

บทที่ 2

โครงสร้างของโปรแกรม และตัวอย่างโปรแกรม

COBOL ย่อมาจากคำว่า **Common Business Oriented Language** เป็นภาษาที่นิยมกันแพร่หลายมาก ภาษานี้เกิดขึ้นเนื่องจากผลของความพยายามของบริษัทผู้สร้างเครื่องคอมพิวเตอร์หลาย ๆ บริษัทร่วมมือกับผู้ใช้บริการของคอมพิวเตอร์ในอันที่จะเขียนโปรแกรมให้ได้ ด้วยภาษาที่เป็นมาตรฐานและเหมาะสมกับการประมวลผลข้อมูลทางด้านธุรกิจโดยเฉพาะ ซึ่งประสบผลสำเร็จเป็นที่ยอมรับและใช้งานได้ในปี ค.ศ. 1959

2.1 ลักษณะเฉพาะของภาษาโคบอล

1. Standardize for all computers

โคบอลเป็นภาษาที่มีมาตรฐาน นำไปใช้ได้กับคอมพิวเตอร์ทุกบริษัท เช่น นักศึกษาเรียนภาษานี้ โดยยึดเอาคู่มือ (manual) ของบริษัทคอมพิวเตอร์ CDC เป็นหลัก แม้จะไม่ได้นำไปโปรแกรมไปประมวลผลกับเครื่องคอมพิวเตอร์ของ CDC ก็อาจเอาไปประมวลผลกับคอมพิวเตอร์บริษัท IBM หรือ VAX หรือ NEAC หรืออื่น ๆ ได้ คือ โปรแกรมเมอร์สามารถที่จะไปทำงานกับคอมพิวเตอร์ของบริษัทอื่น ๆ ได้เสมอ

2. Easy to understand

ภาษานี้เข้าใจง่าย เพราะการเขียนคำสั่งให้ปฏิบัติตามขั้นตอน เขียนด้วยคำสั่งที่เป็นประโยคภาษาอังกฤษธรรมดาทั่วไป แม้ว่าผู้อ่านโปรแกรม หรือผู้ที่จะนำเอาโปรแกรมไปใช้งานต่อจะไม่ได้เป็นตัวโปรแกรมเมอร์เองก็ตาม ก็สามารถเข้าใจได้ง่าย

ตัวอย่าง

ADD A TO B GIVING C.

ADD 100, X, B, TOTAL TO RESULT.

MULTIPLY X BY Y GIVING Z.

MOVE 100 TO PERCENT.

IF CK-1 GREATER THAN 5 GO TO P-LINE.

COMPUTE SUM = 100 + (A * B) / D.

3. Oriented to business procedure

ภาษานี้สร้างขึ้นมาเพื่อประมวลผลข้อมูลทางธุรกิจโดยเฉพาะ ลักษณะขั้นตอนเกี่ยวกับลำดับคำสั่งจึงเหมือนกับทำด้วยมือตนเอง ไม่มีศัพท์เทคนิคเกี่ยวกับตัวเครื่องมากนัก ปัจจุบันเป็นภาษาที่นิยมใช้กันแพร่หลายที่สุด

แต่ทั้งนี้ไม่ได้หมายความว่าภาษานี้จะเขียนโปรแกรมทำงานด้านอื่น ๆ ไม่ได้ เราใช้ภาษาโคบอลเขียนโปรแกรมเพื่อทำงานทางด้าน scientific applications ได้ แต่ถ้าเปรียบเทียบกันระหว่างภาษาโคบอลกับภาษาฟอร์แทรนในงานด้านนี้แล้ว การเขียนโปรแกรมด้วยภาษาฟอร์แทรน สะดวกกว่า และในทำนองเดียวกัน งานทางด้านธุรกิจ การเขียนโปรแกรมด้วยภาษาโคบอล จะสะดวกกว่าเขียนด้วยภาษาฟอร์แทรน เหตุผลก็ดังที่ได้กล่าวมาตั้งแต่ตอนแรกแล้วว่า แต่ละภาษาสร้างขึ้นมาเพื่อให้มีความเหมาะสมเฉพาะด้านหรือเฉพาะงานเป็นอย่างไร ๆ ไป

