

(1) Identification Division

เป็นส่วนแรกของโปรแกรมภาษาโคบอล เพื่อบอกรายละเอียดของโปรแกรม เช่น ชื่อของโปรแกรม, ชื่อของโปรแกรมเมอร์ผู้เขียนโปรแกรม วันที่ที่เขียนโปรแกรม, วัตถุประสงค์ของโปรแกรม เป็นต้น

(2) Environment Division

เป็นส่วนที่ระบุถึงฮาร์ดแวร์ที่ใช้ เช่น ชนิดของคอมพิวเตอร์ที่ใช้, เครื่องพิมพ์, อุปกรณ์ภายนอกอื่น ๆ ที่ใช้ในการประมวลผลโปรแกรม

(3) Data Division

เป็นส่วนของการกำหนดองค์ประกอบและรูปแบบของข้อมูลทั้งหมด ที่ใช้ในการประมวลผลโปรแกรม

(4) Procedure Division

เป็นส่วนของคำสั่งภาษาโคบอล ซึ่งเป็นวิธีการในการประมวลผลข้อมูล ซึ่งวิธีการเขียนโปรแกรมจะคล้ายภาษา BASIC หรือ ภาษา FORTRAN คำสั่งหลักๆ ที่ใช้ สามารถแบ่งได้เป็น 4 พวก คือ

(4.1) คำสั่งในการรับ-ส่ง (Input/Output statements)

คำสั่ง OPEN ใช้สำหรับเตรียมเพิ่มข้อมูลเพื่ออ่าน หรือบันทึก

คำสั่ง CLOSE ใช้สำหรับจบการประมวลผลเพิ่มข้อมูล

คำสั่ง READ ใช้สำหรับอ่านระเบียบข้อมูล 1 ระเบียบจากเพิ่มข้อมูลที่กำหนด

คำสั่ง WRITE ใช้สำหรับบันทึกระเบียบข้อมูล 1 ระเบียบยังเพิ่มข้อมูลผลลัพธ์ที่เตรียมไว้

(4.2) คำสั่งเคลื่อนย้ายข้อมูล (Data movement statements)

คำสั่ง MOVE ใช้สำหรับย้ายข้อมูลจากที่หนึ่งไปยังอีกที่หนึ่ง

(4.3) คำสั่งคำนวณ (Arithmetic statements)

คำสั่ง COMPUTE ใช้สำหรับคำนวณค่าตามรูปแบบของสูตรคณิตศาสตร์

คำสั่ง ADD, SUBTRACT, MULTIPLY, DIVIDE เป็นคำสั่งที่ใช้ในการคำนวณ

ทั้งสิ้น

(4.4) คำสั่งควบคุม (Control statements)

คำสั่ง GOTO และ IF เป็นคำสั่งเงื่อนไขเพื่อควบคุมให้กระทำคำสั่งอื่นๆ

คำสั่ง PERFORM เป็นการควบคุมชั่วคราวเพื่อให้กระทำส่วนอื่นๆของโปรแกรม

คำสั่ง STOP เป็นคำสั่งหยุดการประมวลผลโปรแกรม

ตัวอย่างโปรแกรมภาษาโคบอล

เป็นโปรแกรมหาค่าเฉลี่ยเลขคณิตของคะแนนของนักศึกษา เหมือนกับตัวอย่างโปรแกรมภาษา BASIC จะเห็นว่าโปรแกรมภาษาโคบอล มีขนาดใหญ่มากกว่า แต่ก็ง่ายต่อการอ่าน และเข้าใจ เพราะคำสั่งเหมือนกับคำในภาษาอังกฤษ

