :

ศึกษาเทคนิคการทำข้อสอบ และชี้แจงจุดที่นักศึกษาผิดพลาด

#### 2 รายละเอียดกิจกรรม :

- บอกคะแนนสอบ
- นำโจทย์ข้อสอบกลางภาคมาอธิบายให้นักศึกษาฟังอีกครั้ง
- ให้นักศึกษาเขียนถึงจุดที่ตนเองผิดพลาดในชั้นเรียน และเปิดโอกาสให้ซักถามเพื่อนที่ได้คะแนนดี
- เปิดโอกาสให้นักศึกษามาแสดงความคิดเห็น และชี้จุดที่ผิดพลาดได้ง่าย โดยนักศึกษาเอง

### **P** หัวข้อ 23 ศึกษาวิธีการสุ่มตัวเลข และเลือกกระเป๋เงินจากเพิ่มข้อมูลมาพิมพ์

#### 2 สรุปคำบรรยาย :

- .. การสุ่มนั้นนักศึกษาเรียนมาจากข้อสอบแบบสุ่มแล้ว ต้องการให้ประยุกต์ดูการสุ่มเลขขึ้นมา แล้วไปเลือกกระเป๋เงินมาแสดงผล ซึ่งโครงสร้างโปรแกรมจะไม่ต่างไปจากโปรแกรมสุ่มข้อสอบมากนัก
- .. คำว่าเงื่อนไข ภาษาอังกฤษเรียกว่า **decision** ทุกภาษาจะมีคำว่า **if** และหลาย ๆ ภาษาจะมีคำว่า **case** ซึ่งสามารถเลือกได้หลาย ๆ เงื่อนไข

#### 2 สรุปกิจกรรม :

ทบทวนการเขียนโปรแกรมสุ่มตัวเลข เพื่อเลือกกระเป๋เงินที่ต้องการมาพิมพ์

#### 2 รายละเอียดกิจกรรม :

- เขียนวิธีการสุ่มตัวเลขให้นักศึกษาดูบนแผ่นใสอีกครั้ง เหมือนที่เคยสอนในหัวข้อที่ 17 แต่ปรับปรุงให้เลือกกระเป๋เงินตามค่าที่ต้องการเช่นการเลือกเฉพาะที่สถานภาพเป็น **A** เป็นต้น
- อธิบายคำสั่ง **set filter** หน้า 55

#### 2 งานมอบหมาย ครั้งที่ 14

1. ให้เขียนโปรแกรมสุ่ม กระเป๋เงินที่ต้องการ โดยระบุเงื่อนไขประกอบ

## 2 เขียนแผ่นใส :

วิธีการสุ่มตัวเลขในหน่วยที่ต้องการ

use empl

set filter to status = "A"

go top

count to allrec

tot = allrec

arum = array(allrec)

for I = 1 to allrec

    rnd = mod(second() \* 100,allrec) + 1      // สุ่มเลขที่ไม่เกินจำนวนระเบียบที่ตรงตามเงื่อนไข

    if ascan(arum,rnd) = 0                      // เช็คว่ามีซ้ำ

        arum[I] = rnd

    else

        I --

    endif

next

for I = 1 to allrec

    go arum[I]

    ? eid, ename, esurn, esalary, status

next

## P หัวข้อ 24 ศึกษาวิธีการใช้ปุ่มฟังก์ชันพื้นฐาน

### 2 สรุปคำบรรยาย :

- .. ปุ่มฟังก์ชันด้านบนสุดของแป้นพิมพ์ทั่วไป มีทั้งหมด 12 ปุ่มตั้งแต่ F1 ถึง F12 ซึ่งสามารถประยุกต์ให้มาทำหน้าที่ใน clipper ได้ เช่นการทำงานใน Dos mode การใช้ F3 หมายถึงการเรียกคำสั่งล่าสุดมาแสดงอีกครั้ง ส่วน F1 ในหลายโปรแกรมจะหมายถึงการเรียกโปรแกรม Help มาทำงาน หรือการแสดงส่วนช่วยเหลือนั่นเอง
- .. Clipper มีคำสั่ง set key จะกำหนดว่าปุ่มฟังก์ชันใด จะไปเรียก function ที่ผู้เขียน เขียนขึ้นตัวใด เช่นเมื่อกดปุ่ม F1 ให้ไปเรียก function ที่ผู้เขียนสร้างขึ้นชื่อ Helpuser หรือ กำหนดให้กด F2 แล้วไปเรียก Showtable เป็นต้น
- .. Final project เป็นสิ่งสำคัญกับวิชานี้ จึงไม่สั่งงานครั้งเดียว ป้อนกันการ copy จะทะยอยสั่ง ให้ปรับปรุงด้วยตนเอง และมีการติดตามความคืบหน้า ถ้านักศึกษาไม่ทำก็จะไม่เข้าใจว่า การสั่งงานครั้งต่อไปคืออะไร เพราะไม่ได้ทำมาก่อน

### 2 สรุปกิจกรรม :

ศึกษาการตรวจสอบ แป้นพิมพ์ และการเรียกใช้ function พร้อมตัวอย่างการใช้

### 2 รายละเอียดกิจกรรม :

- ศึกษาการใช้คำสั่ง set key จากหน้า 236 กรณีศึกษาระบบทะเบียน เพื่อเรียกฟังก์ชันที่เขียนขึ้นมาทำงาน
- ศึกษาการใช้คำสั่ง set key จากหน้า 295 กรณีศึกษาระบบขาย เพื่อเรียกฟังก์ชันที่เขียนขึ้นมาทำงาน

- อธิบายคำสั่ง set key หน้า 59
- 2** งานมอบหมาย ครั้งที่ 15
1. ให้เขียนโปรแกรม ที่เรียกใช้จากปุ่ม function เช่น help หรือ exit ได้ เป็นต้น
- สั่งงาน final project ครั้งที่ 1 :** ให้ออกแบบโครงสร้างแฟ้มของระบบที่ต้องการเขียน
- 2** เขียนแผนผัง:
- สิ่งที่ต้องมีใน final project
1. นาฬิกาเดินได้ในเมนู
  2. ทุกอย่างต้องอยู่ภายใต้ shadow
  3. ใช้ function F1 เรียก help, F2 เรียกข้อมูลที่เหมาะสมกับเขตข้อมูล, F3 เลิกการทำงาน
  4. มีระบบถามรหัสผ่านก่อนเปิดโปรแกรม
  5. รายงานต้องมีปกนอก ปกใน คำนำ สารบัญ สรุป ประเมิน
  6. ปกต้องมีรูป email และ homepage ที่เก็บข้อมูลของ project ไว้แล้ว
  7. มีเลขหน้า อธิบายโครงสร้างแฟ้ม อธิบายโปรแกรม และเขียนรูปของเมนูในกระดาษพร้อมอธิบายพอเข้าใจ
- ตัวอย่างของตัวเลือกที่ควรมี แต่ทั้งนี้ให้ขึ้นอยู่กับตัวระบบของนักศึกษาเอง
- มีตัวเลือกหลัก 5 ตัวเลือกคือ และในแต่ละตัวเลือกหลักกำหนดให้มีตัวเลือกย่อย
- งานหลักมี 4 ตัวเลือกย่อย งานปรับปรุงให้มี 4 ตัวเลือกย่อย
- รายงานให้มี 8 ตัวเลือกย่อย รายงานสรุปให้มี 4 ตัวเลือกย่อย ระบบให้มี 9 ตัวเลือกย่อย