2.2 โครงสร้างของโปรแกรม (Program structures)

โปรแกรมแต่ละโปรแกรมที่เขียนด้วยภาษาโคบอลจะประกอบด้วย 4 ส่วน แต่ละส่วนเรียกว่า division ทุก division สามารถเขียนแยกเป็นอิสระต่างหากจากกันได้ เมื่อเขียนเป็นโปรแกรมที่สมบูรณ์เรียบร้อยแล้ว โปรแกรมเมอร์ต้องเรียงลำดับ division ดังนี้

Identification	division
Environment	division
Data	division
Procedure	division

1. **Identification division** เป็น division แรกที่เขียนง่ายและสั้นที่สุด ส่วนนี้แสดงรายละเอียดเกี่ยวกับชื่อโปรแกรม, ชื่อผู้เขียนโปรแกรม, วัน เดือน ปี ที่เขียนโปรแกรม, วัน เดือน ปี ที่คอมไพล์โปรแกรม วัตถุประสงค์ในการเขียนโปรแกรมนี้นวมทั้งข้อจำกัดต่าง ๆ ของโปรแกรม เพื่อให้เป็นข้อมูลแก่ผู้จะนำโปรแกรมนี้ไปใช้ต่อได้ทราบไว้

2. **Environment division** ส่วนนี้แสดงรายละเอียดเกี่ยวกับตัวเครื่องคอมพิวเตอร์ที่ใช้ในการคอมไพล์และ execute โปรแกรม เช่น เป็นคอมพิวเตอร์ของบริษัทอะไร นั่นคือ บอกชื่อของคอมพิวเตอร์ว่าเป็น CDC 3100 หรือ IBM 360 หรือ NEAC 2200 หรือ VAX-11 เป็นต้น และบอกขนาดหน่วยความจำของคอมพิวเตอร์เครื่องนั้น

นอกจากนี้แล้วใน division นี้ยังมีอีกพารากราฟหนึ่งที่สำคัญ เรียกว่า พารากราฟ special-names ซึ่งให้อิสระแก่โปรแกรมเมอร์ในการตั้งชื่อ mnemonic name หรือ symbolic name ให้แก่ input/output devices ต่าง ๆ

3. Data division เป็น division ที่แสดงรายละเอียดเกี่ยวกับข้อมูลทุกอย่างที่นำมาใช้ในโปรแกรมนี้ ข้อมูลที่เป็นอินพุท (input) ข้อมูลที่เป็นเอาต์พุท (output) และข้อมูลที่เกิดขึ้นในระหว่างการคำนวณว่ามีลักษณะอย่างไร

ชื่อที่โปรแกรมเมอร์ตั้งให้กับข้อมูลใน division นี้มีสองชนิด คือ data-name กับ condition name

4. Procedure division เป็น division สุดท้ายที่มีความสำคัญมากที่สุด มิได้เขียนคำสั่งต่าง ๆ ที่ให้คอมพิวเตอร์ทำงาน ตามลำดับขั้นตอน เพื่อให้ได้ผลลัพธ์ออกมาตามต้องการและยังเป็นตัวจัดการให้เลือกทำคำสั่งตามเงื่อนไขต่าง ๆ กัน ชื่อที่ตั้งขึ้นใน division นี้เรียกว่า procedure-name หรือ paragraph name และ section name

ตัวอย่าง 1 โปรแกรมข้างล่างนี้อ่านบัตรข้อมูลจำนวนหนึ่งในบัตรข้อมูลแต่ละใบจะบันทึกหมายเลขเจ้าหน้าที่ในหน่วยคอมพิวเตอร์ของมหาวิทยาลัยรามคำแหง เมื่อปี พ.ศ. 2517 ชื่อและเงินเดือนของเจ้าหน้าที่ผู้นั้น จากนั้นนำข้อมูลทั้งหมดบันทึกลงในเทปและให้พิมพ์ออกมาทางพรินเตอร์ในลักษณะที่สวยงามด้วย

