

2) ถ้ามี UPON, mnemonic-name ต้องเป็นชื่อที่เรากำหนดให้กับเครื่องพิมพ์ดีดหรือ output devices อื่น ๆ ในพารากราฟ special-names ใน Environment division ขนาดของตัวอักษรที่จะให้พิมพ์ออกมานั้นขึ้นอยู่กับอุปกรณ์ที่ใช้

ถ้าเป็นเทปแม่เหล็ก หรือ เครื่องพิมพ์ ขนาดสูงสุดของข้อมูลที่จะให้พิมพ์เท่ากับ 120 ตัวอักษร

ถ้าเป็นเครื่องพิมพ์ดีด, เครื่องเจาะ หรือเอ้าท์พุท devices อื่น ๆ นอกเหนือจากที่กล่าวนี้ ขนาดสูงสุดของข้อมูลที่จะให้พิมพ์เท่ากับ 80 ตัวอักษร

สำหรับตำแหน่งที่เหล้อยู่ใน 1 บรรทัด หรือในบัตร เมื่อย้ายข้อมูลหมดแล้วเครื่องก็จะอ่านเรคคอร์ดถัดไป

หมายเหตุ *

ถ้าให้พิมพ์ออกมาทางเครื่องพิมพ์, ตัวอักษรตัวแรกของเอ้าท์พุทเรคคอร์ดเครื่องจะไม่พิมพ์ให้ เพราะถูกนำไปใช้เป็น carriage control สิ่งที่โปรแกรมเมอร์ควรจะต้องทำคือกำหนดให้ตัวอักษรตัวแรกในคำสั่ง display เป็น literal หรือ data-name ที่จะเป็นตัวควบคุมเกี่ยวกับการพิมพ์

data-name ใด ๆ ในคำสั่งนี้อาจจะเป็นชื่อของข้อมูลกลุ่ม หรือข้อมูลเดี่ยวก็ได้ อาจจะมี subscripted ก็ได้ แต่ต้องไม่ใช่ usage COMPUTATION-1

literal ในที่นี้จะ เป็น numeric หรือ non-numeric literal ได้

ตัวอย่าง

DISPLAY AMOUNT-5, "IS A VALUE OUT OF RANGE" UPON CONSOLE-TYPE.

เมื่อเครื่อง execute คำสั่งนี้ มูลค่าของฟิลด์ AMOUNT-A ในหน่วยความจำจะถูกพิมพ์ออกมาทางเครื่องพิมพ์ดีดบนคอนโซล/เทอร์มินัล ตามด้วย literal ในเครื่องหมายคำพูด

ตัวอย่าง

DISPLAY DETAILS.

DISPLAY "JULY, 1987".

DISPLAY QUOTE "JULY, 1987" QUOTE.

* เฉพาะเครื่อง CDC เท่านั้น

คำสั่งแรก พิมพ์มูลค่าใน DETAILS
คำสั่งที่สอง พิมพ์ JULY, 1987
คำสั่งที่สาม พิมพ์ "JULY, 1987"

ทั้ง 3 คำสั่งนี้ พิมพ์ออกมาทางเครื่องพิมพ์ดีดบนคอนโซล หรือทางจอภาพของเทอร์มินัล

ตัวอย่าง 1

SPECIAL-NAMES.

SYSTEM-OUTPUT-TAPE IS OUT.

PROCEDURE DIVISION.

FIRST-PARA.

DISPLAY "MASTER-COUNT = " CNT-M UPON OUT.

ถ้ามูลค่าใน CNT-M เท่ากับ 18200 เครื่องจะพิมพ์ออกมาทางเครื่องพิมพ์ ดังนี้

MASTER-COUNT = IX200

ตัวอย่าง 2

DISPLAY PRINT-REC UPON OUT.

DISPLAY "COMPLETELY" UPON OUT.

DISPLAY QUOTE "COMPLETELY" QUOTE UPON OUT.

คำสั่งแรก พิมพ์มูลค่าของ PRINT-REC

คำสั่งที่สอง พิมพ์ COMPLETELY

คำสั่งที่สาม พิมพ์ "COMPLETELY"

ทั้ง 3 คำสั่งนี้พิมพ์ออกทางเครื่องพิมพ์

คำสั่ง ACCEPT

คำสั่งนี้สั่งให้นำข้อมูลจากเครื่องพิมพ์ดีดหรือเทอร์มินัล หรือ input device อื่น ๆ เข้าไป execute ในเครื่องคอมพิวเตอร์ และระหว่าง object time นั้น ที่เครื่องพิมพ์ดีดบนคอนโซล จะพิมพ์คำว่า “ACCEPT data-name” ออกมาด้วย

มีรูปแบบดังนี้

ACCEPT data-name [**FROM** mnemonic-name]

1) ถ้าไม่มีคำว่า FROM ข้อมูลจะได้จากเครื่องพิมพ์ดีดหรือเทอร์มินัลโดยการพิมพ์เข้าไปแล้วเก็บแบบ left justified ใน receiving area ซึ่งกำหนดชื่อโดย data-name ถ้าจำนวนตัวอักษรที่พิมพ์เข้าไปน้อยกว่าขนาดของ receiving area ส่วนที่เหลือมันจะได้ blank ให้ แต่ถ้าพิมพ์ตัวอักษรเข้าไปมากกว่าขนาดของเนื้อที่ ที่กำหนดให้ ตัวอักษรส่วนเกินเครื่องจะไม่รับ

2) ถ้ามีคำว่า FROM mnemonic-name หมายถึงชื่อที่เรากำหนดให้เท่ากับ input device ในพารากราฟ special-names ใน Environment division และถ้า mnemonic-name หมายถึงเครื่องพิมพ์ดีดจะได้ผลลัพธ์เหมือนกับแบบไม่มีคำว่า FROM แต่ถ้า mnemonic-name หมายถึง input device อื่น ๆ ในขณะ execut คำสั่งนี้ เครื่องจะอ่าน 1 logical record จาก input device นั้น แล้วเก็บไว้ในลักษณะที่กล่าวมาแล้วข้างต้นโดยมีขนาดสูงสุดของเรคคอร์ดเท่ากับ 80 ตัวอักษร

data-name ที่กำหนดในคำสั่งนี้ อาจเป็นชื่อของข้อมูลกลุ่ม หรือ ชื่อของข้อมูลเดี่ยวก็ได้ อาจจะมี subscripted ก็ได้ แต่จะใช้ usage COMPUTATIONAL-1 ไม่ได้

ตัวอย่าง

WORKING-STORAGE SECTION.

01 WHAT-TO-DO.

02 REPORT-TYPE PICTURE 9

88 REPORT-1 VALUE 1.

X8 REPORT-2 VALUE 2.

```

88 REPORT-3 VALUE 3.
02 CURRENT-DATE PICTURE 9(6).
02 FILLER PICTURE X(73).
PROCEDURE DIVISION.
GET-CONTROL-CARD.
ACCEPT WHAT-TO-DO.
IF REPORT-1 PERFORM FIRST-REPORT
ELSE
    IF REPORT-2 PERFORM SECOND-REPORT
    ELSE
        IF REPORT-3 PERFORM THIRD-REPORT
        ELSE
            PERFORM ERROR-ROUTINE.

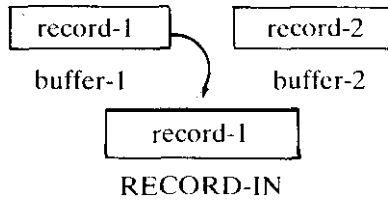
```

บัฟเฟอร์ (Buffers)

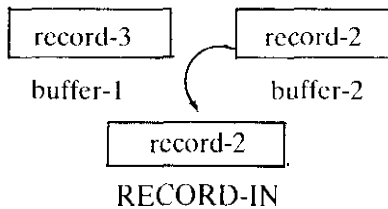
บัฟเฟอร์ หมายถึง เนื้อที่ในหน่วยความจำหลักของคอมพิวเตอร์สำรองไว้สำหรับเก็บ อินพุทหรือเอาต์พุทเรคคอร์ด การใช้บัฟเฟอร์ทำให้คำสั่ง READ และคำสั่ง WRITE มีประสิทธิภาพสูงกว่าคำสั่ง ACCEPT และคำสั่ง DISPLAY ในกรณีที่เราไม่ได้กำหนดข้อจำกัดของบัฟเฟอร์ของโปรแกรมคอมพิวเตอร์ไว้ ส่วนใหญ่จะถือว่าแฟ้มข้อมูลแต่ละชุดจะมีบัฟเฟอร์ใช้ 2 แห่งดังนี้

อินพุทบัฟเฟอร์ (Input buffers)

ตัวอย่างเช่น เราตั้งชื่อเนื้อที่อินพุทว่า RECORD-IN ซึ่งมี บัฟเฟอร์ 2 ชุด คือ buffer-1 และ buffer-2 เมื่อโปรแกรมอ่านอินพุทเรคคอร์ดแรกเข้ามา เครื่องจะส่งข้อมูลนี้จากอินพุท device ไปยัง buffer-1 ในหน่วยความจำหลัก จากนั้นจึงย้ายเรคคอร์ดจาก buffer-1 ไปไว้ในเนื้อที่อินพุทของโปรแกรม (RECORD-IN) และขณะที่โปรแกรมกำลังประมวลผล (process) อินพุทเรคคอร์ดแรกนั้น เครื่องก็จะอ่านเรคคอร์ดถัดไป แล้วส่งอินพุทเรคคอร์ดที่สองไปไว้ใน buffer-2 ดังรูปข้างล่างนี้



ขณะนี้โปรแกรมอ่านเรคคอร์ดที่สองและเก็บไว้ในหน่วยความจำหลักเรียบร้อยแล้ว (ใน buffer-2) และต่อไปเครื่องก็จะส่งเรคคอร์ดนี้ไปยังเนื้อที่ RECORD-IN อีกครั้งหนึ่ง ระหว่างที่โปรแกรมกำลังประมวลผลเรคคอร์ดที่สอง, เครื่องจะอ่านเรคคอร์ดถัดไป แล้วส่งอินพุทเรคคอร์ดที่สามจากอินพุท device ไปยัง buffer-1 ดังรูป



โดยวิธีการนี้การมีบัฟเฟอร์จึงทำให้การทำงานคาบเกี่ยวกันระหว่างการอ่านและการประมวลผลสะดวกสบายขึ้น

เข้าที่พุทบัฟเฟอร์ (Output buffers)

ในทำนองเดียวกัน การพิมพ์ (writing) ก็มีการทำงานคาบเกี่ยวกับการประมวลผลด้วย เมื่อโปรแกรมย้ายเข้าที่พุทเรคคอร์ดไปยังเข้าที่พุทบัฟเฟอร์โปรแกรมก็สามารถทำการประมวลผลต่อไปได้ โดยไม่ต้องคอยให้การพิมพ์นั้นเกิดขึ้นจริง ในตอนท้ายของการ execute โปรแกรมคำสั่ง CLOSE จะพิมพ์มูลค่าของเรคคอร์ดสุดท้ายที่เก็บอยู่ในบัฟเฟอร์

เมื่อเราใช้คำสั่ง DISPLAY และคำสั่ง WRITE บนเครื่องพิมพ์ (printer) ตัวเดียวกัน ให้ระวังด้วยว่าถ้าโปรแกรมนั้นพิมพ์ (write) ผลรวมสุดท้ายแล้วมีคำสั่ง DISPLAY "END OF REPORT" ข้อความที่จะเกิดจากคำสั่ง DISPLAY บางทีอาจจะพิมพ์รูปแบบก่อนพิมพ์ผลรวมสุดท้าย เพราะว่าคำสั่ง DISPLAY ไม่มีการใช้บัฟเฟอร์ตัวที่สอง มันจึงพิมพ์ออกมาทันที แต่สำหรับคำสั่ง WRITE เครื่องจะรอ เรคคอร์ดที่จะพิมพ์ก่อน

สรุป ในกรณีที่โปรแกรมเมอร์ต้องการจะพิมพ์และ display ผลรวมสุดท้าย ให้ใช้คำสั่ง CLOSE print ไฟล์ก่อน แล้วจึง display ข้อความที่ต้องการ

6.2 คำสั่งที่มีเงื่อนไข (Conditional statements)

คำสั่งเหล่านี้เป็นการกำหนด operations ที่จะให้เครื่องคอมพิวเตอร์ executed หรือ bypassed ทั้งนี้ขึ้นอยู่กับเงื่อนไขที่เราจะยึดเป็นหลักกันนั้น ๆ ว่าจริงหรือไม่จริง โดยมีรูปแบบดังนี้

$$\text{IF conditional-expression } \left\{ \begin{array}{l} \text{statement-1} \\ \underline{\text{NEXT SENTENCE}} \end{array} \right\} \left\{ \begin{array}{l} \underline{\text{OTHERWISE}} \\ \underline{\text{ELSE}} \end{array} \right\}$$
$$\left\{ \begin{array}{l} \text{statement-2} \\ \underline{\text{NEXT SENTENCE}} \end{array} \right\}$$

conditional expression เป็น combination ของ literals และ data-name เกี่ยวข้องกันโดยกำหนดความสัมพันธ์ให้ expression นี้ จะจริงหรือไม่จริง ขึ้นอยู่กับมูลค่าของ item เมื่อ expression นี้ได้รับการประมวลผลแล้ว

conditional sentence หมายถึง conditional statements ตั้งแต่ 1 statement ขึ้นไป และจบด้วย 1-period

statement-1 และ statement-2 อาจจะเป็น imperative statements หรือ conditional statement หรือทั้งสองชนิดรวมกันก็ได้ ตั้งแต่ 1 statement ขึ้นไป

ขั้นตอนการทำงาน conditional expression จะถูกตรวจสอบก่อนว่าจริงหรือไม่จริง ถ้าจริง statement-1 จะถูก execute จากนั้น จึงไปทำ sentence ถัดไป ถ้าไม่จริง มันจะข้ามไปทำ statement-2

รูปแบบที่ง่ายที่สุดของ **conditional sentence** นั้น อาจเขียนได้ดังนี้

IF conditional-expression any imperative-statement

ตัวอย่าง

IF STUDENT-SCORE > 89 MOVE "G" TO GRADE.