1. งานหลัก	2. ปรับปรุง	3. รายงาน	4. รายงานสรุป	5. ระบบ
ชื่อ	ปรับ stock สินค้า	ยอดขาย	รับประจำช่วงวันที่	เกี่ยวกับโปรแกรม ...
ขาย	เพิ่มพนักงานใหม่	ข้อมูลลูกหนี้	จ่ายประจำช่วงวันที่	เกี่ยวกับผู้เขียน
ยืม	พนักงานลาออก	ใบทวงหนี้	สถิติขาย	Screen saver ...
คืน	เพิ่มสินค้าใหม่	สินค้าถึงจุด	ยอดขายดี ประจำเดือน	log file ...
ยกเลิกชื่อ	เพิ่มรายชื่อเจ้าหนี้	ยอดรับเงิน	สรุปค้าง stock	color options ...
ยกเลิกขาย	แก้ไขข้อมูลลูกหนี้	ยอดซื้อ	dos shell	
รับสินค้า	กำหนดสินค้าใหม่	ค้างส่ง	restore default ...	
คืนสินค้า	Browse ...	รับสินค้า	source code ...	
ประมวลสิ้นเดือน	Sort ...	Report ...		
บันทึกรับ				

## สัปดาห์ที่ 11

**P หัวข้อ 25** ศึกษาวิธีการใช้ปุ่มฟังก์ชัน ช่วยในการป้อนข้อมูลให้เหมาะสมกับเขตข้อมูล

**2** สรุปคำบรรยาย:

- Set key สามารถกำหนดปุ่ม ซึ่งมักมีผลจนรอรับค่าในคำสั่ง get ในหัวข้อนี้จะประยุกต์ให้ส่งชื่อตัวแปร ไปให้ฟังก์ชัน แล้วแสดงค่าที่เหมาะสมมาให้เลือก เพราะผู้ใช้อาจจำไม่ได้ หรือต้องการเลือกจากรายชื่อเพื่อความสะดวก

.. **Sort** เป็นคำสั่งที่ใช้จัดเรียงข้อมูลในแฟ้ม **Dbf** หลังจัดเรียงจะได้แฟ้มใหม่ที่ผ่านการจัดเรียงแล้ว

## 2 สรุปกิจกรรม :

ศึกษาการเรียกใช้ฟังก์ชันเพื่อช่วยการทำงานของกรรับค่าจากคำสั่ง **get**

## 2 รายละเอียดกิจกรรม :

- ศึกษาการใช้ปุ่มฟังก์ชันช่วยในการป้อนข้อมูล จากตัวอย่างระบบทะเบียน หน้า 236 และ 245
- ศึกษาการใช้ปุ่มฟังก์ชันช่วยในการป้อนข้อมูล จากตัวอย่างระบบขาย หน้า 295 และ 303
- ศึกษาการ **sort** ข้อมูลจากตัวอย่างหน้า 242
- อธิบายคำสั่ง **sort** หน้า 67

## 2 งานมอบหมาย ครั้งที่ 16

1. ให้เขียนโปรแกรมเรียกใช้ ฟังก์ชัน เลือกข้อมูลที่เหมาะสม ตามที่ศึกษาไป

ส่งงาน **final project ครั้งที่ 2** : ให้เขียนเมนู ของระบบ พร้อมกรอกข้อมูลในแฟ้มให้แล้วเสร็จ

**P หัวข้อ 26** ศึกษาระบบรักษาความปลอดภัยในการเรียกโปรแกรมมาประมวลผล เช่นการตรวจรหัสผ่านก่อนประมวลผลต่อไป

## 2 สรุปคำบรรยาย :

.. เรื่องความปลอดภัย (**Security**) กล่าวถึงกันมากในทุกภาษา แต่หน้าที่เริ่มต้นเป็นของ **programmer** ที่จะสร้างโปรแกรมให้ปลอดภัยเพียงใด แต่ผลสุดท้ายความปลอดภัยจะสมบูรณ์หรือไม่ต้องขึ้นกับผู้ใช้ ที่จะดูแลให้กฎเกณฑ์เป็นไปอย่างถูกต้อง มีระเบียบ และระมัดระวังเพียงใด

.. ปกติโปรแกรมต่าง ๆ ที่พัฒนาขึ้นมา เช่นโปรแกรมในแผนกบัญชี เมื่อเริ่มโปรแกรมจะถาม **username** และ **password** ก่อนเสมอ เพื่อให้ระบุว่าใครใช้ เมื่อเข้าไปแล้วมีสิทธิ์เพียงใด สำหรับหัวข้อนี้จะยังไม่ตรวจสอบรายละเอียด แต่นักศึกษาสามารถพัฒนาขึ้นมาเองได้ เพราะจะสอนเฉพาะการนำค่าที่พิมพ์เข้ามา ไปตรวจกับฐานข้อมูล ส่วนจะประยุกต์ให้รับค่าอื่น ๆ และนำไปตรวจในเมนูด้วยหรือไม่ ขอให้เป็นการเปิดทางเลือกให้นักศึกษาดีกว่า

## 2 สรุปกิจกรรม :

ทำความเข้าใจระบบรักษาความปลอดภัยของโปรแกรมอย่างง่าย แบบต่าง ๆ

ทดสอบการเขียนโปรแกรมตรวจสอบการเข้าใช้ได้อย่างง่าย โดยตรวจผู้ใช้จากแฟ้ม

## 2 รายละเอียดกิจกรรม :

- อธิบายหลักการของการรักษาความปลอดภัยอย่างง่ายบนแผ่นใส
- ให้นักศึกษาได้เขียนระบบรับ และตรวจจากแฟ้ม โดยให้ส่งงานในชั้นเรียน
- อธิบายคำสั่ง **exit** ซึ่งทำหน้าที่ออกจาก **loop while** ที่ใกล้ที่สุด
- อธิบายคำสั่ง **quit** หน้า 36
- อธิบายคำสั่ง **cancel** หน้า 21

## 2 เขียนแผ่นใส :

การรับค่ามาตรวจสอบกับตัวแปรที่กำหนดไว้

1. รับค่าด้วย `get` มา 2 ค่า คือ `username` และ `password`
2. นำค่าที่รับเข้ามาไปตรวจสอบกับตัวแปรที่กำหนดไว้ หากไม่ตรงให้ส่งหยุดโปรแกรมด้วยคำสั่ง
3. หากตรงให้เริ่มทำบรรทัดต่อไปตามปกติ

เช่น

```
username = "burin" ; password= "yonok"
```

```
l = 1
```

```
while .t.
```

```
u = space(8) ; p = space(8)
```

```
@ 5,5 say "username = " get u
```

```
@ 6,5 say "password = " get p
```

```
read
```

```
if u != username .or. p != password ; l + + ; else ; exit ; endif
```

```
if l = 3
```

```
quit
```

```
endif
```

```
end
```

## P หัวข้อ 27 ศึกษาการเรียกข้อมูลมาปรับปรุง หรือแสดงผลอย่างง่ายด้วย `Function browse` และ `dbedit`

### 2 สรุปคำบรรยาย :

- `Browse` เป็นคำสั่งที่ใช้เรียกข้อมูลในแฟ้ม `dbf` ทั้งแฟ้มมาแสดงในรูปแบบตารางคล้าย `excel` และสามารถเข้าไปเพิ่ม ลบ หรือแก้ไขข้อมูลได้ สะดวกในการ `key` ข้อมูลเข้าไป เพื่อสร้างข้อมูลเริ่มต้น ก่อนที่จะปล่อยให้โปรแกรมควบคุมทั้งหมด และปกติโปรแกรมต่าง ๆ จะไม่มีการเรียกคำสั่ง `browse` ขึ้นมาใช้งาน เพราะถ้ามีก็จะช้อนไว้ เนื่องจากการใช้ `browse` อาจทำให้ความสัมพันธ์ของตารางต่าง ๆ เสียไป เช่นการแก้ไขรหัสพนักงานในแฟ้มพนักงาน จะทำให้ข้อมูลพนักงานในแฟ้มขาย และแฟ้มเงินเดือนผิดพลาด แต่ถ้าใช้โปรแกรม เมื่อเปลี่ยนรหัส หรือกระทำการใด ๆ ก็จะทำให้การ `update` อย่างถูกต้องไปทั้งหมด
- `Dbedit` มีหน้าที่คล้าย `browse` แต่ยืดหยุ่น เพราะเขียนในลักษณะ `function` เลือก `field` ไปแสดงผลได้ และกำหนด ตำแหน่งที่จะวาง `dbedit` ลงไปได้เหมาะสม สามารถถูกนำมาใช้ในโปรแกรมได้ เพราะไม่ยอมให้ปรับปรุงข้อมูล จึงไม่ต้องห่วงเรื่องผลเสียหลังการปรับปรุง