FIGURE

9-7 Sample COBOL
program

```
1 IDENTIFICATION DIVISION.
2 PROGRAM ID SIX.
3 AUTHOR STEVE RUNDLELL.
4 ENVIRONMENT DIVISION
5 INPUT-OUTPUT SECTION.
6 FILE-CONTROL.
7 SELECT CARD-FILE ASSIGN TO SYSIPT.
8 SELECT PRINT-FILE ASSIGN TO SYSLST.
9 DATA DIVISION
10 FILE SECTION.
11 FD CARD-FILE
12 RECORDING MODE IS F
13 LABEL RECORDS ARE OMITTED
14 DATA RECORD IS CARD-IN
15 01 CARD-IN.
16 02 NAME-IN PIC X(40).
17 02 SCORE PIC 999.
18 02 FILLER PIC X(37).
19 FD PRINT-FILE
20 RECORDING MODE IS U
21 LABEL RECORDS ARE OMITTED
22 DATA RECORD IS PRINT-LINE
23 01 PRINT-LINE PIC X(133).
24 WORKING STORAGE SECTION.
25 77 STORE-NUMBER PIC 999 VALUE IS ZEROS.
26 77 STORE-SCORE PIC 99999 VALUE IS ZEROS.
27 01 PRINTER-LINE.
28 02 FILLER PIC X.
29 02 NAME-OUT PIC X(40).
30 02 FILLER PIC X(5).
31 02 SCORE-OUT PIC ZZ9.
32 02 FILLER PIC X(84).
33 01 AVERAGE-LINE.
34 02 FILLER PIC X(46).
35 02 AVERAGE PIC ZZ9.99.
36 02 FILLER PIC X(81).
37 PROCEDURE DIVISION.
38 OPEN FILES.
39 OPEN INPUT CARD-FILL.
40 OPEN OUTPUT PRINT-FILE.
41 READ-CARDS.
42 READ CARD-FILE AT END GO TO END-OF-JOB.
43 ADD SCORE TO STORE-SCORE. ADD 1 TO STORE-NUMBER
44 MOVE-DATA.
45 MOVE NAME-IN TO NAME-OUT.
46 MOVE SCORE TO SCORE-OUT. WRITE PRINT LINE
47 FROM PRINTER-LINE. LO To READ-CARDS.
'IA END-OF-JOB.
49 DIVIDE STORE-NUMBER INTO STORE-SCORE
50 GIVING AVERAGE-SCORE. MOVE AVERAGE-SCORE
51 TO AVERAGE. WRITE PRINT-LINE FROM AVERAGE-LINE
52 AFTER ADVANCING 2 LINES.
53 CLOSE CARD-FILL.
54 CLOSE PRINT-FILE.
55 STOP RUN.
```

9.2.3 ภาษาฟอร์แทรน (FORTRAN)

คำว่า FORTRAN ย่อมาจาก FORMula TRANslation ซึ่งได้พัฒนาขึ้นในปี ค.ศ. 1957 เป็นภาษาระดับสูงที่เก่าแก่มากที่สุด โดยออกแบบมาเพื่อใช้แก้ปัญหาทางคณิตศาสตร์และวิศวกรรม ซึ่งเป็นนิพจน์ทางพีชคณิต ไม่เหมาะกับงานประมวลผลข้อมูลทางธุรกิจ ซึ่งมีการเก็บข้อมูลบนแฟ้มข้อมูล ภาษานี้ได้พัฒนาต่อมาหลายรุ่น เช่น FORTRAN IV และ FORTRAN77 ซึ่งได้นำมาประยุกต์ใช้กับงานทางธุรกิจบ้างที่เกี่ยวกับการวิจัย การจัดการทางธุรกิจ โดยสร้างเป็นแบบจำลองทางธุรกิจ จากการวิเคราะห์ทางสถิติและคณิตศาสตร์

หลายๆ รุ่นของภาษานี้ ได้พัฒนาจุดเด่นๆ เพิ่มขึ้น เช่น WATFOR, WATFIV, XTRAN และ FASTRAN สามารถนำไปสอนที่โรงเรียน เป็นโปรแกรมในลักษณะของการโต้ตอบ (interactive) คล้ายๆ กับภาษาเบสิก ทำให้ภาษานี้มีมาตรฐาน 2 รูปแบบคือ FORTRAN และ BASIC FORTRAN ถูกกำหนดโดย American National Standards Institute in cooperation และผู้ผลิตคอมพิวเตอร์ โดย Basic FORTRAN จะเป็นรุ่นที่เป็นมาตรฐานโดยทั่วไป ที่ใช้ได้กับคอมพิวเตอร์โดยทั่วไป ส่วน FORTRAN จะเพิ่มจุดเด่นโดยบรรจุคำสั่งต่างๆ ที่ใช้งานที่มากขึ้น

คำสั่งที่ใช้ในภาษาฟอร์แทรน สามารถแบ่งออกได้เป็น 5 พวกคือ

(1) คำสั่งคำนวณ (Arithmetic statement)

เป็นคำสั่งที่ใช้ในการคำนวณ ซึ่งนิพจน์ของการคำนวณจะคล้ายกับนิพจน์ทางคณิตศาสตร์โดยทั่วไป

(2) คำสั่ง รับ-ส่งข้อมูล (Input/Output statement)

คำสั่งที่ใช้ คือ READ และ WRITE เป็นคำสั่งให้อ่านหรือบันทึกข้อมูลตามที่ได้ระบุไว้

(3) คำสั่งระบุรายละเอียด (Specification statements)

คำสั่ง FORMAT เป็นคำสั่งซึ่งระบุรูปแบบของข้อมูลเข้าหรือข้อมูลออก เช่น ขนาดเท่าใด ชนิดของข้อมูล หรือจำนวนข้อมูล เป็นต้น