IF SINGLE ADD 1 TO S-COUNT.

6.2.1 นิพจน์มีเงื่อนไข (Conditional expressions)

นิพจน์มีเงื่อนไขเป็นตัวทำให้ object program มีทางเลือกระหว่างทิศทางของ control ขึ้นอยู่กับค่าจริงหรือไม่จริง จากการทดสอบ นิพจน์มีเงื่อนไขนี้จะอยู่ในคำสั่ง IF, PERFORM และคำสั่ง SEARCH

แบ่งออกเป็น 5 ชนิดคือ

relation condition

• sign condition

class condition

condition-name condition

switch-status condition

นอกจากนี้แล้ว ยังมีอีก 2 โครงสร้างที่มีผลต่อการประเมินผลของเงื่อนไข นั่นคือ

1. (condition)

เครื่องหมายวงเล็บใช้ในการรวมเงื่อนไขต่าง ๆ ให้ดูในหัวข้อเงื่อนไขรวม *

2. NOT condition

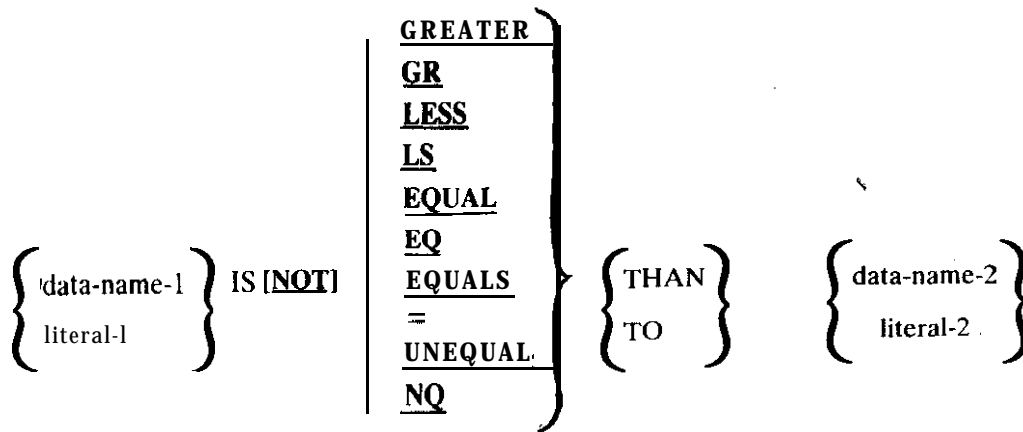
โครงสร้าง ...NOT condition... (เมื่อ condition เป็นเงื่อนไขชนิดใดชนิดหนึ่งใน 5 ชนิดที่กล่าวข้างต้น) ใช้ไม่ได้ ถ้าเงื่อนไขนั้น โดยตัวมันเองมี NOT

เงื่อนไขต่าง ๆ อาจนำมารวม (combined) เข้าด้วยกันโดยใช้ตัวปฏิบัติการตรรกะ (logical operators) เพื่อทำให้เป็นเงื่อนไขรวม

a) Relation condition หรือ Simple relational test

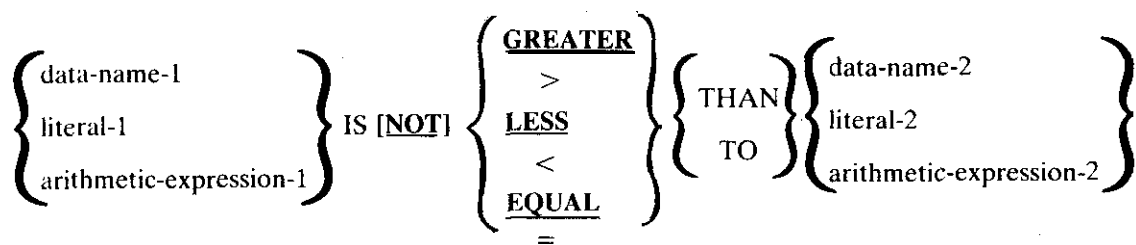
การทดสอบเงื่อนไขชนิดนี้ นิพจน์นั้นจะมีค่าจริงหรือไม่จริง ขึ้นอยู่กับการเปรียบเทียบความสัมพันธ์ทางด้านขนาดของตัวถูกกระทำ 2 ตัว โดยใช้ตัวปฏิบัติการเปรียบเทียบ (relational operators) มีรูปแบบดังนี้ (สำหรับเครื่อง CDC)

* เงื่อนไขรวม (compound conditions) รายละเอียดอยู่ในหัวข้อ 6.2.2



GR. LS. EQ, และ NQ เป็นคำย่อใช้แทน GREATER, LESS, EQUAL(S) และ UNEQUAL ใน expression สำหรับเครื่องหมาย = ใช้แทนคำ EQUAL(S)

สำหรับเครื่อง IBM ตัวปฏิบัติการเปรียบเทียบ ใช้ดังนี้



จากรูปแบบข้างต้นนี้ ตัวถูกกระทำตัวแรก (ตัวซ้ายมือ) เรียกว่า subject ของเงื่อนไข และตัวถูกกระทำตัวที่สอง (ตัวขวามือ) เรียกว่า object ของเงื่อนไขทั้ง subject และ object เป็น literal ทั้งคู่ไม่ได้ แต่อาจจะเป็น data-name ทั้งคู่ หรือ ตัวใดตัวหนึ่งเป็น literal อีกตัวหนึ่งมีมูลค่าเป็น data-name ได้

ถ้าข้อมูลในฟิลด์นั้นเป็นตัวเลขทั้งหมด (numeric fields) ทั้งสองชุด การเปรียบเทียบให้ดูจากค่าทางพีชคณิตได้โดยตรง

ตัวอย่าง

ค่าของ FIELDA เท่ากับ 44.40

ค่าของ FIELDB เท่ากับ 44.4

กรณีนี้ทั้งสองฟิลด์มีมูลค่าเท่ากัน

ถ้าข้อมูลในฟิลด์นั้น ๆ ไม่ใช่ตัวเลขทั้งหมด (non-numeric field) สำหรับเลข 0, 1, 2, 3,.....9 เลขที่มีค่าต่ำสุด คือ เลข 0 และเลขที่มีค่าสูงสุด คือ เลข 9 สำหรับตัวอักษร A,B,C,D,.....,Z ตัวอักษร A จะมีค่าต่ำสุด ในทำนองเดียวกันอักษร Z จะมีค่าสูงที่สุด สำหรับตัวอักษรพิเศษ เช่น . \$. + - ค่าใดสูงหรือค่าใดต่ำกว่ากัน ขึ้นอยู่กับชนิดของเครื่องคอมพิวเตอร์

สำหรับเครื่อง IBM

การเปรียบเทียบของตัวถูกกระทำที่เป็น non-numeric หมายถึง ตัวถูกกระทำทั้ง 2 ตัวเป็น non-numeric ทั้งคู่ หรือมีตัวถูกกระทำตัวหนึ่งเป็น numeric และตัวถูกกระทำอีกตัวหนึ่งเป็น non-numeric การเปรียบเทียบค่าของ 2 ตัวนี้ขึ้นอยู่กับ binary collating sequence ของตัวอักษรใน EBCDIC set

ใน EBCDIC collating sequence มีการเรียงลำดับค่าจากน้อยไปหามาก (ascending) ดังนี้

1		space
2		period or decimal point
3	<	less than symbol
4	(left parenthesis
5	+	plus sign
6	\$	currency symbol
7	*	asterisk
8)	right parenthesis
9		semicolon
10		hyphen or minus symbol
11	/	slash
12		comma
13	>	greater than symbol
14		apostrophe or single quotation mark

15	=	equal sign
16	"	quotation mark
17-42	A	thru Z
43-52	0	thru 9

ขนาดของตัวถูกกระทำ (size of an operand) คือจำนวนตัวอักขระทั้งหมดในตัวถูกกระทำนั้น

ข้อมูลกลุ่ม (group items) ทั้งหมด ถือว่าเป็นตัวถูกกระทำชนิด non-numeric การเปรียบเทียบ non-numeric แบ่งออกเป็น 2 ประเภทคือ ตัวถูกกระทำทั้ง 2 ตัว มีขนาดเท่ากัน และตัวถูกกระทำทั้ง 2 ตัว มีขนาดไม่เท่ากัน

1) การเปรียบเทียบของตัวถูกกระทำที่มีขนาดเท่ากัน ตัวอักขระในตำแหน่งที่ตรงกัน ของตัวถูกกระทำทั้ง 2 ตัวนี้จะนำมาเปรียบเทียบกัน จากปลายระดับสูง (high-order end) ไปจนปลายระดับต่ำ (low-order end) เมื่อ high-order end หมายถึง ตำแหน่งของตัวอักขระซ้ายมือสุด และ low-order end หมายถึง ตำแหน่งของตัวอักขระขวามือสุด

ถ้าตัวอักขระทุกคู่เปรียบเทียบแล้วเท่ากันจนถึงคู่สุดท้าย ตัวถูกกระทำทั้งสองตัวนี้จะมีค่าเท่ากัน

ถ้ามีตัวอักขระที่เปรียบเทียบกันคู่ใดคู่หนึ่งที่ไม่เท่ากัน ตัวอักขระ 2 ตัวนั้นจะถูกตรวจสอบว่ามีระดับของตำแหน่งใน collating sequence อย่างไร ตัวถูกกระทำที่ประกอบด้วยตัวอักขระที่มีระดับของ collating sequence สูงกว่าจะถือว่าเป็นตัวถูกกระทำที่มีค่ามากกว่า

2) การเปรียบเทียบของตัวถูกกระทำที่มีขนาดไม่เท่ากัน

ถ้าตัวถูกกระทำทั้ง 2 ตัวนี้ มีขนาดไม่เท่ากัน ตัวถูกกระทำที่มีขนาดสั้นกว่าเครื่องจะเติมเครื่องหมาย blank ให้จำนวนหนึ่งจนกระทั่งตัวถูกกระทำทั้ง 2 ตัวนี้มีความยาวเท่ากัน จากนั้นจึงทำการเปรียบเทียบค่าเช่นเดียวกับข้อ 1)

ตัวอย่าง

FIELDA มีค่าเท่ากับ ABCX4

FIELDB มีค่าเท่ากับ ABC84

ในที่นี้ FIELDA จะมีค่าน้อยกว่า FIELDB

b) Sign condition และ Sign test

ข้อความในเงื่อนไขนั้นจะเป็นจริง หรือไม่จริงขึ้นอยู่กับค่าของ item นั้นว่าเป็นค่าบวก, ค่าลบ หรือศูนย์ แต่ทั้งนี้ data item นั้นต้องเป็นตัวเลขเสมอ โดยมีรูปแบบดังนี้

data-name IS [NOT] $\left. \begin{array}{c} \underline{\text{POSITIVE}} \\ \underline{\text{NEGATIVE}} \\ \underline{\text{ZERO}} \end{array} \right\}$

ตัวถูกกระทำจะเป็น positive ถ้าค่าของมันมากกว่าศูนย์
 ตัวถูกกระทำจะเป็น negative ถ้าค่าของมันน้อยกว่าศูนย์และตัวถูกกระทำจะเป็น zero ถ้าค่าของมันเท่ากับศูนย์
 ฟังก์ชันที่ไม่มีเครื่องหมาย (unsigned field) เป็น positive หรือ zero

ตัวอย่าง 1

IF FIELDX IS POSITIVE, GO TO SECTIONA.
 IF FIELDX IS NEGATIVE, GO TO SECTIONB.
 GO TO EKROR-ROUTINE.