### 2 สรุปกิจกรรม :

ศึกษาการใช้ฟังก์ชัน `browse` และ `dbedit`

ทดสอบการเรียกใช้ `browse` จากเมนู `prompt` หรือ `achoice`

## 2 รายละเอียดกิจกรรม :

- อธิบายฟังก์ชัน **browse** หน้า 90
- อธิบายฟังก์ชัน **dbedit** หน้า 98
- ศึกษาตัวอย่างการใช้ฟังก์ชัน **browse** หน้า 319
- ศึกษาตัวอย่างการใช้ฟังก์ชัน **dbedit** จากแผ่นใส

## 2 งานมอบหมาย ครั้งที่ 17

1. ให้เขียนโปรแกรมอ่านข้อมูลจากแฟ้มมา ผ่านฟังก์ชัน **browse** หรือ **dbedit** ตามความเหมาะสม

**สั่งงาน final project ครั้งที่ 3** : ให้ระบบเรียกข้อมูลจากแฟ้มมาปรับปรุงหรือแสดงด้วยได้

## 2 เขียนแผ่นใส :

ตัวอย่างการใช้ฟังก์ชัน **dbedit**

```
opt = alert("choose file", {"file1","file2","file3"})
```

```
do case
```

```
case opt =1
```

```
    use file1
```

```
    ar = {"field11","field12"}
```

```
// array ที่ประกอบด้วยชื่อ field จะส่งให้ dbedit
```

```
case opt =2
```

```
    use file2
```

```
    ar = {"field21","field22"}
```

```
case opt =3
```

```
    use file3
```

```
    ar = {"field31","field32"}
```

```
endcase
```

```
dbedit(5,10,20,70,ar)
```

ตัวอย่างการใช้ **dbedit** เลือกระเบียบที่ต้องการลบ

```
cls
```

```
use vdo
```

```
? recount()
```

```
// แสดงจำนวนระเบียบทั้งหมด
```

```
use vdo
```

```
x = dbedit(5,5,20,70,{"idmovie"})
```

```
? recno()
```

```
// แสดงเลขระเบียบปัจจุบันที่กำลังจะถูกลบ
```

```
delete
```

```
pack
```

```
? recount()
```

```
// แสดงจำนวนระเบียบทั้งหมด ลดลง 1 เพราะถูกลบ
```



## สัปดาห์ที่ 12

### P หัวข้อ 28 ศึกษาการจัดเรียงข้อมูลตามเขตข้อมูลที่ต้องการ

#### 2 สรุปคำบรรยาย :

- .. Sort คือการจัดเรียงวิธีหนึ่งทีจัดเรียงเนื้อหาของข้อมูลในแฟ้ม dbf จริง ๆ แต่มีวิธีอื่นที่ใช้จัดเรียงแบบภายนอก คือสร้างแฟ้มสำหรับจัดเรียงขึ้นมาต่างหากเรียกว่าแฟ้มดัชนี (index file) เมื่อจะใช้ต้องเรียกใช้ร่วมกัน
- .. Index file คือแฟ้มที่ใช้งานคู่กับ dbf หน้าหลักคือเรียงข้อมูลใน field ที่กำหนด จะเรียกใช้ต้องเรียกใช้ร่วมกัน ในแฟ้ม dbf หนึ่งสามารถมี index file ได้หลายแฟ้ม เพราะแต่ละแฟ้มกำหนดการจัดเรียงของ field ได้ 1 แบบเท่านั้น และการใช้ index file จะทำให้ใช้คำสั่ง set relation ซึ่งเป็นการ link แฟ้มต่าง ๆ เข้าด้วยกันโดยง่าย
- .. Set relation เป็นความสามารถที่ยอดเยี่ยม เพราะการเชื่อมแฟ้มเข้าด้วยการผ่าน key field ในระบบ index file จะทำให้เรียกข้อมูลที่สัมพันธ์กันมาใช้ได้ทันที เช่นเชื่อมแฟ้มชื่อ และแฟ้มเงินเดือน จะเขียน loop อย่างสั้น ๆ เพื่อแสดงได้ทั้งชื่อ และข้อมูลเงินเดือน แต่ถ้าไม่เชื่อมกัน จะต้องใช้คำสั่ง locate ค้นหาแต่ละครั้งเมื่อต้องการข้อมูล

#### 2 สรุปกิจกรรม :

การใช้คำสั่ง sort จัดเรียงแฟ้มข้อมูลตามต้องการ และเปรียบเทียบกับการใช้ index  
ทำความเข้าใจ และศึกษาการใช้ index เพื่อจัดเรียงข้อมูล  
ทดสอบการใช้คำสั่ง sort ร่วมกับเมนู และรายงาน

#### 2 รายละเอียดกิจกรรม :

- อธิบายคำสั่ง sort หน้า 67 ซึ่งต้องใช้แฟ้มมารองรับผลการจัดเรียง และใช้เวลา
- ศึกษาการจัดเรียงข้อมูลด้วยคำสั่ง sort หน้า 279 และ 336
- อธิบายคำสั่ง index หน้า 31 ใช้แฟ้มมารองรับแต่ขนาด และความเร็วมากกว่า sort
- ศึกษาแนวคิดการจัดเรียงด้วย index หน้า 334, 336
- อธิบายคำสั่ง set relation หน้า 63 มีประสิทธิภาพในการเชื่อมต่อแฟ้มเป็นอย่างมาก
- ศึกษาการใช้ set relation หน้า 328

#### .2 งานมอบหมาย ครั้งที่ 18

1. ให้เขียนโปรแกรมเพื่อเรียกข้อมูลมาจัดทำรายงาน พร้อมสามารถเลือกที่จะจัดเรียงได้

#### 2 เขียนแผ่นใส :

ตัวอย่างการใช้ sort แล้วจัดทำรายงาน

```
use vdo
sort on idmovie /D to vdonew
use vdonew
while !eof()
    ? idmovie
    skip
end
```

ตัวอย่างการใช้ index แล้วจัดทำรายงาน

```
use vdo
index on idmovie to ivdo
use vdo index ivdo
```

```

while !eof()
    ? idmovie
    skip
end
ตัวอย่างการใช้ set relation
use vdo index xxvdo
use member index yymember
set relation to memid into vdo
while !eof()
    ? memid, vdo->idvdo, vdo->vdoname
    skip
end