3300 COBOL BCD/VER-3.0 FOR RTS

EMPLOYEE

```
000010 IDENTIFICATION DIVISION.  
000020 PROGRAM-ID. EMPLOYEE-FILE.  
000030 AUTHOR. SOMPHIT.  
000040 DATE-WRITTEN. 25/12/74.  
000050 DATE-COMPILED.  
  
000060 ENVIRONMENT DIVISION.  
000070 CONFIGURATION SECTION.  
000080 SOURCE-COMPUTER. 3100 MEMORY SIZE IS 16K WORDS.  
000090 OBJECT-COMPUTER. 3100 MEMORY SIZE IS 16K WORDS.  
000100 INPUT-OUTPUT SECTION.  
000110 FILE-CONTROL.  
000120 SELECT CARD-FILE ASSIGN TO CARD-READER 60,
```

000130 SELECT PRNT-FILE ASSIGN TO PRINTER 61
~~000140~~ SELECT TAPE-FILE ASSIGN TO TAPE 01.

000150 DATA DIVISION,
OF10160 FILE SECTION.

000170 FD CARD-FILE LABEL RECORD IS OMITTED DATA RECORD IS
000180 CARD-REC.

~~000190~~ 01 CARD-REC.

000200 02 EMP-NO PICTURE Y(5).

~~000210~~ 02 FILLER PICTURE X(4).

000220 02 EMP-NAME PICTURE X(30).

000230 02 SALARY PICTURE Y(4).

000240 02 FILLER PICTURE X(37).

~~000250~~ FD TAPE-FILE RECORDING MODE IS **DECIMAL HIGH** DENSITY LABEL
~~000260~~ RECORD IS STANDARD VALUE OF ID IS "RU-0001" BLOCK
000270 CONTAINS 10 RECORDS RECORD CONTAINS **80** CHARACTERS
000280 DATA RECORD IS TAPE-REC.

000290 01 TAPE-REC.

000300 02 FILLER PICTURE X(80).

000310 FD PRNT-FILE LABEL RECORD IS OMITTED DATA RECORD IS
000320 **PRNT-REC.**

~~000330~~ 01 PRNT-REC.

000340 02 FILLER PICTURE X(136).

000350 WORKING-STORAGE SECTION.

000360 01 HEADING-1.

000370 02 FILLER PICTURE X(60) VALUE "1".

~~000380~~ 02 FILLER PICTURE X(16) VALUE "COMPUTER SECTION".

~~000390~~ 02 FILLER PICTURE X(60) VALUE SPACES.

000400 01 HEADING-2.

```

000410      02 FILLER PICTURE X(55) VALUE SPACES.
000420      02 FILLER PICTURE X(27) VALUE
(100430      "(RAMKHAMHAENG UNIVERSITY)".
000440      02 FILLER PICTURE X(54) VALUE SPACES.

000450 01  HEADING-3.
000460      02 FILLER PICTURE X VALUE "0".
000470      02 FILLER PICTURE X(15) VALUE SPACES.
000480      02 FILLER PICTURE X(105) VALUE ALL " * ".
000490      02 FILLER PICTURE X(15) VALUE SPACES.

000500 01  HEADING-4.
000510      02 FILLER PICTURE X(3X) VALUE "0".
000520      02 FILLER PICTURE X(12) VALUE "EMPLOYEE NO.".
000530      02 FILLER PICTURE X(14) VALUE SPACES.
000540      02 FILLER PICTURE X(13) VALUE "EMPLOYEE NAME".
000550      02 FILLER PICTURE X(14) VALUE SPACES,
000560      02 FILLER PICTURE X(06) VALUE "SALARY"
000570      02 FILLER PICTURE X(39) VALUE SPACES.

000580 01  DETAILS.
000590      02 FILLER PICTURE X(41).
000600      02 EMP-AREA PICTURE 9(5).
000610      02 FILLER PICTURE X( 10).
000620      02 NAME-AREA PICTURE X(30).
000630      02 FILLER PICTURE X(6).
000640      02 SALARY-AREA PICTURE Z,Z99.
000650      02 FILLER PICTURE X(40).
000660 PROCEDURE DIVISION.

000670 START-1.
000680      OPEN INPUT CARD-FILE.

```

000690 OPEN OUTPUT TAPE-FILE, PRNT-FILE.
000700 MOVE SPACES TO TAPE-REC, PRNT-REC.
000710 MOVE SPACES TO DETAILS.

000720 START-2.
000730 WRITE PRNT-REC FROM HEADING-1.
000740 WRITE PRNT-REC FROM HEADING-2.
000750 WRITE PRNT-REC FROM HEADING-3.
000760 WRITE PRNT-REC FROM HEADING-4.