ตัวอย่าง 2

IF FIELDA IS NOT ZERO THEN NEXT SENTENCE ELSE
 GO TO PART B.

ตัวอย่าง 3

IF FINAL-ANSWER ZERO GO TO EOJ.

c) Class condition และ Class test

ข้อความในเงื่อนไขนั้น จะเป็นจริงหรือไม่จริง ขึ้นอยู่กับว่า content ใน item นั้นเป็น numeric หรือ alphabetic โดยมีรูปแบบ ดังนี้

data-name IS [NOT] $\left. \begin{array}{c} \underline{\text{NUMERIC}} \\ \underline{\text{ALPHABETIC}} \end{array} \right\}$

แต่ data-name item ดังกล่าว ต้องนิยามเป็น alphanumeric ใน DATA division (หมายถึงใช้ PICTURE X)

ตัวอย่าง

IF DATA-I IS NOT NUMERIC NEXT SENTENCE ELSE GO TO BBB
 IF INPUT-AREA IS ALPHABETIC GO TO SECTIONB.

IF PAY-FIELD NOT NUMERIC GO TO ERROR-ROUTINE.

IF STUDENT-NO NOT NUMERIC GO TO 100.

d) Condition-name condition และ Condition-name test

ข้อความในเงื่อนไขนั้น จะเป็นจริงหรือไม่จริง ขึ้นอยู่กับว่ามูลค่าของ item นั้น เหมือนกับเงื่อนไขที่กำหนด โดย condition-name entry ซึ่งใช้เลขบอกระดับ 88 ใน DATA division หรือไม่โดยมีรูปแบบ ดังนี้

[NOT] condition-name

ตัวอย่าง

03 M-STATUS PICTURE 9.

88 SINGLE VALUE 1

88 MARRIED VALUE 2.

88 WIDOWED VALUE 3.

88 DIVORCED VALUE 4.

PROCEDURE DIVISION.

IF SINGLE MOVE DATA-1 TO FIELD-1 ELSE GO TO TEST-234.

ในที่นี้ M-STATUS เป็นตัวแปรมีเงื่อนไข (conditional variable)

SINGLE, MARRIED, WIDOWED และ DIVORCED เป็นชื่อมีเงื่อนไข (condition-names)

เฉพาะเงื่อนไขอย่างใดอย่างหนึ่งซึ่งกำหนดโดยชื่อมีเงื่อนไขจะต้องอยู่ในแต่ละเรคคอร์ดในแฟ้มข้อมูล ในการหาสถานภาพสมรสของแต่ละเรคคอร์ดประมวลผลดังนี้

IF SINGLE...

เมื่อประมวลผลแล้วจะได้ค่าจริงหรือค่าเท็จที่ object program จะนำไปใช้ต่อ

ชื่อมีเงื่อนไขที่ใช้ในเงื่อนไขต่าง ๆ เป็นการย่อของเงื่อนไขเปรียบเทียบ (relation condition) เนื่องจากชื่อมีเงื่อนไข จะมีค่าเท่ากับค่าใดค่าหนึ่ง หรือช่วงของค่าใด ๆ (ranges of values) ที่กำหนดให้กับตัวแปรมีเงื่อนไขนั้นคือ ในการหาว่าเงื่อนไข SINGLE มีหรือไม่เขียนดังนี้

IF M-STATUS = 1...

จะมีความหมายอย่างเดียวกับการใช้ condition-name test ข้างล่างนี้

IF SINGLE...

ถ้าชื่อมีเงื่อนไข ถูกกำหนดให้กับช่วงหนึ่งของค่าต่างๆ (a range of values) หรือหลาย ๆ ช่วงของค่าต่างๆ (several ranges of values) ตัวแปรที่มีเงื่อนไขจะถูกทดสอบว่ามีค่าของมันตกอยู่ในช่วงต่างๆ นั้นหรือไม่ รวมทั้งค่าสุดท้าย ผลลัพธ์ของการทดสอบจะเป็นจริงถ้ามีค่าใดค่าหนึ่งที่สมนัยกับชื่อมีเงื่อนไขเท่ากับค่าของตัวแปรมีเงื่อนไขนั้น

ตัวอย่างของทั้ง elementary condition-name entries และ group condition-name entries ให้ดูรายละเอียดในหัวข้อ VALUE clause ใน Data division

e) Switch-status condition และ Switch-status test

เงื่อนไข switch-status คือการหาสถานภาพ (status) ของ device switch ว่าเปิด (on) หรือปิด (off)

โปรแกรมเมอร์เป็นคนกำหนด switch-status name เพื่อนำมาใช้ทดสอบสถานภาพของ select jump switch (1 ถึง 6) บนคอนโซลของเครื่องคอมพิวเตอร์

switch-status-name ต้องนิยามในพารากราฟ special-names ใน environment division มาก่อนแล้ว โดยมีรูปแบบดังนี้

[NOT] switch-status-name

ตัวอย่าง

IF SW3ON PERFORM RE-WRITE THRU RE-STORE.

IF SW1ON GO TO PARA-CHECK.

6.2.2 เงื่อนไขรวม (Compound conditions)

เงื่อนไขอย่างง่าย (simple condition) ตั้งแต่ 2 ชุดขึ้นไปถ้าเอามารวมเข้าด้วยกันจะเรียกว่า เงื่อนไขรวม

เงื่อนไขอย่างง่ายแต่ละชุดจะถูกกั้นจากกันด้วยตัวปฏิบัติการตรรกะ (logical operators) ตัวใดตัวหนึ่งใน AND หรือ OR

ความหมายของตัวปฏิบัติการตรรกะมีดังนี้

Logical operator**Meaning**

OR

logical inclusive OR นั่นคือตัวใดตัวหนึ่งเป็นจริง หรือ ทั้ง 2 ตัวเป็นจริง

AND

logical conjunction นั่นคือ ทั้ง 2 ตัวต้องเป็นจริง

NOT

logical negation

รูปข้างล่างนี้แสดงความสัมพันธ์ระหว่างตัวปฏิบัติการตรรกะกับเงื่อนไขอย่างง่าย A และ B

A	B	A AND B	A OR B	NOT A	NOT (A AND B)	NOT A AND B	NOT (A OR B)	NOT A OR B
True	True	True	True	False	False	False	False	True
False	True	False	True	True	True	True	False	True
True	False	False	True	False	True	False	False	False
False	False	False	False	True	True	False	True	True

Logical Operators and the Resulting Values upon Evaluation**กฎการประเมินผล (Evaluation rules)**

การประเมินผลตรรกะเริ่มต้นจากภายในวงเล็บคู่ในสุดก่อนแล้วตามด้วยวงเล็บชุดนอกถัดไป

ถ้าลำดับที่ (order) ของการประเมินผล ไม่ได้ถูกกำหนดโดยเครื่องหมายวงเล็บ การประเมินผลนิพจน์จะเป็นไปตามลำดับดังนี้

1. นิพจน์เลขคณิต (arithmetic expressions)
2. ตัวปฏิบัติการเปรียบเทียบ (relational-operators)
3. เงื่อนไข ([NOT] condition)
4. AND และเงื่อนไขโดยรอบของมันจะถูกประเมินผลก่อน, เริ่มจากทางซ้ายของนิพจน์ไปทางขวา
5. OR และเงื่อนไขโดยรอบของมันจะถูกประเมินผล เช่นเดียวกันก็จากซ้ายไปขวา

ตัวอย่าง เงื่อนไขรวม

A IS NOT GREATER THAN B OR A + B IS EQUAL TO C AND D IS POSITIVE

เมื่อประเมินผลจะมีความหมายอย่างเดียวกับนิพจน์รวมชุดข้างล่างนี้

(A IS NOT GREATER THAN B(OR (((A + B) IS EQUAL TO C) AND (D IS POSITIVE))).

ลำดับที่ของการประเมินผลเป็นดังนี้

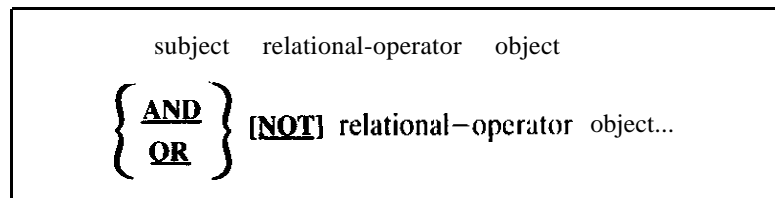
1. (A + B) ได้รับการประเมินผลเป็นอันดับแรก ให้ผลลัพธ์เป็น x
2. (A IS NOT GREATER THAN B) ได้รับการประเมินผล ให้ผลลัพธ์เป็นค่าจริงหรือค่าเท็จ สมมติเป็น t_1
3. (X IS EQUAL TO C) ได้รับการประเมินผล ให้ผลลัพธ์เป็นค่าจริงหรือค่าเท็จ สมมติค่าเป็น t_2
4. (D IS POSITIVE) ได้รับการประเมินผล ให้ผลลัพธ์เป็นค่าจริงหรือค่าเท็จ สมมติเป็น t_3
5. (t_2 AND t_3) ได้รับการประเมินผล ให้ผลลัพธ์เป็นค่าจริงหรือค่าเท็จ สมมติเป็น t_4
6. (t_1 OR t_4) ได้รับการประเมินผล ให้ผลลัพธ์สุดท้ายเป็นค่าจริงหรือค่าเท็จ ซึ่งจะ
เป็นค่าของนิพจน์ชุดนี้

Implied subjects and relational-operators

เมื่อเงื่อนไขเปรียบเทียบหลายชุด (relational conditions) เขียนเรียงตามลำดับกัน เงื่อนไขใด ๆ ก็ตามยกเว้นเงื่อนไขแรกอาจจะเขียนย่อได้ โดย

1. ไม่ต้องเขียน subject ของเงื่อนไขเปรียบเทียบ (relational condition), หรือ
2. ไม่ต้องเขียน subject และ ตัวปฏิบัติการเปรียบเทียบ ของเงื่อนไขเปรียบเทียบภายในเงื่อนไขเปรียบเทียบหลายชุดที่เขียนเรียงลำดับกัน อาจจะใช้รูปแบบการย่อทั้ง 2 ข้อข้างต้นได้ ผลของการใช้การย่อจะถือราวกับว่า subject ที่ไม่เขียนนั้นเอามาจาก subject ตัวล่าสุดซ้ายมือและในทำนองเดียวกัน ตัวปฏิบัติการเปรียบเทียบที่ไม่เขียนนั้น เอามาจากตัวปฏิบัติการเปรียบเทียบตัวซ้ายมือเช่นกัน

รูปแบบของ Implied subject



รูปแบบของ Implied subject and Relational-operator

$$\dots \text{subject} \text{ relational-operator} \text{ object} \left\{ \begin{array}{c} \text{AND} \\ \text{OR} \end{array} \right\} \text{[NOT]} \text{ object} \dots$$

ความคลุมเครือของผลลัพธ์อาจเกิดขึ้นได้จากการใช้ NOT ในการย่อในการปฏิบัติการ AND เมื่อเฉพาะ subject เท่านั้นที่ถูก implied, NOT จะมีความหมายเป็นตัวปฏิบัติการตรรกะ 1 ตัว ไม่ใช่ส่วนหนึ่งของตัวปฏิบัติการเปรียบเทียบ (relational operator)

ตัวอย่าง 1

$$A \text{ NOT } > B \text{ AND } < C$$

หมายถึง $A \text{ NOT } > B \text{ AND } A < C$

แต่ถ้าทั้ง subject และ relational operator ถูก implied ทั้งคู่ NOT จะมีความหมายเป็นส่วนหนึ่งของตัวปฏิบัติการเปรียบเทียบ

ตัวอย่าง 2

$$A \text{ NOT } > B \text{ AND } C$$

หมายถึง $A \text{ NOT } > B \text{ AND } A \text{ NOT } > C$

ตัวอย่าง 3

Implied subject

$$A = B \text{ OR } \text{NOT } > C \text{ (subject A is implied)}$$

หมายถึง $A = B \text{ OR } \text{NOT } A > C$

ตัวอย่าง 4

Implied subject and relational operator

$$A = B \text{ AND } C \text{ (subject and relational-operator A = are implied)}$$

หมายถึง $A = B \text{ AND } A = C$

$$A \text{ NOT } = B \text{ AND } C \text{ (subject and relational-operator A NOT = are implied)}$$

หมายถึง $A \text{ NOT} = B \text{ AND } A \text{ NOT} = C$

ตัวอย่าง 5 Implied subject. and subject and relational-operator

$A > B \text{ AND NOT } < C \text{ AND } D$ (subject A ถูก implied ในเงื่อนไขที่สอง, subject A และ relational-operator $<$ ถูก implied ในเงื่อนไขที่สาม)