```

## P หัวข้อ 29 ศึกษาการตรวจสอบข้อมูลที่ป้อนเข้าไป ก่อนจัดเก็บ หรือปรับปรุง

### 2 สรุปคำบรรยาย :

- .. ความต้องการเพิ่ม ลบ หรือแก้ไขข้อมูล คือพื้นฐานของการเขียนโปรแกรม และทั้ง 3 วิธีต้องมีการตรวจสอบก่อนเสมอว่าข้อมูลที่ต้องการกระทำนั้น เหมาะสมหรือไม่ เพราะถ้าไม่เหมาะสม จะส่งผลโดยรวมต่อระบบฐานข้อมูลทันที
- .. การเข้าใจ 3 เรื่องนี้จะทำให้สามารถประยุกต์ไปสร้างโปรแกรมอื่น ๆ ได้อีกมาก เช่นโปรแกรม ชื่อ ชาย ยิ้ม คีน

### 2 สรุปกิจกรรม :

ทบทวนการรับข้อมูลไปตรวจสอบอย่างระมัดระวังก่อนจัดเก็บ

### 2 รายละเอียดกิจกรรม :

- ศึกษาโปรแกรม เพิ่ม ลบ แก้ไขจากหน้า 222 และ 223
- ศึกษาการตรวจสอบค่าก่อนจัดเก็บจาก แผ่นใส

### 2 เขียนแผ่นใส :

ตัวอย่างตรวจสอบก่อนเพิ่มระเบียบใหม่

```

use empl
_emplid = 0 ; _ename = space(20)
@ 5,5 get _emplid
@ 6,5 get _ename
read
locate for emplid = _emplid
if ! found()
    append blank
    replace emplid with _emplid , ename with _ename
endif

```

ตัวอย่างตรวจสอบก่อนลบระเบียบ

```

use empl
_emplid = 0 ; _ename = space(20)
@ 5,5 get _emplid ; read

```

```

locate for emplid = _emplid
if found()
    askopt = alert ("Are you sure?",{ "yes", "no" })
    if askopt = 1 ; delete ; pack ; endif
endif

```

ตัวอย่างการแก้ไขระเบียบ

```

use empl
_emplid = 0 ; _ename = space(20)
@ 5,5 get _emplid
@ 6,5 get _ename
read
locate for emplid = _emplid
if found()
    replace emplid with _emplid , ename with _ename
endif

```

**P หัวข้อ 30** ศึกษาโปรแกรมปรับปรุงข้อมูลครั้งละหลายแฟ้ม เช่นโปรแกรมขาย อย่างง่าย

**2** สรุปคำบรรยาย :

- โปรแกรมปรับปรุงที่ดี ควรรับข้อมูลเพียงครั้งเดียว แต่ส่งผลกระทบต่อแฟ้มที่เกี่ยวข้องทั้งหมด เช่นการป้อนข้อมูลการขายเมื่อป้อนแล้วควรส่งข้อมูลเข้าแฟ้มขาย แฟ้มขายละเอียด แฟ้มลูกหนี้ และตัดยอดสินค้าในแฟ้มสินค้า
- โปรแกรมขายนั้นเขียนได้หลายวิธี ถ้าเข้าใจตัวอย่าง แล้วขอให้ประยุกต์ไปใช้ให้ตรงกับความต้องการของตนเอง

**2** สรุปกิจกรรม :

ศึกษาแนวคิดการเขียนโปรแกรมสำหรับขาย เช่น ยืม หรือจอง เป็นต้น

**2** รายละเอียดกิจกรรม :

- ศึกษาโปรแกรมขายโดยใช้ตัวแปรธรรมดาเพียงชุดเดียวอย่างง่าย หน้า 305
- ศึกษาโปรแกรมขายโดยใช้ตัวแปรหลายชุด หน้า 307
- ศึกษาโปรแกรมขายโดยใช้ตัวแปรอาเรย์ หน้า 310

**2** งานมอบหมาย ครั้งที่ 19

1. ให้เขียนโปรแกรมขายตามตัวอย่างที่ให้ ให้ใช้งานได้เบื้องต้น

**2** เขียนแผนใส่ :

หลักการดำเนินงานของโปรแกรมขาย

1. กำหนดค่าเริ่มต้น สำหรับแฟ้มขาย
2. รับค่าเก็บเข้าแฟ้มขาย เช่น ชื่อลูกค้า เป็นต้น
3. เตรียมทำซ้ำส่วนขายละเอียด
4. กำหนดค่าเริ่มต้น สำหรับแฟ้มขายละเอียด
5. รับข้อมูลการขายละเอียด เช่นรหัสสินค้า จำนวน ราคา เก็บลงแฟ้มขาย
6. เมื่อรับข้อมูลเรียบร้อยแล้ว จะเข้าส่วนพิมพ์ใบเสร็จ

## สัปดาห์ที่ 13

### P หัวข้อ 31 ศึกษาโปรแกรมปรับปรุงข้อมูลครั้งละหลายแฟ้ม เช่นโปรแกรมขาย โดยตรวจสอบ array

#### 2 สรุปคำบรรยาย :

- .. การรับข้อมูลการขาย เข้าไปเก็บในอาเรย์ เพราะต้องการความสะดวกรวดเร็ว และยืดหยุ่น บางท่านอาจใช้วิธีเก็บเข้า ตัวแปร หรือแฟ้มชั่วคราว แต่ผมเห็นว่าอารีย์นั้นน่าจะดีที่สุด เพราะเร็ว และเรียกใช้สะดวก การเขียนโปรแกรมก็ไม่ซับซ้อน
- .. เพื่อส่งเสริมให้นักศึกษาเข้าใจ ขอให้นักศึกษาเขียน flowchart ของโปรแกรมขาย เพื่อจะได้เห็น process การทำงานต่าง ๆ หากนักศึกษาเข้าใจจะสามารถประยุกต์กับโปรแกรมขาย ชื่อ ยืม คืนได้ไม่ยาก

#### 2 สรุปกิจกรรม :

ศึกษาการปรับปรุง โดยใช้ array มารับค่าเหมือนกับ buffer ก่อนจัดเก็บลงแฟ้มจริง

#### 2 รายละเอียดกิจกรรม :

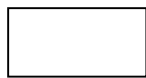
- ศึกษาโปรแกรมขายโดยใช้ตัวแปรธรรมดาเพียงชุดเดียวอย่างง่าย หน้า 305 (ทบทวน เพราะยาก)
- ศึกษาโปรแกรมขายโดยใช้ตัวแปรหลายชุด หน้า 307 (ทบทวน เพราะยาก)
- ศึกษาโปรแกรมขายโดยใช้ตัวแปรอาเรย์ หน้า 310 (ทบทวน เพราะยาก)
- ให้นักศึกษาเขียน flowchart แสดงการทำงาน และให้ส่งในชั้นเรียน

#### 2 งานมอบหมาย ครั้งที่ 20

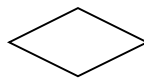
1. ให้เขียนโปรแกรมจัดเก็บข้อมูล โดยใช้หลักการของ array เข้าช่วย

ส่งงาน final project ครั้งที่ 4 : ให้เขียนโปรแกรมรับข้อมูลในระบบของตน ซึ่งเป็นงานหลักของระบบ

#### 2 เขียนแผ่นใส :



หมายถึงการประมวลผล



หมายถึงการตัดสินใจ



หมายถึงการทำงานกับข้อมูล



หมายถึงการพิมพ์

### P หัวข้อ 32 ศึกษาโปรแกรมปรับปรุงข้อมูลครั้งละหลายแฟ้ม เช่นโปรแกรมขาย โดยใช้แฟ้มชั่วคราว

#### 2 สรุปคำบรรยาย :

- .. การใช้ array เก็บข้อมูลการขายเป็นวิธีการหนึ่ง ในบทนี้จะให้ใช้แฟ้มข้อมูลชั่วคราวมารับค่า เพื่อให้เข้าใจว่าเทคนิคนี้เป็นอย่างไร เพราะอาจเหมาะสมกับสถานการณ์ หรืออาจต้องการเก็บข้อมูลนั้นไว้ระยะหนึ่งจริง ๆ
- .. Append from เป็นคำสั่งที่ใช้ copy ข้อมูลจากแฟ้ม dbf แฟ้มหนึ่งไปต่อท้ายแฟ้ม dbf อีกแฟ้มหนึ่ง ในกรณีนี้ก็คือการนำข้อมูลจากแฟ้มขายชั่วคราว หลังตรวจสอบ และปรับปรุงแฟ้มอื่นแล้ว ส่งเข้าไปต่อท้ายแฟ้มขายจริง