คำสั่ง DIMENSION เป็นคำสั่งในการจองเนื้อที่ในหน่วยความจำ

(4) คำสั่งควบคุม (Control statements)

คำสั่ง GOTO และ IF เป็นคำสั่งเงื่อนไขเพื่อควบคุมให้กระทำคำสั่งอื่นๆ

คำสั่ง DO เป็นคำสั่งในการควบคุมการทำงานวนรอบ โดยกระทำตามคำสั่งที่กำหนดซ้ำๆ กัน

คำสั่ง STOP และ END เป็นคำสั่งให้จบโปรแกรม

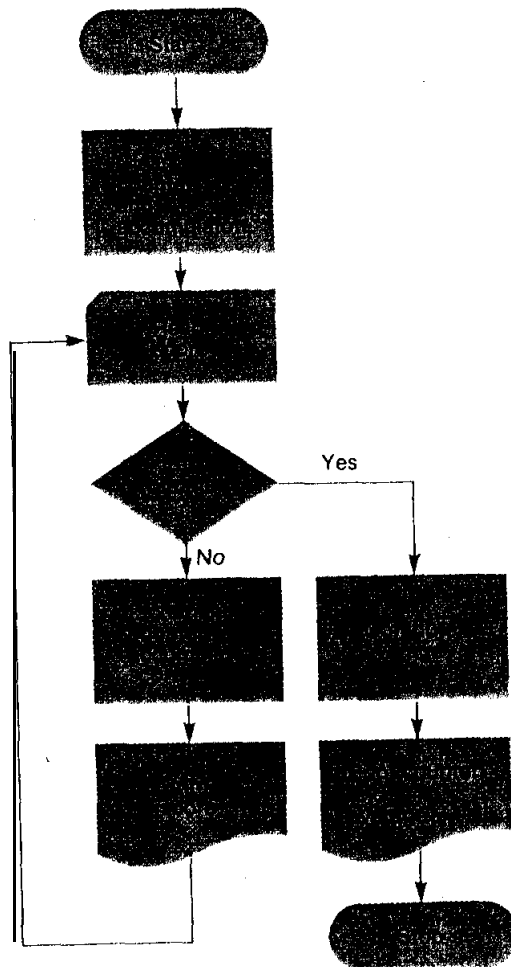
(5) คำสั่ง โปรแกรมย่อย (Subprogram statements)

กลุ่มของคำสั่งนี้ เช่น CALL, RETURN, FUNCTION และ SUBROUTINE ซึ่งเป็นคำสั่งซึ่งอนุญาตให้โปรแกรมเมอร์ สามารถใช้ โปรแกรม และสับรูทีนต่างๆ ที่เก็บใน subroutine library ได้

ตัวอย่างโปรแกรมภาษาฟอร์แทรน

FIGURE 9-8 Sample FORTRAN program

```
01 COUNTR = 0.0
02 TOTAL = 0.0
03 READ (1,04,END=09) SCORE
04 FORMAT (F6.2)
05 COUNTR = COUNTR+1.0
06 TOTAL = TOTAL+SCORE
07 WRITE (2,04) SCORE
08 GO TO 03
09 AVRAGE = TOTAL/COUNTR
10 WRITE (2,11) COUNTR,AVRAGE
11 FORMAT (1X,F10.2,F6.2)
12 STOP
13 END
```



9.2.4 ภาษา PL/I

PL/I ย่อมาจาก Programming Language I ซึ่งพัฒนาขึ้นโดย IBM ในปี ค.ศ. 1965 เป็นภาษาที่พัฒนาขึ้นมาเพื่อให้สามารถแก้ปัญหาได้ตามวัตถุประสงค์ต่างๆ ไป ทั้งทางด้านธุรกิจและวิทยาศาสตร์ โดยรวมเอาจุดเด่นของภาษาฟอร์แทรนและภาษาโคบอล รวมเข้าด้วยกัน และผนวกกับความสามารถบางอย่างของภาษาแอสแซมบลี และ ALGOL ผสมเข้าไปอีกด้วย ก็น่าที่จะทำให้ภาษานี้ได้รับความนิยม แต่กลับเป็นว่า ภาษานี้ไม่ได้ถูกนำมาใช้โดยกว้างขวางเท่าภาษา FORTRAN และภาษาโคบอล อาจเป็นเพราะผู้ผลิตซอฟต์แวร์พัฒนาตัวแปลภาษา PL/I ซ้ำมาก แต่ในเวลาต่อมาก็เป็นที่ยอมรับจากผู้ใช้งานมากขึ้น