000770 PROCESS.
000780 READ CARD-FILE AT END GO TO E-O-J.
000790 WRITE TAPE-REC FROM CARD-REC.
000800 MOVE EMP-NO TO EMP-AREA.
000810 MOVE EMP-NAME TO NAME-AREA.
000820 MOVE SALARY TO SALARY-AREA.
000830 WRITE PRNT-REC FROM DETAILS.
000840 GO TO PROCESS.

000850 E-O-J.
000860 CLOSE CARD-FILE, TAPE-FILE, PRNT-FILE
000570 STOP RUN.

3300 COBOL BCD/VER-3.0 FOR RTS

EMPLOYEE

I COBL 088 NUMBER DIAGNOSTICS 0000

EXTERNAL SYMBOLS

BCDBOXS

READ

OPENI

OPENO

CLOSE

WRITEF

EDITCOBL

ABNORMAL

EQUIP, 01 = MTCOE0U01

LOAD,56

RUN,,NM

COMPUTER SECTION
(RAMKHAMHAENG UNIVERSITY)

EMPLOYEE NO.	EMPLOYEE NAME	SALARY
00001	MR. PAIROCH	3,850
00002	MRS. MALEE	2,520
00003	MR. MORAKOT	2,230
00004	MR. PINYO	2,230
00005	MISS SOMPHIT	2,230
00006	MRS. VENA	1,860
00007	MR. SUCHART	1,375
00008	MRS. JARUWON	1,150
00000	MR. MANUS	1,150
00010	MR. SATHIEN	1,150
00011	MISS SUCHADA	1,015
00012	MISS YUPIN	1,015
00013	MISS PUNNEE	800
00014	MR. BUNDIT	800

ELAPSED TIME 00HRS 03MINS 52SECS

ตัวอย่าง 2 โปรแกรมนี้อ่านบัตรข้อมูลจำนวนหนึ่งซึ่งบันทึกคะแนนสอบย่อย 3 ครั้งของนักศึกษาแล้วคำนวณหาคะแนนเฉลี่ย หากเกรดที่นักศึกษจะได้รับโดยมีหลักเกณฑ์การคิดเกรดดังนี้

คะแนนเฉลี่ย 90 คะแนนขึ้นไป	ได้เกรด A
คะแนนเฉลี่ยระหว่าง 80-89	ได้เกรด B
คะแนนเฉลี่ยระหว่าง 70-79	ได้เกรด C
คะแนนเฉลี่ยระหว่าง 60-69	ได้เกรด D
คะแนนเฉลี่ยตั้งแต่ 59 คะแนนลงมา	ได้เกรด F

IDENTIFICATION DIVISION.

PROGRAM-ID. SAMPLE-1.

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

SOURCE-COMPUTER. 3100 MEMORY SIZE IS 16K WORDS.

OBJECT-COMPUTER. 3100 MEMORY SIZE IS 16K WORDS.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT STD-TEST-SCORES ASSIGN TO CARD-READER 60.

SELECT STD-FINAL-GRADES ASSIGN TO PRINTER 61.

DATA DIVISION.

FILE SECTION.

FD STD-TEST-SCORES LABEL RECORD IS OMITTED DATA RECORD IS IN-REC.

01 IN-REC.

02 STD-NAME PICTURE X(25).

02 TEST-SCORE-1 PICTURE 999.

02 TEST-SCORE-2 PICTURE 999.

02 TEST-SCORE-3 PICTURE 999.

02 FILLER PICTURE X(46).

FD STD-FINAL-GRAGES LABEL RECORD IS OMITTED DATA RECORDS

ARE'HEADING, STANDARD-LINE.

01 HEADING.

02 FILLER PICTURE X(59).

02 LITERAL-1 PICTURE X(14).

02 FILLER PICTURE X(63).

01 STANDARD-LINE.

02 FILLER PICTURE X(15).

02 NAME-OUT PICTURE X(25).

02 FILLER PICTURE X(15).

02 GRADE PICTURE A.

02 FILLER PICTURE X(80).

WORKING-STORAGE SECTION.