หมายถึง $A > B \text{ AND NOT } A < C \text{ AND } A < D$

ตัวอย่างข้างต้นนี้ subject ที่ไม่ได้เขียนเอามาจาก subject ตัวล่าสุดซ้ายมือ คือ A และ relational-operator ที่ไม่ได้เขียน เอามาจาก relational-operator ตัวล่าสุดซ้ายมือ คือ $<$

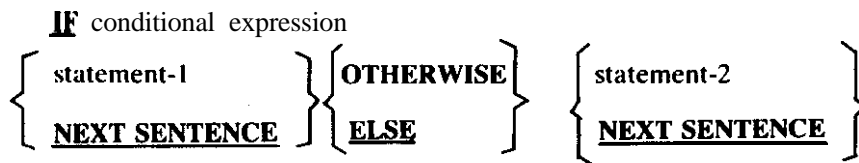
แบบฝึกหัด

ให้ A, B, C และ D เป็นชื่อมีเงื่อนไข (condition names) สมมติว่า A และ B มีค่าเป็นจริง (true) ในขณะที่ C และ D มีค่าเป็นเท็จ (false) จงบอกมูลค่าของ compound conditions ข้างล่างนี้ว่าเป็นจริงหรือเป็นเท็จ

- a) $A \text{ AND NOT } C \text{ OR } D$
- b) $A \text{ AND NOT } C \text{ OR NOT } D$
- c) $A \text{ AND NOT } (C \text{ OR NOT } D)$
- d) $A \text{ AND } C \text{ OR } B \text{ AND } D$
- e) $A \text{ OR } C \text{ AND } B \text{ OR } D$

6.2.3 Nested conditional statements

มีรูปแบบดังนี้



เมื่อ statement-1 และ statement-2 อาจเป็น

- a) an imperative statement
- b) a series of imperative statements
- c) หรือ imperative statement ตามด้วย a conditional statement

61 statement-1 หรือ statement-2 อย่างใดอย่างหนึ่ง หรือทั้งสอง statements เป็น conditional statement, statement ที่อยู่ในรูปแบบดังกล่าวข้างต้นจะมีชื่อเรียกว่า nested conditional statement ซึ่งอาจจะแทนด้วยโครงสร้างต่อไปนี้

IF (C₁) (imp₁) (**IF** (C₂) (imp₂) (. (. (. **IF** (C_n) (imp_n) ELSE n) .) .) .) ELSE 2) ELSE 1

(C_i) แต่ละตัว แทน a conditional expression

(imp_i) แต่ละตัว แทน an imperative statement
หรือ a series of imperative statements

(i) แต่ละตัว แทน an imperative statement
หรือ a conditional statement (และตัวของมันเองอาจจะประกอบด้วย conditional statement อื่นก็ได้ และ ต่อ ๆ ไป หรือ NEXT SENTENCE

(C_n) แต่ละตัว แทน conditional expression ใน conditional statement และตัวมันเองจะประกอบด้วย conditional statement อีกต่อไปไม่ได้

ตามรูปแบบนี้ จำนวน ELSE หรือ OTHERWISE ใน nested conditional statement จะต้องเท่ากับ จำนวน IF ใน statement เดียวกันนั้นเสมอไป ยกเว้นกรณีที่ข้อความ ELSE (หรือ OTHERWISE) NEXT SENTENCE ตามด้วย period ของ sentence นั้น ข้อความดังกล่าวจะไม่เขียนก็ได้ ซึ่งจะทำให้จำนวน IF และจำนวน ELSE ใน sentence นี้ไม่เท่ากัน

เวลา execute เครื่องคอมพิวเตอร์จะทำตามลำดับขั้นตอนดังนี้

	ถ้า	(C_1)	เป็นจริง	ทำ (\underline{imp}_1)
	แล้ว ถ้า	(C_2)	เป็นจริง	ทำ (\underline{imp}_2)
	แล้ว ถ้า	(C_3)	เป็นจริง	ทำ (\underline{imp}_3)

ถ้า (C_n) เป็นจริง ทำ (\underline{imp}_n)

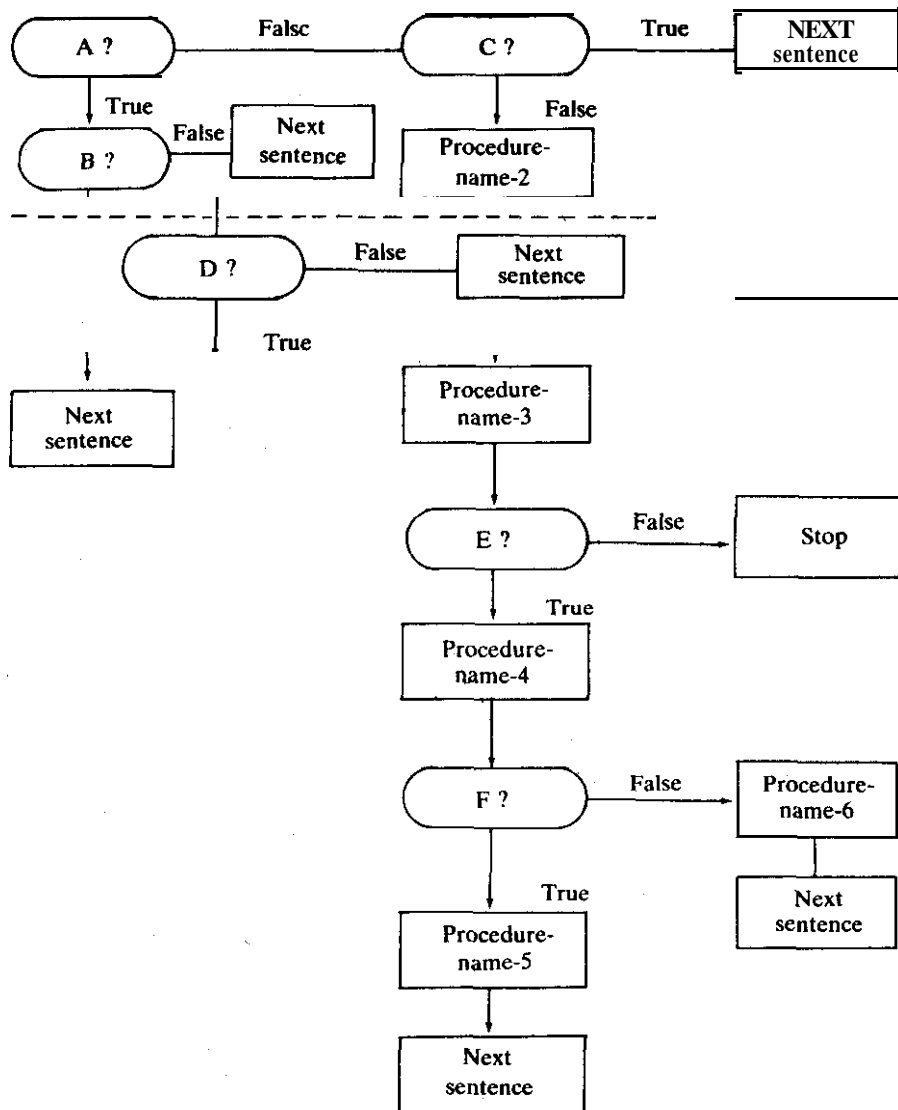
แล้วทำ sentence ถัดไป

(\underline{imp}_i) ทุกตัว ยกเว้น (\underline{imp}_n) จะเป็นคำสั่ง GO TO หรือ NEXT SENTENCE ไม่ได้ เพราะถ้าเช่นนั้น จะทำให้ sequence ถัดไปไม่ถูก execute

(\underline{imp}_n) จะเป็น imperative statement ชนิดไหนก็ได้ หรืออาจจะเป็น NEXT SENTENCE ก็ได้

ถ้า (C_i) ตัวใดตัวหนึ่งเป็นเท็จ (\underline{imp}_i) ตัวนั้นก็ไม่ถูก execute จากนั้น i ที่ตามหลัง ELSE จะถูก execute เป็นลำดับถัดไป ส่วนที่เหลือของ sequence จะถูก execute ไปตามปกติ

ตัวอย่าง ให้นักศึกษาเขียน one conditional sentence จาก flow-chart ข้างล่างนี้



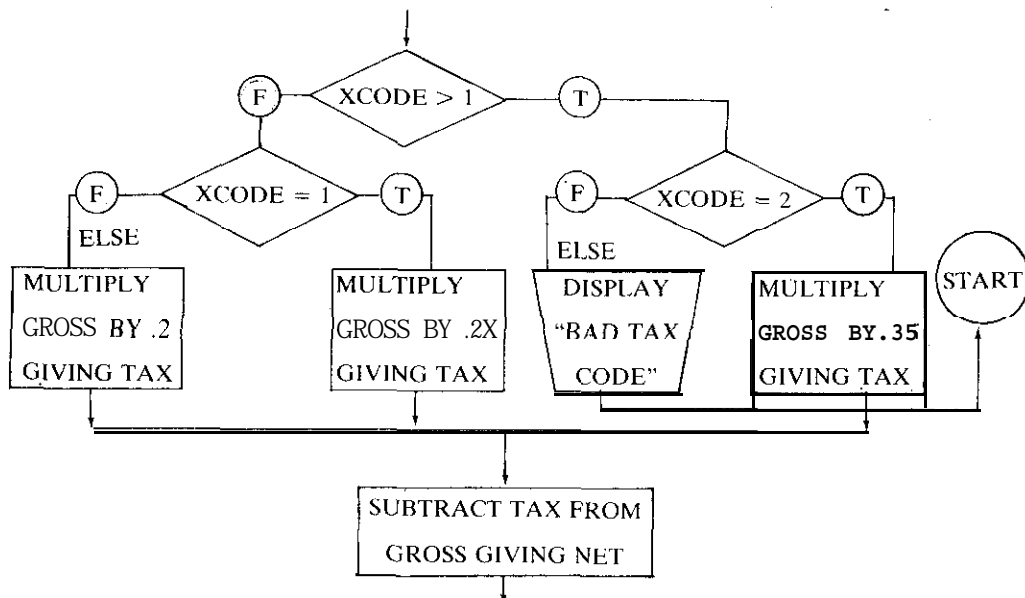
flow-chart ดังกล่าวประกอบด้วย independent nested of conditional statements 2 ชุด
 nest ชุดแรก จบหลังคำสั่ง PERFORM procedure-name-2, nest ชุดที่สองประกอบด้วยส่วนที่เหลือ
 ทั้งหมด รวมทั้งมี ELSE NEXT SENTENCE แล้วมี period เมื่อ A, B, C, D, E และ F เป็น
 conditional expressions

เฉลย

IF A THEN IF B PERFORM procedure-name-1 ELSE
 NEXT SENTENCE ELSE IF C NEXT SENTENCE ELSE PERFORM
 procedure-name-2 IF D PERFORM procedure-name-3 IF E
 PERFORM procedure-name-4 IF F PERFORM procedure-name-5
 ELSE PERFORM procedure-name-h ELSE STOP RUN.

ตัวอย่าง

จาก flowchart ข้างล่างนี้



สามารถเขียนเป็น nested conditional statement ได้ดังนี้

บรรทัดที่ 1 . IF XCODE GR 1
 2 IF XCODE EQ 2
 3 MULTIPLY GROSS BY .35 GIVING TAX
 4 ELSE DISPLAY "BAD TAX CODE"
 5 GO TO START
 6 ELSE IF XCODE EQ 1

- 7 MULTIPLY GROSS BY .28 GIVING TAX
- 8 ELSE MULTIPLY GROSS BY .2 GIVING TAX.
- 9 SUBTRACT TAX FROM GROSS GIVING NET.