#### 2 สรุปกิจกรรม :

ศึกษาการปรับปรุง โดยใช้แฟ้มชั่วคราว ซึ่งเป็นอีกหลักการหนึ่งของการทำโปรแกรมหลัก

## 2 รายละเอียดกิจกรรม :

- ศึกษาการทำงานกับแฟ้มชั่วคราว แทนการใช้อาเรย์ จากแผ่นใส
- ศึกษาข้อดี และข้อเสียของแฟ้มชั่วคราว โดยให้นักศึกษาได้ปรึกษากันในชั้นเรียน แล้วสรุปส่ง
- อธิบายคำสั่ง delete หน้า 27
- อธิบายคำสั่ง pack หน้า 36
- อธิบายคำสั่ง append from หน้า 20
- ศึกษา และทบทวนการใช้ประโยชน์จาก procedure

## 2 เขียนแผ่นใส :

การเพิ่มระเบียบโดยใช้แฟ้มชั่วคราวในการจัดเก็บข้อมูล

```

use empl
delete all
pack
while .t.
    _emplid = 0 ; _ename = space(20)
    @ 5,5 get _emplid
    @ 6,5 get _ename
    read
    append blank
    replace emplid with _emplid , ename with _ename
    askopt = alert ("Continue ?",{ "yes", "no" })
    if askopt != 1 ; exit ; endif
end
do printfile
askopt = alert ("Put all record to main file ?",{ "yes", "no" })
if askopt = 1
    use mainempl
    append from empl
    do printfile
endif
procedure printfile
    l = 1
    while !eof()
        ? l, emplid, ename
        l ++
        skip
    end
return

```

## P หัวข้อ 33 ศึกษาการเขียนโปรแกรมรายงาน โดยเรียกข้อมูลจากหลายแฟ้ม

### 2 สรุปคำบรรยาย:

- .. รายงานคือการอ่านข้อมูลมาพิมพ์ หลาย ๆ ท่านคิดว่าการทำรายงานง่าย แต่ความจริงแล้ว รายงานนั้นยากที่จะทำ ให้ผู้ใช้พอใจตลอดไป เพราะผู้ใช้มักมีความต้องการเพิ่มขึ้นเสมอ และบางความต้องการ อาจต้องไปปรับปรุงระบบฐาน ข้อมูล และโปรแกรมทั้งหมดใหม่ เพื่อให้ได้รายงานที่ไม่คาดคิดว่าจะขอ .. แต่ programmer ปฏิเสธไม่ได้
- .. คำสั่งที่นิยมใช้กันบ่อย ๆ ในรายงานคือ locate และ set filter เพราะต้องให้เลือกกลุ่มข้อมูลที่ต้องการ

### 2 สรุปกิจกรรม:

ศึกษาการเรียกแฟ้มข้อมูล โดยใช้หลักการของ select แบบพื้นที่เรียกแฟ้มมาใช้งาน

### 2 รายละเอียดกิจกรรม:

- ศึกษาโปรแกรมรายงานระบบทะเบียน หน้า 263
- ศึกษาโปรแกรมรายงานระบบขาย หน้า 337
- ศึกษาการเชื่อมแฟ้มด้วย locate จากแผ่นใส และอธิบายคำสั่ง locate หน้า 34
- ศึกษาการเชื่อมแฟ้มด้วย set filter จากแผ่นใส และอธิบายคำสั่ง set filter หน้า 55

### 2 งานมอบหมาย ครั้งที่ 21

1. ให้เขียนโปรแกรม เชื่อมแฟ้มหลายแฟ้มเข้าด้วยกัน ไปทำรายงาน

สั่งงาน final project ครั้งที่ 5 : ให้ปรับปรุงส่วนของการออกรายงานให้ได้ 10 รายงาน

### 2 เขียนแผ่นใส:

ตัวอย่างการเชื่อมแฟ้มด้วย locate

```
sele 1 ; use cust
```

```
sele 2 ; use sale
```

```
while ! eof()
```

```
    _cid = cid
```

```
    sele 1
```

```
        locate for cid = _cid
```

```
        if found() ; _cname = cname ; endif
```

```
    sele 2
```

```
    ? sid, cid, _cname
```

```
    skip
```

```
end
```

ตัวอย่างการเชื่อมแฟ้มด้วย set filter

```
sele 1 ; use saled
```

```
sele 2 ; use sale
```

```
while ! eof()
```

```
    _sid = sid
```

```
    ? sid, cid, sdate
```

```
    sele 1
```

```
        set filter to sid = _sid
```

```
        go top
```

```
        while ! eof()
```

```

? sid, proid, quantity, price
skip
end
sele 2
skip
end

```

## สัปดาห์ที่ 14

**P หัวข้อ 34** ศึกษาการเขียนโปรแกรมรายงานหาผลรวม โดยตรวจเงื่อนไขจากหลายแฟ้ม

2 สรุปคำบรรยาย :

- .. dtoc เป็นฟังก์ชันเปลี่ยนวันที่เป็นตัวอักษร เพราะอาจต้องนำไปใช้ให้เหมาะกับความต้องการ
- .. ctod เป็นฟังก์ชันเปลี่ยนตัวอักษรเป็นวันที่ เพราะอาจต้องการส่งค่าวันที่เข้าไปในระบบ หรือใช้คำนวณ
- .. ย้ำอีกครั้งว่า การหาผลรวมจะมีการคำนวณใน loop แต่การแสดงผลรวมจะแสดงหลังจากจบ loop แล้ว เช่นการหาผลรวมของเงินเดือน การหาจำนวนนักศึกษาที่อยู่ในเงื่อนไข หรือนับเกรดแต่ละเกรด เป็นต้น
- .. คำว่า loop ในภาษา clipper ไม่มี แต่มีความหมายคือการทำซ้ำ โดยมีคำสั่งที่ใช้ใน clipper คือ while และ for

2 สรุปกิจกรรม :

ศึกษาการเรียกแฟ้มข้อมูล มาทำรายงานโดยเลือกข้อมูลจากหลายแฟ้มได้

2 รายละเอียดกิจกรรม :

- ศึกษาการหาผลรวมจากการคำนวณจากแผ่นใส
- ศึกษาการเลือกข้อมูลจากหลายแฟ้ม ให้ได้รายงานที่ต้องการจากแผ่นใส
- อธิบายคำสั่ง dtoc หน้า 109
- อธิบายคำสั่ง ctod หน้า 92
- อธิบายคำสั่ง set epocd หน้า 53

2 งานมอบหมาย ครั้งที่ 22

1. ให้เขียนโปรแกรม เชื่อมแฟ้มหลายแฟ้ม พร้อมกำหนดเงื่อนไข ไปทำรายงาน

สั่งงาน final project ครั้งที่ 6 : ให้ปรับปรุงส่วนของการออกรายงานให้ได้อีก 5 รวมเป็น 15 รายงาน

2 เขียนแผ่นใส :

ตัวอย่างโปรแกรมหาผลรวมจากการคำนวณ

```

use salary ; sum = 0
while ! eof()
    ? eid, ename, salary
    sum = sum + salary
skip
end