77 SUM-SCORES PICTURE 999 VALUE ZEROS.

77 AVERAGE PICTURE 999 VALUE ZEROS.

PROCEDURE DIVISION.

START.

OPEN INPUT STD-TEST-SCORES OUTPUT STD-FINAL-GRADES.

MOVE SPACES TO HEADING.

MOVE 'STUDENT GRADES' TO LITERAL-I.

WRITE HEADING.
BEGIN.
MOVE SPACES TO STANDARD-LINE.
READ STD-TEST-SCORES AT END GO TO FINISH.
MOVE STD-NAME TO NAME-OUT.
ADD TEST-SCORE-1, TEST-SCORE-2, TEST-SCORE-3.GIVING SUM-SCORES.
DIVIDE 3 INTO SUM-SCORES GIVING AVERAGE.
IF AVERAGE IS GREATER THAN 89 MOVE 'A' TO GRADE GO TO PRINT.
IF AVERAGE IS GREATER THAN 79 MOVE 'B' TO GRADE GO TO PRINT.
IF AVERAGE IS GREATER THAN 69 MOVE 'C' TO GRADE GO TO PRINT.
IF AVERAGE IS GREATER THAN 59 MOVE 'D' TO GRADE GO TO PRINT.
MOVE 'F' TO GRADE.
PRINT.
WRITE STANDARD-LINE.
GO TO BEGIN.
FINISH.
CLOSE STD-TEST-SCORES, STD-FINAL-GRADES.
STOP RUN.

แบบฝึกหัด

จงเติมคำหรือข้อความลงในช่องว่างให้ได้ความหมายถูกต้อง

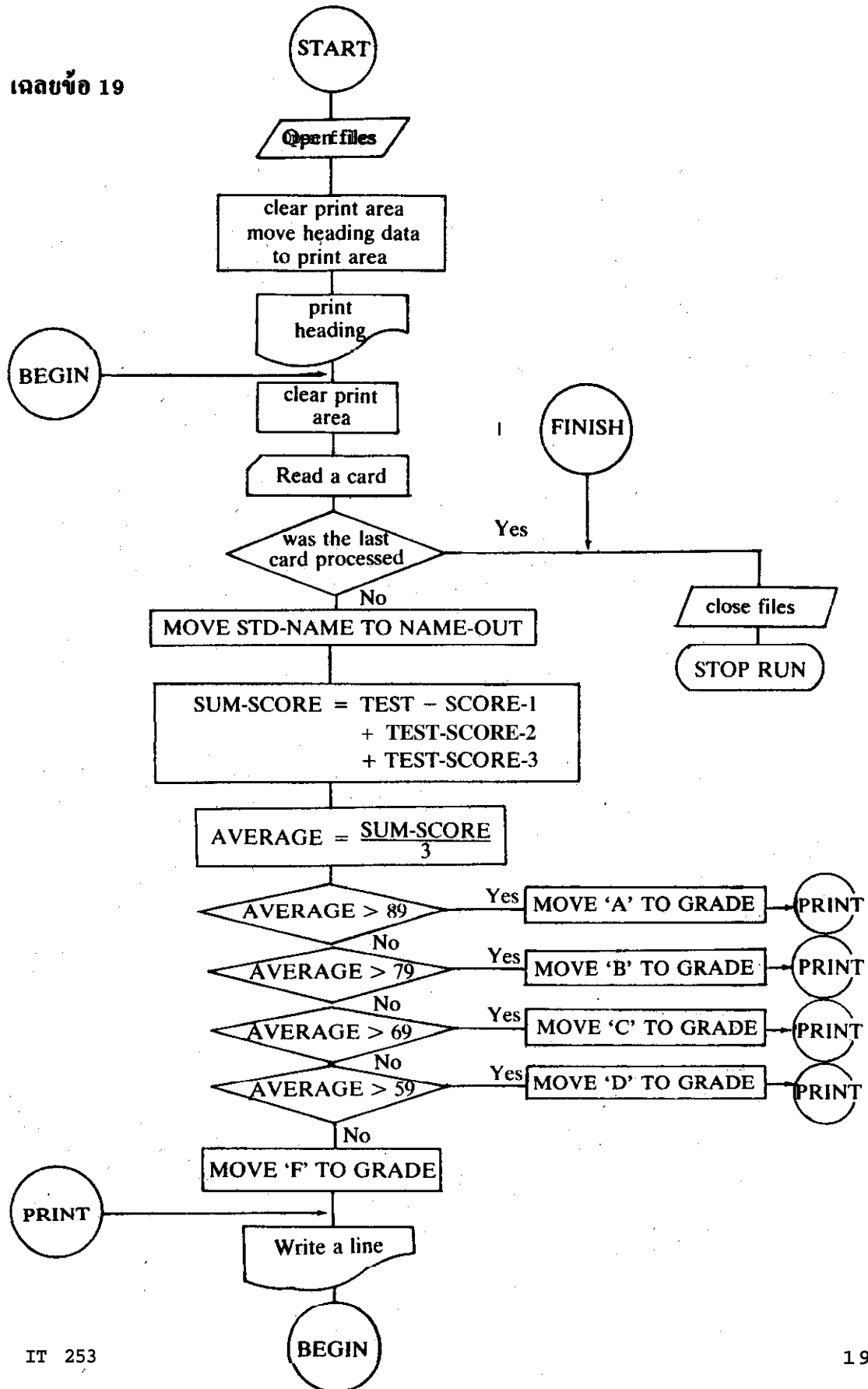
1. วัตถุประสงค์ของ Identification division คือ
2. Environment division ของโปรแกรมนี้ มีไว้เพื่อ.....
3. เพิ่มข้อมูลทั้ง 2 ชุดที่ใช้ในโปรแกรมนี้นี้ชื่อ
4. เพิ่มข้อมูลชุดแรกเป็นเพิ่มข้อมูลประเภท
5. เพิ่มข้อมูลชุดที่สองเป็นเพิ่มข้อมูลประเภท.....
6. card file ไม่มี label เรคคอร์ด และ logical เรคคอร์ดของมันชื่อ.....
7. ฟیلด์แรกของ card file ชื่อ
8. ฟیلด์ดังกล่าวข้างต้นต้องใช้เนื้อที่.....ตำแหน่ง และต้องเป็นฟیلด์ที่มีลักษณะดังนี้