	บรรทัดที่
a) ถ้า XCODE มีค่าเท่ากับ 0 บรรทัดที่จะถูก execute คือ
b) ถ้า XCODE มีค่าเท่ากับ 1 บรรทัดที่จะถูก execute คือ
c) ถ้า XCODE มีค่าเท่ากับ 2 บรรทัดที่จะถูก execute คือ
d) ถ้า XCODE มีค่าเป็นจำนวนบวกใด ๆ นอกจากนี้ บรรทัดที่จะถูก execute คือ

เฉลย

- a) 1, 6, 8, 9
- b) 1, 6, 7, 9
- c) 1.2.3.9
- d) 1. 2. 4. 5

แบบฝึกหัด

1. คำสั่ง IF ต่อไปนี้ถูกต้องหรือไม่เพราะเหตุใด

```
IF A = B WRITE EQUAL- RECORD.
```

```
ELSE WRITE UNEQUAL-RECORD.
```

2. จงอธิบายขั้นตอนการทำงานของคำสั่ง IF ข้างล่างนี้ แล้วบอกด้วยว่าทั้งสองเซทนี้ทำงานแตกต่างกันอย่างไร ?

```
IF SALES IS LESS THAN QUOTA, MOVE "BELOW" TO MEMO.
```

```
ADD SALES TO Y-TO-D-SALES.
```

```
IF SALES IS LESS THAN QUOTA; MOVE "BELOW" TO MEMO.
```

```
ADD SALES TO Y-TO-D-SALES.
```

3. จงเขียน flowchart จากคำสั่ง nested IF ต่อไปนี้แล้วเขียนคำสั่งใหม่ โดยใช้ Simple IF หลาย ๆ คำสั่ง สำหรับ compound conditions ให้ใช้ AND, OR หรือ NOT operator.

```
IF C1 IF C2 ADD 1 TO S1 ELSE IF C3
```

```
MOVE 0 TO S2 ELSE NEXT SENTENCE ELSE
```

```
ADD 1 TO S3.
```

4. จงเขียน flowchart และอธิบายความหมายของคำสั่งข้างล่างนี้

```
IF AMOUNT IS LESS THAN 100
```

```
IF AMOUNT IS GREATER THAN 50
```

```
MOVE 0.3 TO RATE
```

```
ELSE
```

```
MOVE 0.4 TO RATE
```

```
ELSE
```

```
MOVE 0.2 TO RATE.
```

5. จากตารางเงื่อนไขข้างล่างนี้

QUANTITY	PRICE	RATING	DISCOUNT
> 100	> 10	< 2	0.05
> 100	> 10	≥ 2	0.10
> 100	≤ 10	< 2	0.15
> 100	≤ 10	≥ 2	0.20
≤ 100	{ < } { = 10 }	{ < } { = 2 }	0.25

- a) จงเขียน flowchart ที่มีความหมายอย่างเดียวกับข้อมูลในตาราง
 b) จงเขียนคำสั่งโดยใช้ nested IF เพื่อย้ายค่าไปไว้ที่ DISCOUNT ทั้งนี้ให้ขึ้นกับเงื่อนไขต่าง ๆ ในตาราง

6. กำหนด input field ไว้ดังนี้

03 IN-FIELD PICTURE X(10).

และนิยามอีก 2 ฟیلด์ใน working-storage section

01 AMOUNT PICTURE 9(10).

01 NAME PICTURE A(10).

ต้องการทดสอบค่าของ IN-FIELD, ถ้าค่านี้เป็นตัวเลขให้เก็บใน AMOUNT, ถ้าเป็นชื่อให้เก็บใน NAME และถ้าค่านี้เป็นตัวอักษรหลายอย่างปนกัน ให้ไปที่พารากราฟชื่อ ERROR

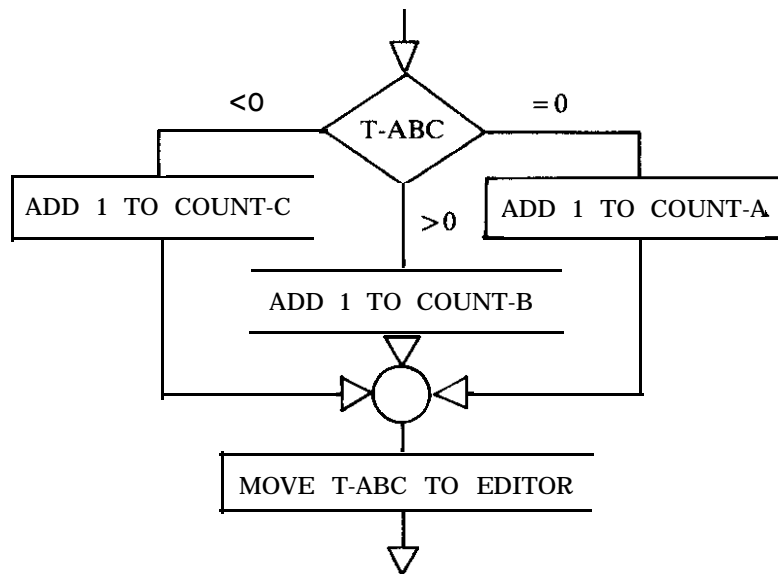
7. จงเขียน flowchart และส่วนหนึ่งของโปรแกรมใน procedure division เพื่อให้ทำเงื่อนไขต่าง ๆ ดังนี้

ถ้า ACCUM มีค่าเท่ากับ 0 และ TOTAL มีค่าเป็นลบ ให้ย้าย N ไปยัง PRINT-SIGNAL

ถ้าผลบวกของ TOTAL และ ACCUM มีค่ามากกว่า 0 ให้ไปทำพารากราฟ ชื่อ C25

ถ้าทั้ง TOTAL และ ACCUM มีค่ามากกว่า 0 ทั้งคู่ ให้ย้าย Y ไปยัง PRINT-SIGNAL แล้วไปทำพารากราฟชื่อ C80

8. จงเขียนคำสั่งจาก flowchart ข้างล่างนี้



9. จากกลุ่มของคำสั่งข้างล่างนี้ ให้เขียนใหม่เป็น 1 conditional sentence โดยให้มีความหมายเหมือนเดิม

```

IF OVER90-DUE IS NOT NEGATIVE GO TO P900-RETURN.
MOVE OVER90-DUE TO AMOUNT.
MOVE ZEROS TO OVER90-DUE
ADD AMOUNT TO OVER60-DUE.
IF OVER60-DUE IS NOT NEGATIVE GO TO P900-RETURN.
MOVE OVER60-DUE TO AMOUNT.
MOVE ZEROS TO OVER60-DUE.
ADD AMOUNT TO OVER30-DUE.
IF OVER30-DUE IS NOT NEGATIVE GO TO P900-RETURN.
MOVE OVER30-DUE TO AMOUNT,
MOVE ZEROS TO OVER30-DUE
ADD AMOUNT TO CURR-DUE
P900-RETURN0
  
```

10. จากคำสั่งข้างล่างนี้

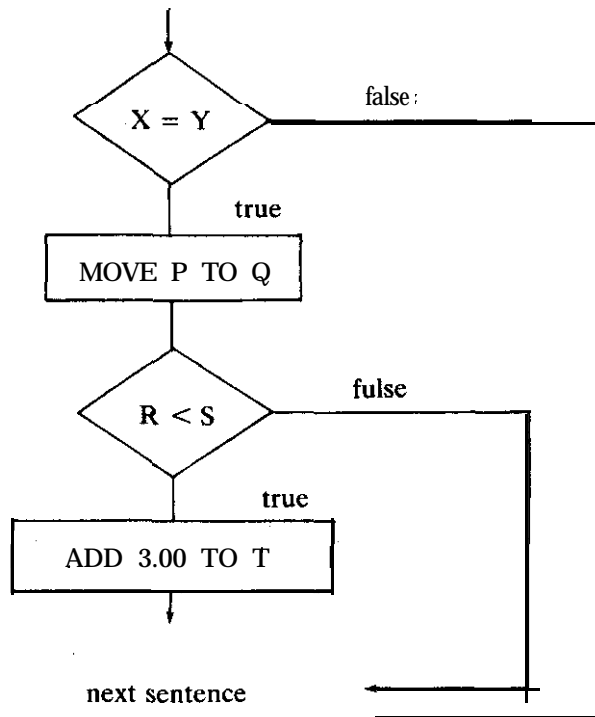
```
IF GRSPAY > 1000.00
  IF GRSPAY > 500.00
    MOVE 0.05 TO RETRMNT-DEDUC
  ELSE
    MOVE 0.03 TO RETRMNT-DEDUC
ELSE
  MOVE 0.07 TO RETRMNT-DEDUC.
```

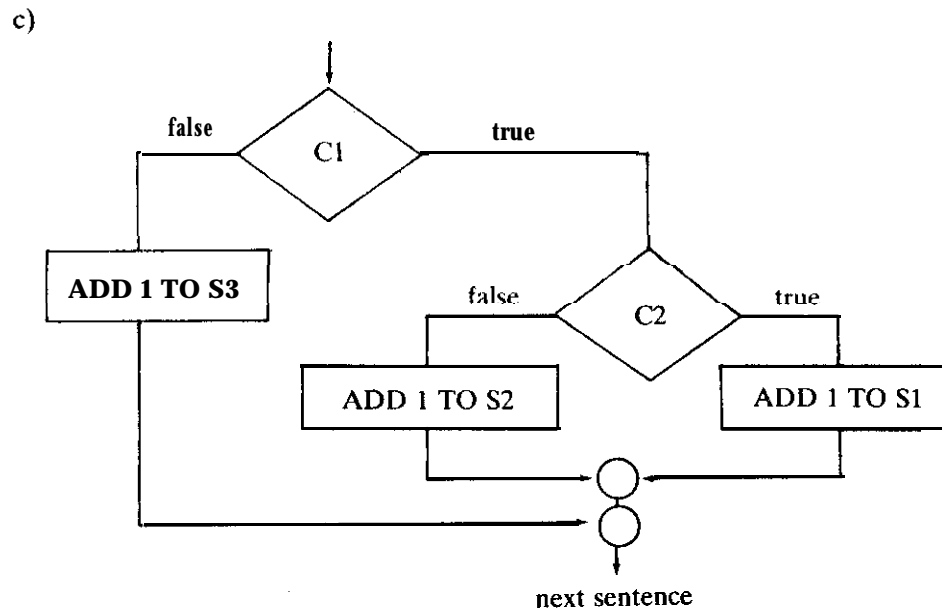
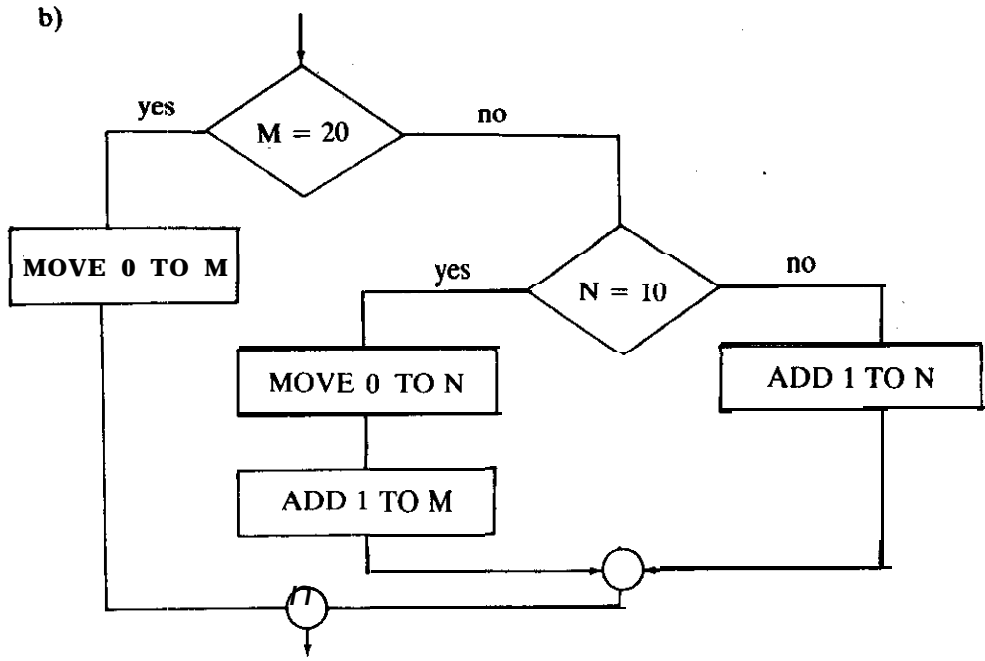
จงบอกมูลค่าของ RETRMNT-DEDUC ในกรณีต่อไปนี้, ถ้า GRSPAY มีมูลค่า

- C.1) มากกว่า 1,000
- C.2) มากกว่า 600 แต่น้อยกว่า 1,000
- C.3) น้อยกว่าหรือเท่ากับ 500

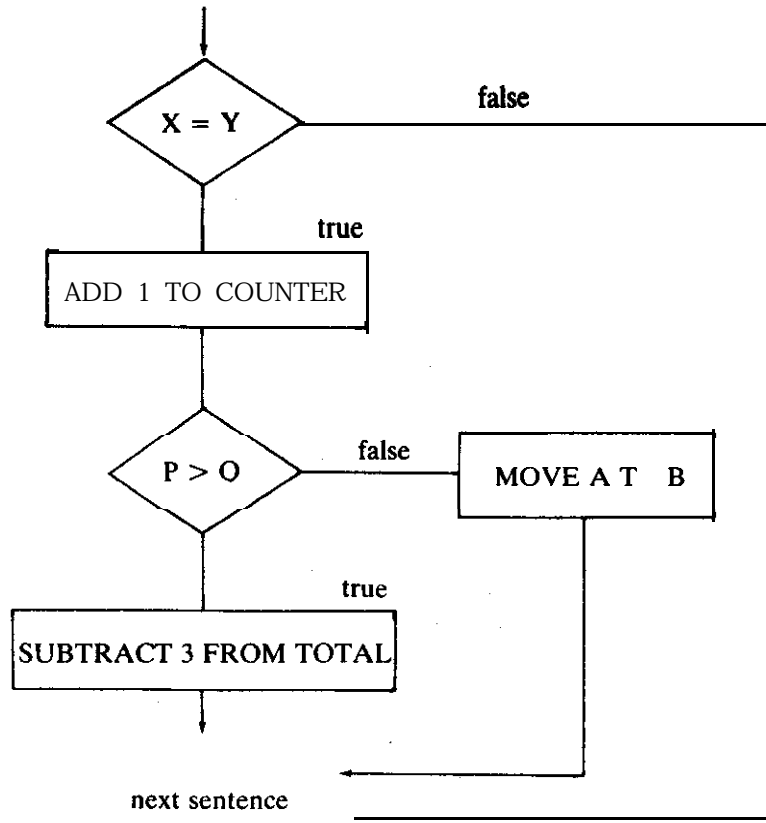
11. จงเขียนคำสั่ง 1 conditional sentence จาก flowchart ข้างล่างนี้

a)