```

end

? sum

ตัวอย่างโปรแกรมเลือกเงื่อนไขจากหลายแฟ้ม เพื่อทำรายงาน

เช่นต้องการที่วันที่ขายเป็นวันที่ 31 มกราคม 2543 และปริมาณที่มาแสดงต้องมากกว่า 5

```

sele 1 ; use saled
sele 2 ; use sale
while ! eof()
    _sid = sid
    if sdate = ctod("1/31/2000")
        ? sid, cid, sdate
    sele 1
        set filter to sid = _sid .and. quantity > 5
        go top
        while ! eof()
            ? sid, proid, quantity, price
            skip
        end
    sele 2
end
endif
skip
end

```

## P หัวข้อ 35 ศึกษาการจัดเก็บตัวแปร และการเรียกตัวแปรที่จัดเก็บในแฟ้ม เพื่อใช้เก็บค่าสีของจอภาพ

### 2 สรุปคำบรรยาย :

- .. การเก็บค่าที่ผ่านมาจะเก็บในแฟ้ม **dbf** เท่านั้น แต่ในหัวข้อนี้ จะแนะนำการจัดเก็บตัวแปรไว้ในแฟ้มตัวแปร โดยใช้คำสั่ง **save** หากจะเรียกตัวแปรที่จัดเก็บมาให้ใช้คำสั่ง **restore** ทำให้ประยุกต์โปรแกรมจนสามารถจดจำค่าต่าง ๆ เช่น การเลือกสีพื้นของโปรแกรมเหมือนที่โปรแกรม **windows** ทำได้นั่นเอง
- .. การกำหนดสีใน **clipper** มี 2 วิธีคือการใช้ฟังก์ชัน **setcolor** หรือคำสั่ง **set color to**

### 2 สรุปกิจกรรม :

ศึกษาการจัดเก็บตัวแปร ลงแฟ้ม เพื่อประยุกต์ไปใช้กับการเก็บต่าง ๆ แทนการเก็บลงแฟ้ม

### 2 รายละเอียดกิจกรรม :

- อธิบายคำสั่ง **save** หน้า 42 เพื่อจัดเก็บตัวแปรลงแฟ้มตัวแปร
- อธิบายคำสั่ง **restore** หน้า 40 เพื่อเรียกตัวแปรกลับคืนมา
- อธิบายฟังก์ชัน **setcolor** หน้า 165 เพื่อกำหนดสีให้ระบบ และใช้ตัวแปรได้ ต่างจาก **set color to** ที่ใช้ไม่ได้
- ศึกษาหน้าที่ของโปรแกรมชุดเกี่ยวกับการจัดเก็บค่าสี

### 2 งานมอบหมาย ครั้งที่ 23

1. ให้เขียนโปรแกรม เพื่อประยุกต์ใช้การเก็บค่าสีลงแฟ้มตัวแปร เพื่อกำหนดสีสำหรับแสดงผลบนจอภาพ

### 2 เขียนแผ่นใส :

ตัวอย่างโปรแกรมเริ่มต้น ซึ่งประมวลผลครั้งเดียวเพื่อเก็บค่า

```

cc = "n/w"
setcolor(cc)
cls
? "test"
save all to varcc

```

ตัวอย่างโปรแกรมเรียกตัวแปรที่เก็บไว้ในแฟ้มตัวแปรมาใช้



```

setcolor("n/w")
cls
? "test"
restore from varcc
setcolor(cc)
? "test"
ตัวอย่างโปรแกรมเปลี่ยนค่าในแฟ้มตัวแปร และการเรียกใช้งาน
cls
restore from varcc
setcolor(cc)
@ 5,5 prompt "n/r"
@ 6,5 prompt "b/n"
@ 7,5 prompt "w/b"
menu to opt
do case
case opt = 1 ; cc = "n/r"
case opt = 2 ; cc = "b/n"
case opt = 3 ; cc = "w/b"
endcase
if opt > 0
setcolor(cc)
save all to varcc
endif
? "test"

```

## P หัวข้อ 36 ศึกษาการทำโปรแกรม screen saver อย่างง่าย

### 2 สรุปคำบรรยาย :

- .. หลักการพื้นฐานของ screen saver คือให้โปรแกรม run อยู่ตลอดเวลา เมื่อหยุดกดปุ่มใด ๆ จะเริ่มนับ เพื่อการนับถึงค่าที่กำหนด จะไปเรียกโปรแกรมอีกตัวหนึ่งมาทำงาน มักเป็นโปรแกรมสร้างอะไรขึ้นมาบนจอภาพก็ได้ และมีการเคลื่อนไหวตลอดเวลา บางคนอาจสั่งให้จอดำมืดไปเลยก็ถือเป็นหลักการนี้เช่นกัน
- .. ตัวอย่างบนแผ่นใสจะเป็น screen saver อย่างง่ายเพียงให้เข้าใจพื้นฐาน และสามารถนำไปแก้ไขใน final project

### 2 สรุปกิจกรรม :

ศึกษาหลักการทำงานของการทำซ้ำ และการเขียน screen saver อย่างง่าย แบบ text

### 2 รายละเอียดกิจกรรม :

- ทบทวนโปรแกรมสร้างลูกศรวิ่ง จากแผ่นใสในหัวข้อที่ 19
- โปรแกรมนี้มีปัญหาให้นักศึกษาศึกษาถึงปัญหา และปรับปรุงโปรแกรมโดยลดปัญหาลง แล้วส่งท้ายชั่วโมง ตัวอย่างปัญหาเช่น เมนูหายไปหลังเลือกกด 2 เป็นต้น และเมนูจะไม่กลับมาอีกเลย

### 2 เขียนแผ่นใส :

ตัวอย่างลูกศรวิ่งไป และกลับ แต่ให้ทำงานกับระบบเมนู

```

cls
i = 10
? "1. print 1 to 10"
? "2. clear"

```

```

? "3. exit"
while .t.
    inkey(0.2)
    do case
    case lastkey() = 49 ; for j = 1 to 10 ; ? j ; next
    case lastkey() = 50 ; cls
    case lastkey() = 51 ; exit
    endcase
    do arrow
    keyboard chr(0)
    inkey()
end
procedure arrow
    if i <= 60
        @ 10,i say " =>"
    else
        @ 10,61 - mod(i,60) say "<= "
        if i > 110; i = 9; endif
    endif
    i++
    inkey(0.1)
return

```

## สัปดาห์ที่ 15

**P หัวข้อ 37** ศึกษาการอ่าน Source code มาแสดงผลอย่างง่าย

**2** สรุปคำบรรยาย :

- .. e.com เป็นโปรแกรมหนึ่งที่สามารถอ่านข้อมูลมาแสดงผลอย่างสั้น ๆ ได้ เป็นโปรแกรมภายนอกเหมือนกับ edit ในระบบ Dos หรือ Q editor แต่ตัวเล็กกว่า มีหน้าที่เพียงแสดง text file และสามารถสั่ง print จะแก้ไขข้อมูลไม่ได้
- .. Memo เป็นกลุ่มคำสั่งที่ใช้จัดการกับข้อมูลแบบ text หลายบรรทัดที่มีประสิทธิภาพอย่างมาก ถ้าใช้อย่างระมัดระวังจะสามารถทำให้คล้ายกับ text displayer ได้ทีเดียว

**2** สรุปกิจกรรม :

ศึกษาการอ่านแฟ้ม prg ใน diskette มาพิมพ์ อย่างเป็นระเบียบ  
ศึกษาการใช้คำสั่ง memo เพื่อจัดการกับแฟ้มแบบ text