ภาษา PL/I นี้ได้รับการวิพากษ์วิจารณ์ว่าเป็นภาษาที่ยากต่อการเรียนรู้และขาดประสิทธิภาพในการนำมาเขียนโปรแกรม แต่ข้อดีคือเป็นภาษาที่ใช้ได้ตามวัตถุประสงค์ที่ต้องการ โดยมีความยืดหยุ่นสูง การออกแบบเป็นโมดูล (modular) ซึ่งมีลักษณะเป็นโปรแกรมโครงสร้าง (structured programming) มากกว่าภาษาอื่นๆ สามารถแก้ปัญหาทางด้านธุรกิจ วิทยาศาสตร์ โปรแกรมระบบ และ realtime ได้

ภาษา PL/I ประกอบด้วยโมดูล หรือ subsets ซึ่งแต่ละโมดูลจะเป็นงานประยุกต์ที่แตกต่างกัน จุดเด่นอีกประการคือ PL/I จะอนุญาตให้โปรแกรมเมอร์ เพิกเฉย (ignore) บางโมดูลที่ไม่ถูกใช้งานได้ ซึ่งตัวแปลภาษาจะทราบโดยอัตโนมัติ เป็นการแปลภาษาที่เรียกว่า default interpretation

คำสั่ง PL/I

คำสั่งในภาษา PL/I จะประกอบด้วยตัวอักษรที่เรียงติดต่อกันไป โดยจบด้วยเครื่องหมายเซมิโคลอน (;) โปรแกรมจะประกอบด้วยคำสั่งหลายๆ คำสั่งซึ่งจัดเป็นกลุ่มๆ เรียกว่า blocks หรือ groups ซึ่งแต่ละบล็อก จะมีวิธีการทำงาน (procedure) ในโปรแกรมหนึ่งอาจประกอบด้วยหนึ่งหรือหลายบล็อก ในแต่ละบล็อก อาจประกอบด้วย หนึ่งหรือมากกว่าหนึ่ง procedure ก็ได้

คำสั่งภาษา PL/I สามารถแบ่งได้เป็น 5 กลุ่ม คือ

(1) คำสั่งกำหนดค่า หรือ คำสั่งคำนวณ (Assignment or arithmetic statements)

เป็นคำสั่งกำหนดค่า และคำสั่งที่ใช้ในการคำนวณ และเปรียบเทียบ

(2) คำสั่ง รับ-ส่งข้อมูล (Input/Output statements)

เป็นคำสั่งในการเคลื่อนย้ายข้อมูลระหว่าง CPU กับอุปกรณ์ในการรับ-ส่งข้อมูล

คำสั่ง READ หรือ WRITE เป็นคำสั่งในการรับหรือบันทึกข้อมูล ระเบียบข้อมูลใน

แฟ้มข้อมูล

คำสั่ง GET และ PUT เป็นคำสั่งในการรับหรือบันทึกข้อมูลในรูปแบบของ 'free form' เป็นสายของตัวอักษร (stream)

(3) คำสั่งควบคุม (Control statements)

เป็นคำสั่งที่ใช้ในการควบคุมการทำงานของลำดับของคำสั่ง อาจเกิดจากการเปรียบเทียบค่า และทดสอบเงื่อนไขต่างๆ ผลลัพธ์ที่ได้ จะเป็นตัวควบคุมการทำงานให้กระโดดผ่านไปปฏิบัติงานต่างๆ ซึ่งอาจเป็นการทำงานซ้ำๆ กันได้ คำสั่งเหล่านี้ เช่น DO, GOTO และ IF-THEN

(4) คำสั่งประกาศข้อมูล (Data declaration statements)

เป็นคำสั่งในการระบุชนิดและรูปแบบของตัวแปรข้อมูลที่ใช้ในโปรแกรม คำสั่งที่ใช้คือ DECLARE การระบุนี้ อาจระบุเป็นระเบียนข้อมูล (record) หรือ ข้อมูลชนิดแถว (array) ก็ได้

(5) คำสั่งโครงสร้างโปรแกรม (Program structure statements)

เป็นคำสั่งในการกำหนดและระบุชนิดของส่วนโปรแกรม (program segments) ที่ใช้รวมทั้งการกำหนด ลาดเบล (label) ด้วย การกำหนดโดย คำสั่ง procedure และจบด้วยคำสั่ง END

ตัวอย่างโปรแกรมภาษา PI/I

```
AVERAGE: PROCEDURE OPTIONS (MAIN);
/* PROGRAM FOR AVERAGE OF STUDENT SCORES */
/* END OF SCORES IS INDICATED BY A NEGATIVE NUMBER */
DECLARE (AVERAGE, TOTAL, VALUE, SCORE, COUNTER)
FIXED DECIMAL;
COUNTER = 0;
TOTAL = 0;
GET LIST (SCORE);
DO WHILE (SCORE >= 0)
COUNTER = COUNTER + 1;
TOTAL = TOTAL + SCORE;
GET LIST (SCORE);
END;
AVERAGE = TOTAL / SCORE;
PUT LIST (AVERAGE SCORE IS / AVERAGE);
END AVERAGE;
```