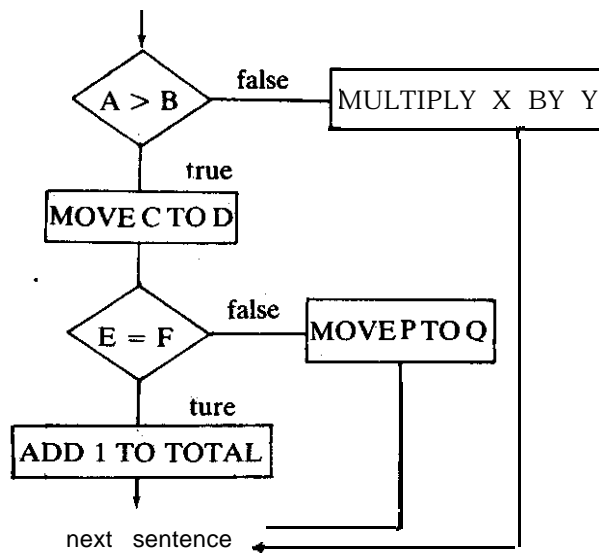




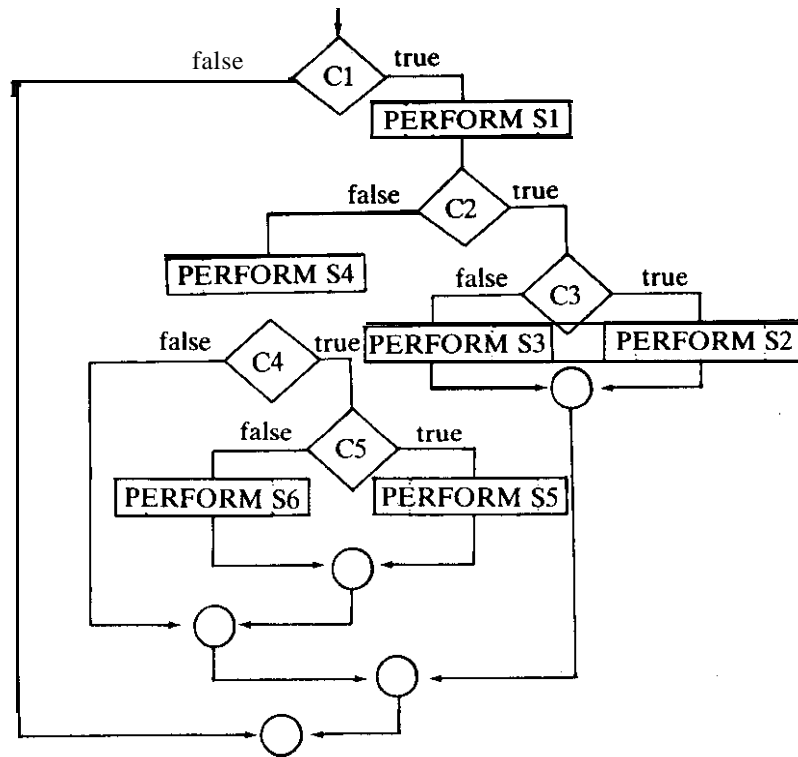
d)



e)



12. จงเขียน 1 conditional sentence จาก flowchart ข้างล่างนี้



13. จงเขียน flowchart จากคำสั่งข้างล่างนี้

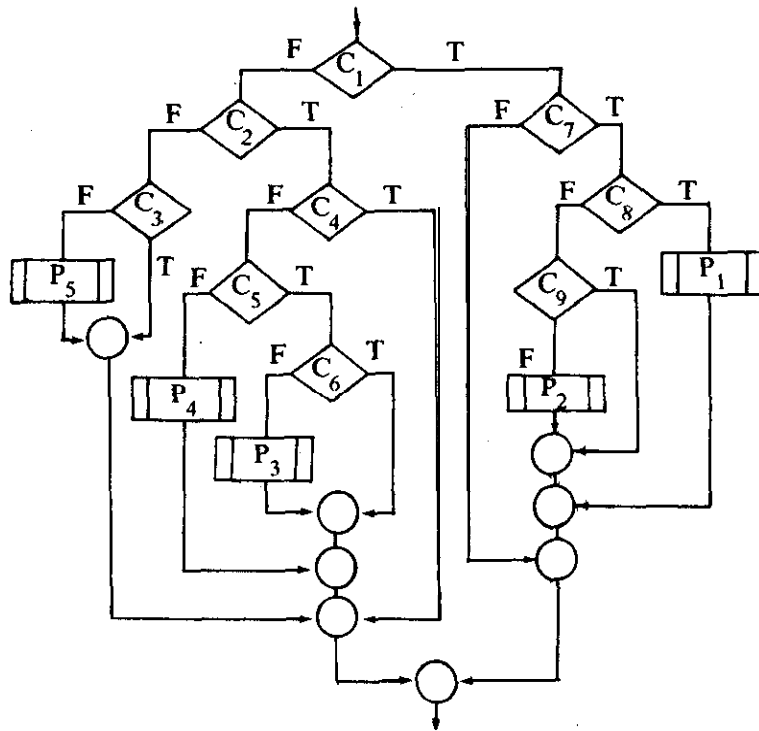
```

IF C1
  IF C2
    PERFORM S1
  ELSE
    IF C3
      PERFORM S2
    ELSE
      NEXT SENTENCE
  ELSE
    PERFORM S3.

```

แล้วเปลี่ยนคำสั่ง nested IF ข้างต้นให้เป็น simple IF หลาย ๆ คำสั่ง ถ้าจำเป็นอาจจะ
ต้องใช้ AND หรือ NOT operator ช่วยได้

14. a) จงเขียน 1 conditional sentence or flowchart ข้างล่างนี้ เมื่อ C_1, C_2, \dots, C_9 เป็นชื่อมีเงื่อนไข (condition names) และ P_1, P_2, \dots, P_5 เป็นชื่อพารากราฟ (paragraph names)



b) จงเขียน flowchart จากคำสั่งข้างล่างนี้

```

IF C1
    AND (C2 OR C3)
    PERFORM F1
    PERFORM F2
ELSE
    IF C3
        OR (C6 AND C7)
        PERFORM F3
    ELSE
        NEXT SENTENCE.
    
```

15. data description entries **ข้างล่างนี้เป็นข้อมูลของผู้สมัครเข้าทำงานแต่ละคน**

02 EDUCATION PICTURE 99.

88 U-S-GRAD VALUE IS 12.

88 COLLEGE-GRAD VALUE IS 16.

88 MASTERS-GRAD VALUE IS 17.

88 DOCTORATE-GRAD VALUE IS 20.

02 YEARS-OF-EXPERIENCE PICTURE 99.

02 SEX PICTURE X.

88 MALE VALUE 'M'.

88 FEMALE VALUE 'F'.

02 GEOGRAPHIC-PREFERENCE PICTURE 9.

88 EAST VALUE 1.

88 WEST VALUE 2.

88 NORTH VALUE 3.

88 SOUTH VALUE 4.

88 WILLIN-TO-TRAVEL VALUE 5.

บริษัทต้องการพนักงานเฉพาะที่มีคุณสมบัติข้อใดข้อหนึ่งในสามข้อต่อไปนี้

- 1) ผู้ชาย, มีประสบการณ์ทำงาน 5 ปี, จบการศึกษาระดับมัธยม, ชอบเดินทาง
- 2) ผู้ชาย, มีประสบการณ์ทำงาน 1 ปี, จบการศึกษาระดับปริญญาโท, ชอบภูมิภาคทางตะวันตกหรือทางใต้
- 3) ผู้หญิง, มีประสบการณ์ทำงาน 3 ปี, จบการศึกษาระดับปริญญาเอก, ชอบภูมิภาคทางตะวันออก

จงเขียน 1 compound conditional sentence ตรวจสอบว่ามีผู้สมัครคนใดหรือไม่ที่มีคุณสมบัติครบตามข้อใดข้อหนึ่งข้างต้น ถ้าพบให้ใช้คำสั่ง WRITE PRINTLINE FROM NAME ถ้าไม่มีผู้สมัครคนใดเลยที่มีคุณสมบัติดังกล่าวให้ไปที่พารากราฟ ชื่อ NOT-REQUIRED

16. Do the **following matching** exercise :

- | | |
|---------------------|---------------------|
| 1) data-name | -a) STOP RUN |
| 2) STOP statement | ___b) DISPLAY WIDTH |
| 3) Input statement | ___c) LENGTH |
| 4) reserved word | ___d) ACCEPT WIDTH |
| 5) Output statement | ___e) BY |

17. **Fill the blank with the letter of the appropriate entry on the right.**

A **choice from the right hand column may be used once, more than once, or not at all.**

- | | |
|--|-------------------------------|
| a) Conditional branching statement | 1) ACCEPT statement |
| b) The COBOL relational operator which test equality | 2) GO TO statement |
| c) Identifiers a position in a COBOL program | 3) HAS THE SAME VALUE AS |
| d) An entry written four spaces to the left of other entries | 4) ELSE |
| e) Unconditional branching statement | 5) period |
| f) Locates the end of the false branch of an IF-----ELSE statement | 6) diamond |
| g) May be used to create a loop | 7) IF statement |
| h) COBOL word which may be used instead of OTHERWISE | 8) STOP RUN. |
| i) Locate the end of the true branch of a simple IF statement | 9) A IS GREATER THAN B |
| j) Flow charting symbol for a conditional branch | 10) MOVE statement |
| k) May be a number | 11) paragraph-name |
| l) Must immediately follow a paragraph- name | 12) DISPLAY PAY |
| m) Input statement | 13) IS EQUAL TO |
| n) Last executable statement | 14) rectangle |
| o) Condition | 15) hyphen |
| p) Locates the end of the true branch of an IF_____ELSE statement | 16) data-name |

18. Study the following **PROCEDURE DIVISION** of a COBOL program and **answer the questions. Numbers before the statements are not part of the program. They are for reference only.**

- 1) MOVE 1 TO SQUARE-YARDS.
- 2) TESTING.

- 3) IF SQUARE-YARDS IS GREATER THAN 25
- 4) GO TO FINISH-UP.
- 5) MULTIPLY SQUARE-YARDS BY 9 GIVING SQUARE-FEET.
- 6) DISPLAY SQUARE-YARDS.
- 7) DISPLAY SQUARE-FEET.
- 8) ADD 1, SQUARE-YARDS GIVING SQUARE-YARDS.
- 9) GO TO TESTING.
- 10) FINISHUP.
- 11) STOP RUN.

- a) Does this program contain an input statement?
- b) What line first puts a number into the computer so that it has something to work with
- c) What line changes the value **of** SQUARE-YARDS?
- d) How rapidly does SQUARE-YARDS increase?
- e) What line tests the value of SQUARE-YARDS against aprearranged top limit?
- f) How many values for SQUARE-YARDS will the computer print out?
- g)** Which lines cause the computer to print out values?
- h) How many lines will the computer print out?
- i) Could the programmer use STOP RUN in the line 4) position?
- j) What line(s) would have to be changed if the computer is to continue working until the value of SQUARE-YARDS is greater than **50**?
- k) What is **the** effect on the program if the period at the end of the line 4) is omitted and **an ELSE** is inserted between line 4) and line **5**)?
- l) Is this program an example of a loop?
- m) How many times will the computer execute line **1**)?
- n)** How many times will the computer execute line **6**)?
- o)** How many times will the computer execute line **11**)?

19. Write the required entries for a program whose PROCEDURE DIVISION contain the following statements :

```
START.
  ACCEPT ITEM-NAME.
  ACCEPT QUANTITY-ON-HAND.
  IF QUANTITY-ON-HAND IS EQUAL TO 0
  STOP RUN.
```

เฉลย

ข้อ 16.

- a) 2
- b) 5
- c) 1
- d) 3
- e) 4

ข้อ 17.

- a) 7
- b) 13
- c) 11
- d) 11
- e) 2
- f) 5
- g) 2
- h) 4
- i) 5
- j) 6
- k) 11
- l) 5
- m) 1
- n) 8
- o) 9
- p) 4

ข้อ 18.