9. สามฟیلด์ถัดไปของ card file หมายถึง มีความยาว 3 ตำแหน่ง จะมีมูลค่าจริงได้มากที่สุดเท่ากับ.....
10. ฟیلด์ที่มีค่า FILLER มีความหมายว่า
11. FILLER เป็นฟیلด์ที่อยู่ในบัตร หมายถึง คอลัมน์
12. print file ประกอบด้วย logical เรคคอร์ด ชนิด
13. ให้วาดรูปแสดงลักษณะของเรคคอร์ดชื่อ HEADING
14. ฟیلด์ชื่อ LITERAL-I จะมีข้อมูล..... บันทึกที่อยู่
15. ทำไมจึงต้องมีตำแหน่ง FILLER ระหว่าง date field ต่าง ๆ ในเรคคอร์ดชื่อ
STANDARD-LINE?
16. WORKING-STORAGE SECTION มีไว้สำหรับทำอะไร ?
17. คำสั่ง OPEN มีไว้ทำไม?
18. คำสั่ง MOVE SPACES TO HEADING. มีความหมายอย่างไร ?
19. ให้เขียน flowchart ของโปรแกรมนี้นี้
20. คำสั่ง CLOSE มีไว้ทำไม ?
21. คำสั่ง STOP RUN มีไว้ทำไม ?
22. ชื่อพารากราฟในโปรแกรมนี้นี้คือ
23. วัตถุประสงค์ของชื่อพารากราฟคือ