9.2.5 ภาษาปาสคาล (PASCAL)

ภาษาปาสคาลถูกพัฒนาขึ้นในปี ค.ศ. 1960 โดย Professor Niklaus Wirth ของ Zurich เพื่อให้เป็นภาษาซึ่งใช้สำหรับสอนให้รู้จักการเขียน โปรแกรมที่เป็นโครงสร้าง และการออกแบบ โปรแกรมจากบนลงล่าง ชื่อ PASCAL นั้นตั้งเพื่อเป็นเกียรติแก่นักคณิตศาสตร์และปรัชญา ชื่อ Blaise Pascal ผู้ซึ่งผลิตเครื่องคำนวณ ภาษาปาสคาลเป็นภาษาที่ง่ายต่อการเรียนรู้ และสามารถ ใช้ได้กับเครื่องไมโครคอมพิวเตอร์ได้

จุดเด่นๆ ของภาษาคือ ตัวแปรต่างๆ ที่ใช้ในโปรแกรมต้องมีการกำหนดชนิดของข้อมูล (type) ข้อมูลที่ส่งผ่านจากโมดูลหนึ่งไปยังอีกโมดูลหนึ่งต้องมีชนิดเดียวกัน กรณีที่การส่งผ่าน หรือค่าของข้อมูลที่ใช้ในโปรแกรม มีค่าผิดชนิด เมื่อทำการแปลภาษา จะมีข้อมูลความผิดพลาด ปรากฏให้ผู้เขียนโปรแกรมทราบ

คำสั่งต่างๆ ในภาษาปาสคาลประกอบด้วยตัวอักษรเรียงต่อกัน โดยสิ้นสุดคำสั่งด้วย เครื่องหมายเซมิโคลอน (semicolons) และแต่ละคำสั่งเรียงต่อกัน โดยสิ้นสุดโปรแกรมด้วย จุด (period) ตัวแปรทั้งหมดที่ใช้ในโปรแกรมรวมทั้งค่าคงที่ด้วยจะต้องถูกกำหนดและประกาศ ชนิดของตัวแปรในส่วนที่เรียกว่า declaration นอกจากนี้ ในส่วนนี้เป็นส่วนกำหนดโพรซีเจอร์ (procedures) และฟังก์ชัน (function) ที่ผู้ใช้กำหนดขึ้นมาด้วย

คำสั่งที่ใช้ในการปฏิบัติงาน สามารถแบ่งออกได้ 5 กลุ่มคือ

(1) คำสั่งกำหนดค่า (Assignment statements)

เป็นคำสั่งใช้สำหรับกำหนดค่าให้กับตัวแปร หรือ ผลลัพธ์จากการทำงาน

(2) คำสั่ง รับ-ส่งข้อมูล (Input-Output statements)

เป็นคำสั่งที่ผู้ใช้ใช้สำหรับย้ายข้อมูลจากที่หนึ่งไปยังอีกที่หนึ่ง

(3) คำสั่งกลุ่ม (Compound statements)

เป็นลำดับคำสั่ง ซึ่งอยู่ภายในระหว่าง begin และ end

(4) คำสั่งเงื่อนไข (Conditional statements)

เป็นการเลือกการทำงาน โดยขึ้นอยู่กับค่าของนิพจน์ คำสั่งในกลุ่มนี้ ได้แก่

IF-THEN, IF-THEN-ELSE, CASE

(5) คำสั่งกระทำซ้ำ (Repetitive statements)

เป็นคำสั่งซึ่งกระทำงานหรือคำสั่งใดๆ ซ้ำๆ กัน จนกระทั่งเงื่อนไขเป็นจริงแล้วจึงหยุดกระทำซ้ำ คำสั่งในกลุ่มนี้ได้แก่ WHILE-DO, REPEAT-UNTIL และ FOR

(6) คำสั่ง GOTO

เป็นคำสั่งที่มีการกระโดดการทำงานไปยังคำสั่งที่ต้องการ แต่การเขียนโปรแกรมที่มีการใช้คำสั่งนี้ ถือว่า ไม่ดี (poor) เพราะเป็นการเขียนโปรแกรมที่ไม่เป็นโครงสร้าง ตัวอย่างการเขียนโปรแกรมภาษาปาสคาล