## 2 รายละเอียดกิจกรรม :

- ศึกษาการใช้โปรแกรม e.com ซึ่งช่วยให้สามารถอ่าน prg หรือ txt มาแสดงอย่างง่ายได้เป็นอย่างดี พร้อม print ได้ เช่น run e x.prg เป็นต้น
- อธิบายฟังก์ชัน memoread หน้า 138 ทำหน้าที่อ่านข้อมูลจากแฟ้มส่งเข้าตัวแปร
- อธิบายฟังก์ชัน memotran หน้า 139 ทำหน้าที่เปลี่ยน หรือแทนที่(replace) ตัวอักษรที่ต้องการได้โดยง่าย
- อธิบายฟังก์ชัน memowrit หน้า 139 ทำหน้าที่เขียนตัวแปรลงไปในแฟ้ม
- อธิบายฟังก์ชัน memoline หน้า 138 ทำหน้าที่นำค่าจากตัวแปรที่ถูกเก็บอย่างแยกบรรทัดมาใช้ที่ละบรรทัด
- อธิบายฟังก์ชัน memoedit หน้า 136 ทำหน้าที่แสดง และแก้ไขข้อมูลในตัวแปรแบบ memo ได้อย่างดี
- อธิบายฟังก์ชัน mlcount หน้า 141 ทำหน้าที่นับบรรทัดในแฟ้มข้อความ
- ให้นักศึกษาสร้างโจทย์ และใช้งาน memo ให้ครบทุกคำสั่ง และส่งในชั้นเรียน

## P หัวข้อ 38 ศึกษาการจัดเก็บชื่อ วัน และเวลา การเข้าใช้โปรแกรมของผู้ใช้

### 2 สรุปคำบรรยาย :

- มีคำสั่งเกี่ยวกับเวลามากมายที่แนะนำในหัวข้อนี้ เพื่อให้สามารถนำค่าวันที่ และเวลาไปเก็บในแฟ้ม และนำมาใช้ได้ อย่างถูกต้อง เช่น dtos หมายถึงเปลี่ยนตัววันที่เป็นชุดตัวอักษรไม่มีเครื่องหมาย / เป็นต้น
- set device เป็นการกำหนดว่าจะให้แสดงผลทางจอภาพหรือ printer มักใช้คู่กับ set printer เพราะถ้าต้องการเขียนข้อมูลจัดเก็บใน text file จะต้องกำหนด set printer ไปที่ เช่น set printer to "x.txt" เป็นต้น

### 2 สรุปกิจกรรม :

ศึกษาหลักการ การจัดเก็บร่องรอยการใช้โปรแกรมของผู้ใช้

วิธีการเปลี่ยนรูปแบบของค่า เช่นวันที่ เวลา ตัวเลข หรือตัวอักษร

### 2 รายละเอียดกิจกรรม :

- จุดที่แนะนำให้เก็บร่องรอยการใช้โปรแกรม
  1. เมื่อเริ่มต้น และปิดโปรแกรม
  2. เมื่อเริ่ม และปิด กระทำตัวเลือก หรือโปรแกรมย่อยใด
- การจัดเก็บข้อมูลในแฟ้มแบบ text มีหลายวิธี
  1. คำสั่ง set device หน้า 52 และ set printer หน้า 62
  2. ฟังก์ชัน memowrit หน้า 139
  3. คำสั่ง text หน้า 68 ซึ่งส่งผลออกทางเครื่องพิมพ์ หรือแฟ้มได้
- อธิบายฟังก์ชัน ctod หน้า 92 ทำหน้าที่เปลี่ยนตัวอักษรเป็นวันที่
- อธิบายฟังก์ชัน dtoc หน้า 109 ทำหน้าที่เปลี่ยนวันที่เป็นตัวอักษร ในรูปแบบ dd/mm/yy
- อธิบายฟังก์ชัน dtos หน้า 110 ทำหน้าที่เปลี่ยนวันที่เป็นตัวอักษร ในรูปแบบ yyyyymmdd
- อธิบายฟังก์ชัน date หน้า 93 ทำหน้าที่แสดงวันที่ มีตัวอย่างการใช้หน้า 194
- อธิบายฟังก์ชัน time หน้า 171 ทำหน้าที่แสดงเวลา มีตัวอย่างการใช้หน้า 194

## P หัวข้อ 39 ศึกษาการทำระบบแสดงส่วนช่วยเหลือ หรือแสดงประวัติผู้เขียนจากแฟ้มข้อมูล

### 2 สรุปคำบรรยาย:

- .. คำสั่งต่าง ๆ ในหัวข้อนี้ท่านเคยศึกษามาแล้ว แต่การประยุกต์ให้แสดงเป็นส่วนช่วยเหลือ ขอให้เป็นการทบทวน และประยุกต์ใช้งาน เพื่อนำไปทำ final project
- .. การแสดงส่วนช่วยเหลือ กำหนดให้มี 3 วิธีคือ say, อ่านจาก dbf และอ่านจาก text file ด้วย memo read
- .. ก่อนแสดงส่วนช่วยเหลือควร save screen หลังแสดงส่วนช่วยเหลือให้ restore screen เพื่อให้ผลลัพธ์มีระเบียบ

### 2 สรุปกิจกรรม:

ศึกษาการใช้ปุ่ม function เพื่อเติมเพื่อเรียกข้อมูลจากฐานข้อมูลมาแสดง  
ทบทวนการใช้ savescreen ในสถานการณ์ที่เหมาะสม

### 2 งานมอบหมาย ครั้งที่ 24

1. ให้เขียนโปรแกรม เรียกฟังก์ชันจากแป้นพิมพ์ เพื่อแสดงค่าวันเวลาจากแฟ้ม  
สั่งงาน final project ครั้งที่ 7 : ให้ปรับปรุงส่วนของ help และประวัติผู้เขียน ให้สมบูรณ์

### 2 รายละเอียดกิจกรรม:

- การแสดงประวัติผู้เขียน หรือรายละเอียดโปรแกรม สามารถกระทำได้หลายวิธี
  1. ใช้การ say จากค่าคงที่ที่กำหนดในโปรแกรม
  2. อ่านมาจาก dbf เช่นการใช้ use while ? skip และ end
  3. อ่านมาจาก text เช่นการใช้ memoread mlcount และ memoline
- อธิบายคำสั่ง save screen หน้า 43 ทำหน้าที่จัดเก็บภาพของหน้าจอปัจจุบันไว้
- อธิบายคำสั่ง restore screen หน้า 40 ทำหน้าที่จำภาพของหน้าจอที่จัดเก็บไว้กลับมาแสดง
- อธิบายคำสั่ง set date หน้า 49

### 2 เขียนแผ่นใส:

ประวัติของผู้เขียนควรประกอบด้วย

ชื่อ - สกุล

ที่อยู่

e-mail และ homepage ที่จัดเก็บ project ขึ้นนี้

ประวัติการศึกษา

ภูมิลำเนา

สิ่งที่ชอบ และไม่ชอบ

ประสบการณ์ในชีวิต

ประสบการณ์การศึกษา

โปรแกรมที่เคยใช้ และใช้เป็น

ส่วนช่วยเหลือควรประกอบด้วย

โครงสร้างแฟ้ม โดยละเอียด รวมถึงขนาดของแต่ละเขตข้อมูล และระเบียน

ลักษณะของโปรแกรม ว่าทำอะไรได้บ้าง หรืออาจละเอียดถึงระดับนำเมนูมาอธิบาย

แนวการออกแบบระบบนี้ได้แนวคิด และแรงจูงใจ ที่จะให้มีขอบเขตเพียงใด

จุดเด่น จุดด้อย สิ่งอำนวยความสะดวกที่เตรียมไว้ และข้อจำกัดของโปรแกรม  
วิธีการเขียน และการแปลโปรแกรม หรือตัวภาษาที่ใช้  
แฟ้มข้อมูล โปรแกรม หรือ **source program** ที่ใช้ทั้งหมดมีอะไรบ้าง  
ลักษณะของข้อมูลในแต่ละเขตข้อมูล (data dictionary)