- a) no
- b) 1
- c) 8
- d) by 1's
- e) 3
- f) 25
- g) 6 and 7
- h) 50
- i) yes
- j) 3
- k) none
- l) yes
- m) once
- n) 2.5 times
- o) once

ข้อ 19.

IDENTIFICATION DIVISION.
PROGRAM-ID. SAMPLE.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. 3100 MEMORY SIZE IS 16K WORDS.
OBJECT-COMPUTER. 3100 MEMORY SIZE IS 16K WORDS.
DATA DIVISION.
WORKING-STORAGE SECTION.

01 ITEM-NAME PICTURE X(24). (X(8) OR X(16) **also correct**)
01 QUANTITY-ON-HAND **PICTURE 9(8).**
01 VALUE-PER-UNIT PICTURE 9(5) V999.
01 TOTAL-VALUE PICTURE 9(5) **V999.**

PROCEDURE DIVISION.

6.3 Processor-directing statements

เป็นคำสั่งเดียวตามด้วย period และสั่งงานกับโคบอลคอมไพเลอร์โดยตรง ได้แก่ คำสั่ง ENTER, EXIT และ NOTE

คำสั่ง EXIT

คำสั่งนี้ ใช้แสดงจุดสุดท้ายที่ทำการ execute ซึ่งเป็นผลลัพธ์ของคำสั่ง PERFORM โดยมีรูปแบบดังนี้

EXIT.

ตัวอย่าง

PARA1

READ INPUT-FILE AT END GO TO PARA4

IF FIELD A LS 5, GO TO PARA3

PARA2

IF FIELD B GR 9, GO TO PARA4.

ADD FIELD D TO FIELD E GIVING ANSWER-FIELD

GO TO PARA4.

PARA3.

ADD FIELD A TO FIELD B GIVING ANSWER-FIELD

PARA4

EXIT.

PARA5.

PERFORM PARA1 THRU PARA4.

คำสั่ง NOTE

คำสั่งนี้ อนุญาตให้โปรแกรมเมอร์เขียนอธิบายคำสั่งต่าง ๆ ใน procedure division ของโปรแกรม แล้วเครื่องคอมพิวเตอร์จะพิมพ์ออกมาให้เห็นใน listing โดยไม่มีผลกระทบต่อ object program แต่อย่างใด โดยมีรูปแบบดังนี้

NOTE any-comment.

ตัวอย่าง 1

PARA-2.

NOTE THE MASTER FILE CONTAINS RECORDS FOR TWO
TYPES OF PROCESSING. RECORD TYPE A CONTAINS

PERSONNEL HISTORY. RECORD TYPE B CONTAINS
PERSONAL SALARY DATA.

ตัวอย่าง 2

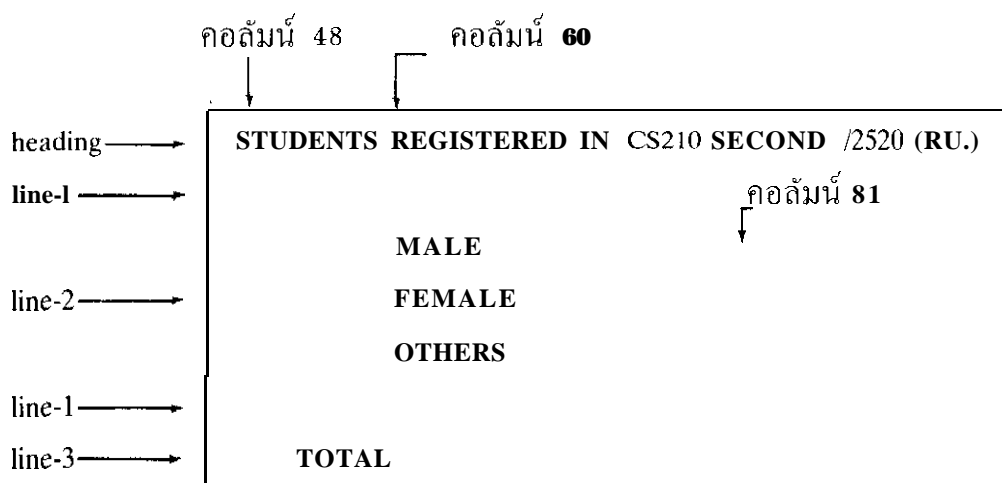
NOTE CONTROL AREA IS FIELD WITH 9'S AFTER LAST
RECORD OF MASTER FILE HAS BEEN READ.

ตัวอย่างโปรแกรม

1. จงเขียนโปรแกรมนับจำนวนนักศึกษาที่ลงทะเบียนเรียน วิชา CS 210 ในภาค 2/2520 แล้วแยกว่าเป็นเพศชายกี่คน หญิงกี่คน ข้อมูลของเราเป็นบัตรวิชา (Course cards) ทั้งหมดของนักศึกษาที่ลงทะเบียนเรียน ในภาค 2/2520 เก็บไว้ในเทป (1 บล็อก มี 20 เรคคอร์ด) โดยมีรูปแบบดังนี้

คอลัมน์ 1	มีรหัสเป็นเลข 5								
คอลัมน์ 2-9	รหัสนักศึกษา								
คอลัมน์ 10	เพศ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr> <td style="font-size: 3em; vertical-align: middle;">{</td> <td>0 ชายโสด</td> </tr> <tr> <td></td> <td>1 ชายแต่งงาน</td> </tr> <tr> <td></td> <td>2 หญิงแต่งงาน</td> </tr> <tr> <td></td> <td>3 หญิงโสด</td> </tr> </table>	{	0 ชายโสด		1 ชายแต่งงาน		2 หญิงแต่งงาน		3 หญิงโสด
{	0 ชายโสด								
	1 ชายแต่งงาน								
	2 หญิงแต่งงาน								
	3 หญิงโสด								
คอลัมน์ 11-35	ชื่อ นามสกุล								
36-40	วิชา								
41-80	อื่นๆ								

เข้าที่พุทพิมพ์ลงในกระดาษต่อเนื่อง ดังนี้



ขั้นตอนในการพิมพ์

- 1) ขึ้นหน้าใหม่แล้วพิมพ์หัวเรื่อง
- 2) เว้นสองบรรทัด
- 3) พิมพ์เข้าที่พุทเรคคอร์ดแต่ละบรรทัด
- 4) เว้นสองบรรทัด
- 5) พิมพ์ total line

เฉลย

IDENTIFICATION DIVISION.
PROGRAM-ID. CLASSIFICATION.
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 CC-FILE ASSIGN TO TAPE 03.
 SELECT PR-FILE ASSIGN TO PRINTER 61.
DATA DIVISION
FILE SECTION.
FD CC-FILE RECORDING MODE IS DECIMAL HIGH DENSITY
 LABEL RECORD IS STANDARD VALUE OF ID IS "TAPE-10002"
 BLOCK CONTAINS 20 RECORDS RECORD CONTAINS 80
 CHARACTERS DATA RECORD IS TAPE-REC.
01 TAPE--REC.
 03 FILLER PICTURE X(Y).
 03 SEX PICTURE 9.
 88 MALE VALUES ARE 0, 1.
 88 FEMALE VALUES ARE 2, 3.
 03 FILLER PICTURE X(25).
 03 CC-NO PICTURE X(S).
 88 CODE VALUE IS "CS210".
 03 FILLER PICTURE X(40).
FD PR-FILE LABEL RECORD IS OMITTED DATA RECORD IS OUT-REC.
01 OUT-REC PICTURE X(136).
WORKING-STORAGE SECTION.
77 M-CNT PICTURE 9(5).
77 F-CNT PICTURE 9(5).

77 N-MF-CNT PICTURE 9(5).
 77 T-CNT PICTURE 9(5).
 01 HEADING.
 03 FILLER PICTURE X(47) VALUE "1".
 03 FILLER PICTURE X(89) VALUE "STIDENTS REGISTERED
 IN CS210 SECOND /2520 (RU.)".
 01 LINE-1.
 03 FILLER PICTURE: X(136) VALLUE "0".
 01 LINE-2.
 03 FILLER PICTURE X(59) VALLUE SPACES.
 03 SEX-CODE PICTURE X(16).
 03 FILLER PICTURE X(15) VALUE SPACES.
 03 SEX-CNT PICTURE ZZ,ZZ9
 03 FILLER PICTURE X(50) VALUE SPACES.
 01 LINE-3.
 03 FILLER PICTURE X(51) VALUE SPACES.
 03 FILLER PICTURE X(5) VALUE "TOTAL".
 03 FILLER PICTURE X(23) VALUE SPACES.
 03 NO-CNT PICTURE ZZ,ZZ9.
 03 FILLER PICTURE X(61) VALUE SPACES.

PROCEDURE DIVISION.

0001.

OPEN INPUT CC-FILE, OUTPUT PR-FILE.
 MOVE SPACES TO OUT-REC.
 WRITE OUT-REC FROM HEADING.
 MOVE ZEROS TO M-CNT, F-CNT, N-MF-CNT, T-CNT

0002.

READ CC-FILE AT END GO TO 0003.
 IF CODE NEXT SENTENCE ELSE GO TO 0002.
 IF MALE ADD 1 TO M-CNT GO TO 0002.
 IF FEMALE ADD 1 TO F-CNT GO TO 0002.
 ADD 1 TO N-MF-CNT GO TO 0002.

0003.

WRITE OUT-REC FROM LINE-1.

0004.

MOVE M-CNT TO SEX-CNT.
 MOVE "MALE" TO SEX-CODE.
 WRITE OUT-REC FROM LINE-2.
 MOVE F-CNT TO SEX-CNT.
 MOVE "FEMALE" TO SEX-CODE.
 WRIE OUT-REC FROM LINE-2.
 MOVE N-MF-CNT TO SEX-CNT.
 MOVE "OTHERS" TO SEX-CODE.

WRITE OUT-REC FROM LINE-2.
 0005.
 PERFORM 0003.
0006.
 ADD M-CNT, F-CNT, N-MF-CNT GIVING T-CNT.
 MOVE T-CNT TO NO-CNT.
 WRITE OUT-REC FROM LINE-3.
0007.
 CLOSE CC-FILE, PR-FILE.
 STOP RUN.

2. ในการสอบย่อยวิชา CS 210 คราวหนึ่ง, กำหนดข้อสอบเป็นปรนัย ให้นักศึกษาเลือกคำตอบที่ถูกต้องที่สุดเพียง 1 ข้อ จากคำตอบทั้งหมด 5 ตัวเลือก ซึ่งมีหมายเลข 1, 2, 3, 4, และ 5 กำกับ ข้อสอบมีทั้งหมด 25 ข้อ แล้วให้เจาะคำตอบนี้ลงในบัตร 80 คอลัมน์ โดยมีรูปแบบดังนี้

คอลัมน์	ความหมาย	picture
1 - 8	รหัสประจำตัว	9(8)
9	คำตอบข้อ 1	9
10	คำตอบข้อ 2	9
11	คำตอบข้อ 3	9
.	.	.
.	.	.
33	คำตอบข้อ 25	9

- จงเขียน flowchart และโปรแกรมอ่านข้อมูลจากบัตรแล้วพิมพ์ผลลัพธ์ต่อไปนี้
- พิมพ์หัวเรื่องหนึ่งบรรทัด
 - รหัสประจำตัว, จำนวนข้อที่ตอบถูก, จำนวนข้อที่ตอบผิด และเปอร์เซ็นต์ของจำนวนข้อที่ตอบถูก
 - คะแนนเฉลี่ยของผลสอบในครั้งนี้

หมายเหตุ

บัตรข้อมูลใบแรกจะเป็นเฉลยคำตอบที่ถูกต้องทุกข้อในคอลัมน์ที่ 1-8 จะเป็นเลข 00000000 และคำตอบที่ถูกจะอยู่ในคอลัมน์ 9-33 บัตรใบต่อไปจึงเป็นบัตรคำตอบของนักศึกษาแต่ละคน

สูตรในการหาคะแนนเฉลี่ยมีดังนี้

$$\text{Mean} = \frac{\sum X_i}{N}$$

เมื่อ X_i = คะแนนสอบของนักศึกษาแต่ละคน ($i = 1, 2, 3, \dots, N$)