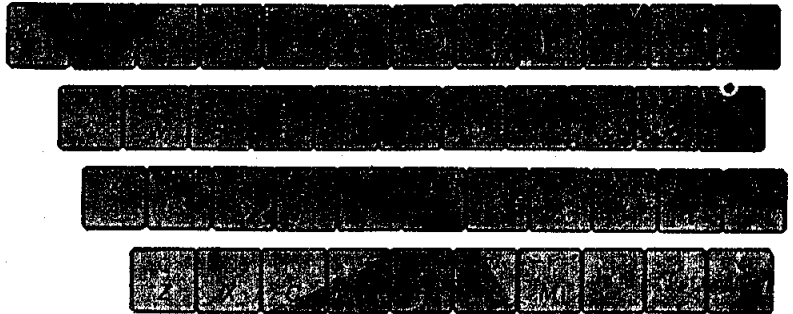
```
PROGRAM averagescore (infile,outfile);
VAR score, sum, average, count : real;
    infile, outfile : text;
BEGIN
    sum:=0.0; count:=0.0;
    REPEAT
        read(infile, score);
        sum:=sum + score;
        count:=count + 1.0
    UNTIL eof(infile);
    average:=sum/count;
    write (outfile, ' Average score is', average)
END
```

9.2.6 ภาษา APL

คำสั่ง APL ย่อมาจาก A Programming Language เป็นภาษาโต้ตอบ ซึ่งสร้างโดย Kenneth Iverson ของ IBM โดยออกแบบให้เป็นภาษา problem-oriented ซึ่งแทนการปฏิบัติงาน หรือ ฟังก์ชันด้วยสัญลักษณ์ ซึ่งง่ายต่อการเขียนโปรแกรม ตัวอย่างการใช้กลุ่มอักขระพิเศษ ดังรูปที่ 9-11

FIGURE

9-1 1 A typical APL character set and keyboard



ข้อจำกัดในการใช้โปรแกรม APL คือ แป้นพิมพ์ที่ใช้ต้องมีอักขระ APL เท่านั้น เนื่องจากอักขระต่างๆ เหล่านี้แทนการปฏิบัติงานต่างๆ ในโปรแกรม ซึ่งมีผลให้มีการใช้ตัวแปรและคำสั่งมีจำนวนน้อย โปรแกรมเมอร์ไม่ต้องระบุนายละเอียด วิธีการต่างๆ เพื่อควบคุม หรือ ใช้คำสั่งกระทำซ้ำๆ เหมือนภาษาระดับสูงอื่นๆ ในปี ค.ศ. 1968 บริษัท IBM ได้ขายตัวแปลภาษาให้กับบริษัทอื่นๆ เช่น Digital Equipment Corporation ทำให้สามารถใช้กับระบบคอมพิวเตอร์อื่นๆ ได้ ซึ่งต่อมา ภาษา APL เป็นภาษาที่ได้รับความนิยม โดยเฉพาะในระบบแบ่งเวลา (time-sharing systems) ของข่ายงานคอมพิวเตอร์ ถึงแม้ภาษา APL เป็นโปรแกรมที่อ่านยาก และต้องการหน่วยความจำขนาดใหญ่ แต่ก็ไม่ยากที่จะเรียนรู้การใช้

คำสั่ง APL

ผู้เขียนโปรแกรมสามารถใช้คำสั่ง APL ในการแก้ปัญหาต่างๆ ในโปรแกรมมากมาย อย่างเช่น ตัวอย่างในรูปที่ 9-12

FIGURE

9-12 Examples of APL statements

APL Statement

A ← □

A + B

A ⌈ B

A - 1 2 3

3 2 ρ A

Function

Provide input data for A

A plus B

Find the larger of A and B

Assign this list (vector) of numbers to A

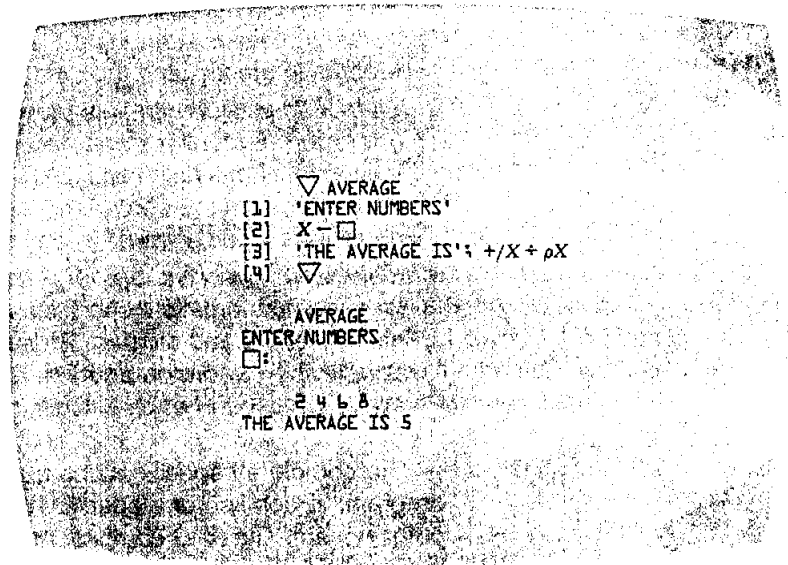
Create a 3 × 2 matrix or two-dimensional array from the vector A