## สัปดาห์ที่ 16

**P หัวข้อ 40** ศึกษาการทำรายงานทางแฟ้ม เพื่อนำไปจัดรูปแบบ และพิมพ์ในโปรแกรมอื่น

**2** สรุปคำบรรยาย :

- .. ปกติรายงานสามารถ **print** ออกไปทาง **printer** ได้ แต่บ่อยครั้งที่ต้องการพิมพ์รายงานเก็บเป็น **text file** หรือในระบบ **CSV (Comma separate variable)** ซึ่งสะดวกในการนำไปใช้ใน **word** หรือ **excel**
- .. ตัวอย่างในระบบบัญชี การจ่ายเงินเดือนบุคลากร **1000** คน จำเป็นต้อง **key** และตรวจสอบทุกคน การส่งให้ธนาคารเป็นกระดาษอย่างเดียวอาจมีปัญหา จึงได้จัดให้ส่งเป็น **diskette** เพื่อนำไปเข้าระบบ ทางธนาคารเพียงแต่ตรวจสอบ ไม่จำเป็นต้องพิมพ์ใหม่ **1000** คนเข้าไปทุกเดือน

**2** สรุปกิจกรรม :

ศึกษาการสร้างรายงานลงไปในแฟ้ม เพื่อนำไปเปิดด้วย **word** หรือ **text editor** ใด ๆ ได้  
ศึกษาการจัดรูปแบบข้อมูล และรหัสต่าง ๆ ที่ใช้พิมพ์ลงไปในแฟ้มแบบ **text**

**2** รายละเอียดกิจกรรม :

- อธิบายคำสั่ง **set decimals** หน้า 50
- อธิบายฟังก์ชัน **asc** หน้า 85
- อธิบายฟังก์ชัน **chr** หน้า 91
- ทบทวนคำสั่ง ? และ ?? หน้า 14
- ทบทวนคำสั่ง @ .. **say** หน้า 18
- ทบทวนการจัดเก็บข้อมูลในแฟ้มแบบ **text** มีหลายวิธี
  1. คำสั่ง **set device** หน้า 52 และ **set printer** หน้า 62
  2. ฟังก์ชัน **memowrit** หน้า 139
  3. คำสั่ง **text** หน้า 68 ซึ่งส่งผลออกทางเครื่องพิมพ์ หรือแฟ้มได้
- ให้นักศึกษาได้ฝึกจัดทำรายงานออกทางแฟ้ม โดยมีลำดับ การคำนวณ หาผลรวม เลือกข้อมูล จากหลายแฟ้ม มาทำรายงาน เมื่อเสร็จแล้วให้ส่งในชั้นเรียน

**P หัวข้อ 41** ศึกษาการแปลโปรแกรมด้วย **RMAKE**

**2** สรุปคำบรรยาย :

- .. การเขียน **procedure** ภายนอก เมื่อแปล (**compile**) จะต้องแปลทุก **procedure** การใช้ **rmake** จะช่วยแปลเฉพาะ **procedure** ที่ปรับปรุงเท่านั้น เพราะเครื่องเก่าจะทำงานช้า การใช้ **rmake** จะช่วยให้แปลโปรแกรมได้เร็วขึ้น

- .. ปัญหาคือนักศึกษาหลายคนไม่ได้แยก **procedure** ออกมาจากโปรแกรมหลัก จะเขียน **procedure** เก็บไว้ภายใน ทำให้มี **prg** เพียงตัวเดียว จึงให้นักศึกษาทำการแยก **procedure** ออกมา และแปลใหม่ และให้ทดสอบแก้เพียงบาง **procedure** จะพบว่าผลการแก้ไข จะเป็นการเรียกบาง **procedure** มา แปลเท่านั้น ไม่ได้แปลใหม่ทุก **procedure**

## 2 สรุปกิจกรรม :

ศึกษาหลักการในการแปลโปรแกรมแบบต่าง ๆ เพื่อให้การนำไปใช้ เหมาะสมกับงานที่สุด  
ให้นักศึกษาเขียนโปรแกรมเพื่อสั่งให้แปลแบบ **RMAKE** โดยอาศัยใช้ **final project** ของแต่ละคน

## 2 รายละเอียดกิจกรรม :

- ศึกษาหลักการของ **rmake** ในบทที่ 1 หน้า 4
- ทำความเข้าใจหลักการของ **rmake** จากแผ่นใส
- จุดเด่นของ **rmake** คือเมื่อแก้ไขเฉพาะ **prg** ตัวใด จะทำการ **compile** ใหม่เฉพาะ **prg** ตัวนั้น แล้วนำ **obj** ที่ผ่านการแปล มารวมกันใหม่ จนได้ **exe** ที่ต้องการ

## 2 เขียนแผ่นใส :

สิ่งที่ต้องกระทำในการแปลโปรแกรมด้วย **rmake**

1. สร้างแฟ้ม นามสกุล **rmk** เพื่อรวบรวมชื่อโปรแกรมที่ต้องการแปลด้วย **rmake**
2. โดยปกติการแปลโปรแกรมด้วย **rmake** จะนำแฟ้มสกุล **prg** มาหลาย ๆ โปรแกรม แต่ไม่เกิน 11 โปรแกรม
3. โปรแกรมทุกตัว ต้องขึ้นต้นด้วยคำว่า **procedure** หรือ **function** จะเริ่มด้วยบรรทัดคำสั่งตามปกติไม่ได้
4. สั่งแปลด้วยคำว่า **c:\rmake x** หรือ **c:\rmake x.rmk** ก็ได้ ซึ่ง **x** มีนามสกุล **rmk** แต่จะละไว้ก็ได้

## P หัวข้อ 42 สรุปเนื้อหาวิชาที่ผ่านมา

### 2 สรุปคำบรรยาย :

- .. ให้แต่ละคนเล่าประสบการณ์ จากการทำ **final project** และเสนอแนะว่าการสอนที่ผ่านมาเป็นอย่างไร ซึ่งเชื่อแน่ว่าทุกคนต้องบอกว่างานเยอะ แต่ขอให้รู้ว่ำนั่นคือประสบการณ์ การรู้จักบริหารเวลา การเรียนรู้ภาษาใหม่อีกภาษาหนึ่ง พอที่จะไปประกอบอาชีพได้
- .. ข้อสอบปลายภาคก็เหมือน **final project** เพียงแต่ตัดมาเป็นบางส่วนเป็นข้อเท่านั้น ถ้าทำ **project** กันเองด้วยความตั้งใจ และทบทวน เชื่อแน่ว่า **final exam** ต้องได้คะแนนเต็ม

### 2 สรุปกิจกรรม :

ให้นักศึกษานำเสนอ **final project**

สรุปเนื้อหา และสิ่งที่ได้รับจาก **project** ให้นักศึกษาฟัง

เปิดโอกาสให้นักศึกษา แสดงความคิดเห็น ต่อวิชา เนื้อหา และวิธีการสอน หน้าชั้นเรียน

รับรายงาน **final project** ที่นักศึกษาพิมพ์รายงานเป็นรูปเล่ม พร้อมโปรแกรม

นัดให้นักศึกษา กลับมารับรายงานคืน หลังจากสอบปลายภาคไปแล้ว

## สัปดาห์ที่ 17

### ๑ สอบปลายภาค