เฉลย

1. กำหนดชื่อโปรแกรม
2. บอกชนิดของ equipments ที่ใช้ในโปรแกรม
3. STD-TEST-SCORES และ STD-FINAL-GRADES
4. card file
5. print file
6. IN-REC
7. STD-NAME
8. 25, เป็นตัวอักษร
9. คะแนนสอบย่อยสามครั้ง, 100
10. เนื้อที่ซึ่งไม่ใช้
11. 35 ถึง 80
12. สอง
13. คอลัมน์ 1-59 เว้นว่าง
คอลัมน์ 60-73 พิมพ์ STUDENT GRADES
คอลัมน์ 74-80 เว้นว่าง
14. STUDENT GRADES
15. ต้องการเว้นช่องว่างระหว่างฟิลด์ต่าง ๆ ในเรคคอร์ดนี้เพื่อให้อ่านสะดวกขึ้น
16. SUM-SCORES และ AVERAGE เป็นฟิลด์ที่ไม่ได้เป็นทั้งอินพุทหรือเอาท์พุทฟิลด์ แต่ต้องจองเนื้อที่เอาไว้ใช้ในการปฏิบัติงาน ดังนั้น WORKING-STORAGE SECTION จึงมีไว้สำหรับกำหนดมูลค่าคงที่ และเนื้อที่สำหรับปฏิบัติงาน
17. เตรียมอินพุทและเอาท์พุทไฟล์ให้พร้อมที่จะทำงาน
18. ทำเนื้อที่ในหน่วยความจำชื่อ HEADING ให้ว่างทั้งหมด โดยการใส่ตัวอักษร blanks
19.
20. เป็นการบอกให้รู้ว่าแฟ้มข้อมูลชุดนั้นไม่ใช้แล้ว
21. อนุญาตให้อ่านโปรแกรมถัดไปเข้ามาปฏิบัติงานได้
22. START, BEGIN, PRINT และ FINISH
23. จัดตำแหน่งของกลุ่มคำสั่ง เพื่อประโยชน์ในการเรียกเอามาใช้ตามเงื่อนไขต่าง ๆ กันภายหลัง

เลขข้อ 19



ตัวอย่าง 3 โปรแกรมอ่านข้อมูลจำนวนหนึ่งซึ่งบันทึกในดิสก์ (disk) ข้อมูลแต่ละเรคคอร์ดมีรายละเอียดดังนี้ หมายเลขสินค้า, จำนวนหน่วยที่ผลิต, ราคาต้นทุนต่อหน่วย, ชื่อสินค้า แล้วคำนวณมูลค่าทั้งหมดของสินค้าชนิดนั้น กำหนดให้ใช้รูปแบบของอินพุท/เอาต์พุทข้างล่างนี้

Input

คอลัมน์	
1-4	ว่างไว้
5-9	หมายเลขสินค้า
10-12	จำนวนหน่วยที่ผลิต
13-16	ราคาต้นทุนต่อหน่วย (ทศนิยม 2 ตำแหน่ง)
17-36	ชื่อสินค้า
37-80	ว่างไว้

ตัวอย่าง

ลักษณะของอินพุท

```
002300010100ELECTRIC-FAN
002360150035ELECTRIC-POT
002451001525ELECTRIC-IRON
002775250375ELECTRIC-DRYER
```

output

คอลัมน์		คอลัมน์	
1-5	ว่างไว้	6-10	หมายเลขสินค้า
11-15	ว่างไว้	16-35	ชื่อสินค้า
36-39	ว่างไว้	40-43	จำนวนหน่วยที่ผลิต
44-48	ว่างไว้	49-53	ราคาต้นทุนต่อหน่วย (ทศนิยม 2 ตำแหน่ง)
54-58	ว่างไว้	59-69	มูลค่าทั้งหมดของสินค้าชนิดนั้น (ทศนิยม 2 ตำแหน่ง)
70-136	ว่างไว้		