ตัวอย่างโปรแกรม APL

ตัวอย่างตามรูปภาพ 9-13 นี้ เป็นโปรแกรมในการหาค่าเฉลี่ยของกลุ่มตัวเลขที่ผู้ใช้ป้อนทางคีย์บอร์ด

FIGURE

9-13 Sample APL program



92.7 ภาษา RPG

คำว่า RPG ย่อมาจาก Report Program Generator เป็นภาษาที่พัฒนาขึ้นมาเพื่อช่วยแก้ปัญหาในการผลิตรายงาน (reports) เป็น problem oriented language นอกจากนี้ เป็น procedure-oriented กล่าวคือ เป็นภาษาที่ไม่ต้องกำหนดวิธีการทำงานโดยละเอียด ผู้ใช้เพียงกำหนดรูปแบบของข้อมูล อุปกรณ์ในการรับ-ส่ง แฟ้มข้อมูลที่ใช้ รูปแบบของรายงาน ผลลัพธ์ การคำนวณที่ต้องการ ตัวแปลภาษา RPG จะทำการสร้างโปรแกรมภาษาเครื่องที่ผู้ใช้ต้องการพร้อมกับผลิตรายงานตามที่กำหนดให้ได้อย่างถูกต้อง

การเขียนโปรแกรมภาษา RPG จะเป็นการกำหนดข้อมูลตามฟอร์มต่างๆ ซึ่งแบ่งได้เป็น 4 ฟอร์ม คือ

(1) File description specifications

เป็นฟอร์มในการกำหนดแฟ้มข้อมูลที่ใช้ เช่น แฟ้มข้อมูลเข้า, แฟ้มข้อมูลออก, คุณลักษณะของแฟ้มที่ใช้ รวมทั้งอุปกรณ์ในการรับ-ส่งข้อมูล

(2) Input specifications

เป็นฟอร์มที่ใช้ระบุถึงรูปแบบของระเบียบข้อมูลที่ใช้ของแฟ้มข้อมูลเข้า

(3) Output specifications

เป็นฟอร์มที่ใช้ระบุถึงรูปแบบของรายงาน ซึ่งกำหนดรายละเอียดของรายการข้อมูล (field) ของระเบียบข้อมูล รวมทั้งการระบุถึงข้อความต่างๆ ที่ต้องการพิมพ์ เช่น ข้อความในส่วนของหัวตาราง หรือ ไตเติล เป็นต้น นอกจากนี้ ผู้ใช้สามารถระบุถึงเงื่อนไขต่างๆ ในการพิมพ์รายงาน ได้อีกด้วย

(4) Calculation specifications

เป็นฟอร์มที่ใช้สำหรับการคำนวณในโปรแกรม

ซึ่งฟอร์มต่างๆ ที่กล่าวมานี้ ผู้เขียนโปรแกรมต้องระบุให้สมบูรณ์ ซึ่งฟอร์มต่างๆ เหล่านี้จะแทนคำสั่งในภาษา RPG ซึ่งตัวแปลภาษา RPG จะแปลเป็น Object program ซึ่งสามารถประมวลผลได้ตามวิธีการที่ระบุในฟอร์มต่างๆ

ตัวอย่างภาษา RPG

เป็นตัวอย่างในการคำนวณค่าเฉลี่ยของคะแนนสอบนักศึกษา

REPORT PROGRAM GENERATOR FILE DESCRIPTION SPECIFICATIONS
IBM System/360

File: _____ Page: 1 of 1

File Name	File Type	File Organization	File Format	File Code	File Path	File Attributes
SCORES	SEQ	FB	130	130	READD1 S7S RDR	
PRINT	SEQ	FB	130	130	PRINTER S7S LST	

RPG INPUT SPECIFICATIONS

File: _____ Page: 1 of 1

Field Name	Field Position	Field Length	Field Data	Field Indicator
SCORES	1	10		
PRINT	1	10		

RPG CALCULATION SPECIFICATIONS

File: _____ Page: 1 of 1

Field Name	Field Position	Field Length	Field Data	Field Indicator	Comments
SCORES	1	10			
PRINT	1	10			

RPG OUTPUT FORMAT SPECIFICATIONS

File: _____ Page: 1 of 1

Field Name	Field Position	Field Length	Field Data	Field Indicator	Comments
PRINT	1	10			
SCORES	1	10			
AVRAGE	1	10			

การทำงานของโปรแกรม RPG มีขั้นตอนดังนี้

(1) อ่านคะแนน (SCORE) ของนักศึกษา

(2) นำคะแนนของนักศึกษาทุกคนรวมที่ TOTAL และนับจำนวนนักศึกษาเก็บไว้ที่
COUNTR

(3) คำนวณหาค่าเฉลี่ย (AVRAGE) โดยนำ TOTAL หารด้วย COUNTR

(4) พิมพ์คะแนนเฉลี่ย และ หัวตารางที่กำหนด

สรุปท้ายบท

1. ภาษาโปรแกรม (Programming Languages) เป็นซอฟต์แวร์ที่ทำให้คอมพิวเตอร์และคนสามารถสื่อสารกันได้ สามารถแบ่งได้ 4 ระดับชั้น คือ machine languages, assembler languages, high-level languages และ fourth-generation languages ซึ่งภาษาโปรแกรมต่างๆ เหล่านี้ต้องแปลเป็นคำสั่งภาษาเครื่อง โดย compilers หรือ assemblers หรือ interpreter
2. ภาษา BASIC (Beginner's All-purpose Symbolic Instruction Code) เป็นภาษาที่ใช้สำหรับธุรกิจเล็กๆ ของระบบแบ่งปันเวลา (time-sharing) ในระบบที่ใช้เครื่องไมโครคอมพิวเตอร์
3. ภาษา COBOL (Common Business Oriented Language) เป็นภาษาที่ใช้ในการประมวลผลข้อมูลทางธุรกิจ ซึ่งคล้ายกับภาษาอังกฤษ สามารถประมวลผลข้อมูลจากแฟ้มข้อมูลที่มีขนาดใหญ่ได้หลายแฟ้ม
4. ภาษา FORTRAN (FORmula TRANslation) เป็นภาษาที่นิยมใช้ในการประมวลผลงานทางด้านวิทยาศาสตร์ และวิศวกรรมศาสตร์ ซึ่งเกี่ยวข้องกับทางด้านกรคำนวณทางคณิตศาสตร์
5. ภาษา PL/I (Programming Language I) เป็นภาษาที่ออกแบบให้ใช้ได้กับงานประยุกต์โดยทั่วไป ทั้งงานประยุกต์ทางธุรกิจและวิทยาศาสตร์ ภาษานี้ดีกว่าภาษาอื่นๆ ที่ว่าเป็น modular language ซึ่งเป็นวิธีการเขียนโปรแกรมที่เป็นโครงสร้าง
6. ภาษา Pascal เป็นภาษาที่ออกแบบเป็น top-down ซึ่งเป็นแนวความคิดของการเขียนโปรแกรมที่เป็นโครงสร้าง เป็นภาษาที่นิยมใช้โดยทั่วไป
7. ภาษา APL (A Programming Language) เป็นภาษาที่ออกแบบเพื่อแก้ปัญหาแบบโต้ตอบ ซึ่งเป็น problem-oriented programming โดยใช้สัญลักษณ์ต่างๆ เพื่อแทนการทำงานที่ต้องการนิยมใช้ในงานประยุกต์แบ่งปันเวลา (time-sharing)
8. โปรแกรม RPG (Report Program Generator) เป็นภาษาที่เป็น problem-oriented โดยแก้ปัญหาทางด้านกรผลิตรายงานทางธุรกิจโดยเฉพาะ ซึ่งระบุการทำงานที่ต้องการตามฟอร์มต่างๆ ของภาษา

คำศัพท์ที่สำคัญ

Machine language	Subroutine
Assembler language	BASIC
High-level language	COBOL
Fourth-Generation language	FORTRAN
Natural language	PL/I
Procedural language	Pascal
Nonprocedural language	APL
Problem-oriented language	RPG
Macro instruction	

คำถามท้ายบท

1. ภาษาโปรแกรมของระดับชั้นต่างๆ แตกต่างกันอย่างไร และระดับใดที่ง่ายต่อการเข้าใจและเขียนโปรแกรม
 2. macro instruction คืออะไร แตกต่างจาก subroutine อย่างไร
 3. อะไรคือความแตกต่างระหว่าง procedural และ nonprocedural language
 4. fourth generation language คืออะไร และใช้อย่างไร
 5. ประโยชน์ และข้อจำกัดของภาษาระดับสูง มีอะไรบ้าง
 6. การใช้ natural conversational language เป็นอย่างไร จงอธิบาย
 7. อะไรคือคุณลักษณะเด่นๆ ของภาษาต่อไปนี้ BASIC, COBOL, FORTRAN, PL/I, APL, RPG และ PASCAL
 8. คำสั่ง APL แตกต่างจาก conventional programming languages อย่างไร
-