ตัวอย่าง

เข้าที่ทุกจะมีลักษณะดังนี้

STOCK CALCULATION REPORT

00230	ELECTRIC FAN	1	1.00	1.00
00236	ELECTRIC POT	15	15.25	5.25
00245	ELECTRIC IRON	100		1,525.00
00277	ELECTRIC DRIER	525	3.75	1,968.75

```

1 IDENTIFICATION DIVISION.
2 PROGRAM-ID. DDT.
3 ENVIRONMENT DIVISION.
4 CONFIGURATION SECTION.
5 SOURCE-COMPUTER. VAX-11.
6 OBJECT-COMPUTER. VAX-11.
7 INPUT-OUTPUT SECTION.
8 FILE-CONTROL.
9     SELECT IN-FILE ASSIGN TO "RS.DAT".
10    SELECT OUT-FILE ASSIGN TO "REPORT.LST".
11 DATA DIVISION.
12 FILE SECTION.
13 FD IN-FILE 'ABEL RECORD IS OMITTED DATA RECORD IS IN-REC.
14 01 IN-REC.
15     03 FILLER          PIC X(4).
16     03 NO-44           PIC 9(5).
17     03 NUMBER-44      PIC 999.
18     03 PRICE-44       PIC 99V99.
19     03 NAME-44        PIC X(20).
20     03 FILLER          PIC X(44).
21 FD OUT-FILE LABEL RECORD IS OMITTED DATA RECORD IS OUT-REC.
22 01 OUT-REC           PIC X(132).
23 WORKING-STORAGE SECTION.
24 77 EOF               PIC X          VALUE "N".
25 77 A-44              PIC 99999V99  VALUE ZERO.
26 01 H-HEAD.
27     03 FILLER          PIC X(50)   VALUE SPACE.
28     03 FILLER          PIC X(82)   VALUE "STOCK CALCULATION REPORT-
2 9 01 H-OUT.
30     03 FILLER          PIC X(20)   VALUE SPACE.
31     03 NO-44           PIC 99999.
32     0 3                FILPIC X(10) VALUE SPACE.
33     03 NAME-44        PIC X(20).
34     03 FILLER          PIC X(10)   VALUE SPACE.
35     03 NUMBER-44      PIC ZZ9.
36     03 FILLER          PIC X(10)   VALUE SPACE.
37     03 PRICE-44       PIC ZZ.99.
38     03 FILLER          PIC X(10)   VALUE SPACE.
39     03 TOT            PIC ZZ,ZZZ.99.
40     03 FILLER          PIC X(20)   VALUE SPACE.
41 PROCEDURE DIVISION.
42 P-00.
43     OPEN INPUT IN-FILE, OUTPUT OUT-FILE.
44     MOVE SPACE TO OUT-REC H-OUT.
45     WRITE OUT-REC FROM H-HEAD BEFORE 2 LINES.
46 P-11.
47     PERFORM P-33.
48     PERFORM P-44 UNTIL EOF = "Y".
49     CLOSE IN-FILE OUT-FILE.
50     STOP RUN.
51 P-33.
52     READ IN-FILE AT END MOVE "Y" TO EOF.
53 P-44.
54     COMPUTE A-44 = NUMBER-44 OF IN-REC * PRICE-C' OF IN-REC.
55     MOVE CORR IN-REC TO H-OUT.
56     MOVE A-44 TO TOT.
57     WRITE OUT-REC FROM H-OUT.
58     PERFORM P-33.

```


หมายเหตุ

1. ตัวอย่างโปรแกรมในหนังสือเล่มนี้ประมวลผลจากคอมพิวเตอร์ CDC 3100, IBM-370-138 และ VAX-11/750 ฉะนั้น ถ้านักศึกษาจะนำโปรแกรมแบบฝึกหัดไปประมวลผลกับคอมพิวเตอร์เครื่องอื่น อาจต้องเปลี่ยนแปลงรูปแบบบางส่วนในโปรแกรมบ้าง แต่รายละเอียดส่วนใหญ่ก็ยังคงเหมือนกัน
2. ลักษณะคำสั่งของภาษาโคบอลเหมือนกับประโยคภาษาอังกฤษทั่วไป, การใช้เครื่องหมายกำกับบรรทัดตอนการสะกดคำต้องถูกต้อง, ต้องถูกหลักไวยากรณ์ นักศึกษาควรต้องมีความระมัดระวังให้มาก เพื่อจะได้ประหยัดเวลาเครื่องคอมพิวเตอร์และจะได้ไม่ต้องนำโปรแกรมแบบฝึกหัดกลับไปแก้ไขบ่อย ๆ
3. ปัญหาในข้อ 1 นั้นแก้ไขได้โดยให้ดูจากรูปแบบ คำสั่ง กฎเกณฑ์ และข้อจำกัดต่างๆ ในหนังสือคู่มือโปรแกรมเมอร์ของคอมพิวเตอร์เครื่องที่นักศึกษานำไปประมวลผล