N = จำนวนนักศึกษาทั้งหมดที่เข้าสอบ

ตัวอย่าง

บัตรข้อมูล

000000001243134232323231421241313

192017822243124332422231421241314

207053033443144332341242421441331

207997424244134332322243422143313

เข้าที่พหุพิมพ์ออกมาในลักษณะข้างล่างนี้

STUDENT NO.	RIGHT	WRONG	PERCENT
19201782	19	6	76
20705303	14	11	56
20799742	16	9	64

MEAN = 14.17

00001 IDENTIFICATION DIVISION.
0002 PROGRAM-ID. JIM.
00003 ENVIRONMENT DIVISION.
00004 CONFIGURATION SECTION.
00005 SOURCE-COMPUTER. IBM370-135.
00006 OBJECT-COMPUTER. IBM370-135.
00007 INPUT-OUTPUT SECTION.
00008 FILE:-CONTROL.
00009 SELECT CARDFL ASSIGN TO SYS 004-UR-2540R-S.
00010 SELECT PRINTFL ASSIGN TO SYS 006-UR-1403-S.
uuu 11 DATA DIVISION.
00012 FILE: SECTION.
00013 FD CARDFL LABEL RECORD IS OMITTED DATA RECORD IS JEFF.
00014 01 JEFF.
00015 02 STD-NO PICTURE Y(S).
00016 02 SCORES PICTURE X(25).
00017 02 FILLER PICTURE X(47).
00018 FD PRINTFL LABEL RECORD IS OMITTED DATA RECORD IS T-REC.
00019 01 T-REC.
00020 02 FILLER PICTURE X(132).
00021 WORKING-STORAGE SECTION.
00022 77 EOF PICTURE X(3) VALUE 'NO'.
00023 77 P PICTURE YY VALUE ZERO.
00024 77 Q PICTURE 99 VALUE ZERO.
00025 77 TT PICTURE YY VALUE ZERO.
00026 77 FF PICTURE 99 VALUE ZERO.
00027 77 SUM-I PICTURE 999 VALUE ZERO.
00028 77 CENT-1 PICTURE 999 VALUE ZERO.
00029 77 AVER-1 PICTURE YYVYY VALUE ZERO.
00030 01 TOY-1.
00031 02 PEN-1 OCCURS 25 TIMES PICTURE X.
00032 01 TOY-2.
00033 02 PEN-2 OCCURS 25 TIMES PICTURE X.
00034 01 HEAD-1.
00035 02 FILLER PICTURE X(42) VALUE SPACES.
00036 02 FILLER PICTURE X(10) VALUE 'STUDENT NO'.
00037 02 FILLER PICTURE X(7) VALUE SPACES.
00038 02 FILLER PICTURE X(5) VALUE 'RIGHT'.
00039 02 FILLER PICTURE X(7) VALUE SPACES.
00040 02 FILLER PICTURE X(S) VALUE 'WRONG'.
00041 02 FILLER PICTURE X(7) VALUE SPACES.
00042 02 FILLER PICTURE X(7) VALUE 'PERCENT'.
00043 02 FILLER PICTURE X(42) VALUE SPACES.

00044 01 DETAIL-1.
00045 02 FILLER PICTURE X(43) VALUE SPACES.
00046 02 ID-NO PICTURE X(8).
00047 02 FILLER PICTURE X(10) VALUE SPACES.
00048 02 TT-1 PICTURE XX.
00049 02 FILLER PICTURE X(10) VALUE SPACES.
00050 02 FF-1 PICTURE XX.
00051 02 FILLER PICTURE X(10) VALUE SPACES.
00052 02 CENT-2 PICTURE XXX.
00053 02 FILLER PICTURE X(44) VALUE SPACES.
00054 01 DETAIL-2.
00055 02 FILLER PICTURE X(60) VALUE SPACES.
00056 02 FILLER PICTURE X(6) VALUE 'MEAN =',
00057 02 X-BAR PICTURE 999.99.
00058 02 FILLER PICTURE X(60) VALUE SPACES.
00059 PROCEDURE DIVISION.
00060 AAA.
00061 OPEN INPUT CARDFL OUTPUT PRINTFL.
00062 MOVE HEAD-1 TO T-REC. WRITE T-REC.
00063 READ CARDFL AT END MOVE 'YES' TO EOF.
00064 MOVE SCORES TO TOY-1.
00065 MOVE 0 TO P, MOVE 0 TO SUM-I.
00066 READ CARDFL AT END MOVE 'YES' TO EOF.
00067 PERFORM BBB THRU DDD UNTIL EOF = 'YES'
00068 PERFORM EEE.
00069 BBB.
00070 MOVE 0 TO TT, FF.
00071 ADD 1 TO P.
00072 MOVE SCORES TO TOY-2.
00073 MOVE 0 TO Q.
00074 PERFORM CCC 25 TIMES.
00075 DDD.
00076 COMPUTE CENT-1 = TT * 4.
00077 ADD TT TO SUM-I,
00078 MOVE STD-NO TO ID-NO.
00079 MOVE TT TO TT-1.
00080 MOVE FF TO FF-1.
00081 MOVE CENT-1 TO CENT-Z.
00082 MOVE DETAIL-1 TO T-REC.
00083 WRITE T-REC.
00084 READ CARDFL AT END MOVE 'YES' TO EOF.
00085 EEE.
00086 COMPUTE AVER-1 = SUM-I/P.

```

00087      MOVE AVER-1 TO X-BAR.
00088      MOVE DETAIL-2 TO T-REC
00089      WRITE T-REC.
00090      CLOSE CARDFL, PRINTFL.
00091      STOP RUN.
00092      ccc.
00093      ADD 1 TO Q.
00094      IF PEN-1 (0) = PEN-2(Q)
00095          ADD 1 TO TT
00096      ELSE
00097          ADD 1 TO FF.

```

STUDENT NO	RIGHT	WRONG	PERCENT
14520001	08	17	032
15520034	05	20	020
16520041	14	11	056
16520009	06	19	024
18520039	08	20	020
19520011	03	22	012
19520080	19	06	076
19520088	04	21	016
20520013	06	19	024
20520024	06	19	024
20520038	03	22	012
20520066	10	15	040
20520142	09	16	036
21520035	02	23	008

MEAN = 007.14

3. บริษัท “คุณไว้ใจเราได้” วางแผนที่จะใช้เครื่องคอมพิวเตอร์พิมพ์รายการฝาก/ถอนของลูกค้าในแต่ละเดือน โดยที่ลูกค้าแต่ละคนจะมีบัตรข้อมูลชุดหนึ่งมีรายการฝาก/ถอนเงินในเดือนนั้น เริ่มด้วยบัตรพิเศษ 1 ใบ มีชื่อลูกค้า ที่อยู่และตัวเลขเงินคงเหลือในบัญชีของเดือนก่อนหน้านั้น บัตรข้อมูลใบต่อไปจะเป็นบัตรที่มีรายการเปลี่ยนแปลงและจำนวนเงินที่เปลี่ยนแปลง ถ้าเป็นการถอนเงินตัวเลขจำนวนเงินที่เปลี่ยนแปลงจะมีเครื่องหมายลบกำกับอยู่ด้วย

ตัวอย่าง บัตรข้อมูล

บัตรใบที่ 1	MR.S. PHONGMITR	12/5 SUKHAPHIBAN 1	1,000.00
บัตรใบที่ 2	MRS. PHONGMITR	PAY CHECK	500.00
บัตรใบที่ 3	MR.S. PHONGMITR	MACYS	-50.00
บัตรใบที่ 4	MR.S. PHONGMITR	CASH WITHDRAWAL	-75.00
บัตรใบที่ 5	MRS. SAWATDEE	91/9 RAMKHAMHAENG ROAD	18,075.00

หมายเหตุ ถ้าข้อมูลลูกค้าเปลี่ยนแปลงแสดงว่าเป็นข้อมูลของลูกค้าคนต่อไป

จึงเขียน flow chart และโปรแกรมพิมพ์บัญชีของลูกค้าแต่ละคนโดยมีรูปแบบดังนี้

YOU-TRUST-ME COMPANY
88 RAMKHAMHAENG ROAD
HUAMARK, BANGKOK 10240

TO : MR.S. PHONGMITR

12/5 SUKHAPHIBAN 1 ROAD, BANGKOK

ITEM	DEPOSITS	WITHDRAWALS	BALANCE
OPENING BALANCE	10,000.00		1,000.00
PAY CHECK	500.00		1,500.00
MACYS		50.00	1,450.00
CASH WITHDRAWAL		75.00	1,375.00
INTEREST PAID	3.75		1,378.75

อัตราดอกเบี้ยคิด 1% ต่อเดือนเฉพาะยอดเงินคงเหลือสุดท้ายส่วนที่เกิน 1,000.00 บาท
เท่านั้น ถ้าบัตรข้อมูลลูกค้าเปลี่ยนไปให้พิมพ์เข้าที่ทุกในกระดาดแผ่นใหม่

IDENTIFICATION DIVISION.
PROGRAM-ID. PROGRAM 2.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-370.138.
OBJECT-COMPUTER. IBM-370.138.

```

INPUT-OUTPUT SECTION..
FILE-CONTROL.
    SELECT FILE-IN ASSIGN TO UT-S-SYSIN.
    SELECT FILE-OUT ASSIGN TO UT-1403.S-SYSPRINT.
DATA DIVISION.
FILE SECTION.
FD FILE-IN LABEL RECORD IS OMITTED DATA RECORD IS REC-IN.
01 REC.IN
    (15 FILLER      PIC X(7).
    05 I-1          PIC X(15).
    05 I-2          PIC X(42).
    05 I-3          PIC S9(5)V9(2).
    05 FILLER      PIC X(8).
FD FILE-OUT LABEL RECORD IS OMITTED DATA RECORD IS REC-OUT.
01 REC.OUT.
    05 FILLER      PIC X(132).
WORKING-STORAGE SECTION.
1   77 M 1          PIC S9(6)V9(2) VALUE ZEROES.
2   77 END-FILE    PIC X(3) VALUE "NO".
3   77 INT         PIC 9(4)V9(2) VALUE ZEROES.
01  L-1.
4   05 FILLER     PIC X(56) VALUE SPACES.
5   05 FILLER     PIC X(76) VALUE "YOU-TRUST-ME COMPANY".
01  L-2.
6   05 FILLER     PIC X(56) VALUE SPACES.
7   05 FILLER     PIC X(76) VALUE "88 RAMKHAMHAENG ROAD".
01  L-3.
8   05 FILLER     PIC X(56) VALUE SPACES.
9   05 FILLER     PIC X(76) VALUE "HUAMARK, BANGKOK".
01  L-4.
10  05 FILLER     PIC X(34) VALUE SPACES.
11  05 FILLER     PIC X(5) VALUE "TO : ".
    05 N-1        PIC X(93).
01  L-5.
12  05 FILLER     PIC X(39) VALUE SPACES.
    05 N-2        PIC X(93).
01  L-6.
13  05 FILLER     PIC X(39) VALUE SPACES.
14  05 FILLER     PIC X( 18) VALUE "ITEM".
15  05 FILLER     PIC X(13) VALUE "DEPOSITS".
16  05 FILLER     PIC X(20) VALUE "WITHDRAWAL".
17  05 FILLER     PIC X(42) VALUE "BALANCE".

```

```

01 L-7.
18     05 FILLER PIC X(32) VALUE SPACES.
19     05 N-3     PIC X(68) VALUE ALL " = "
20     05 FILLER PIC X(32) VALUE SPACES.
01 L-8.
21     05 FILLER PIC X(34) VALUE SPACES.
      05 O-1     PIC X(20).
      05 o-2     PIC .....,...9.99.
      05 FILLER PIC X(5).
      05 o-3     PIC ..... 9 9 .
      05 FILLER PIC X(5).
      05 o-4     PIC ....., 9 9 .
      OS FILLER PIC X(35).
22     PROCEDURE DIVISION.
      PRO-1.
23         OPEN INPUT FILE-IN OUTPUT FILE-OUT.
24         MOVE SPACES TO REC-OUT.
25         MOVE SPACES TO L-8.
26         READ FILE-IN AT END GO TO E-O-F.
      PRO-2.
2x         MOVE I-1 TO N-1.
29         MOVE I-2 TO N-2.
30         MOVE "OPENING BALANCE" TO O-1.
31         MOVE I-3 TO O-2.
32         MOVE I-3 TO O-4.
33         MOVE I-3 TO M-1.
34         WRITE REC.OUT FROM L-1 AFTER 3.
35         WRITE REC-OUT FROM L-2 AFTER 1.
36         WRITE REC-OUT FROM L-3 AFTER 1.
37         WRITE REC.OUT FROM L-4 AFTER 2.
38         WRITE REC-OUT FROM L-5 AFTER 1.
39         WRITE REC-OUT FROM L-6 AFTER 2.
40         WRITE REC-OUT FROM L-1 AFTER 2.
41         WRITE REC-OUT FROM L-8 AFTER 2.
      PRO-3.
42         MOVE SPACES TO L-X.
43         READ FILE-IN AT END MOVE "YES" TO END-FILE GO TO PRO-4.
46         IF I-I = N-1 GO TO PRO-6.
      PRO-4.
48         IF M-1 > 1000 GO TO PRO-X.
      PRO-S.
50         WRITE REC-OUT FROM L-7. AFTER 2.

```

```

51         IF END-FILE = "YES" GO TO E-O-F.
53         GO TO PRO-2.
PRO-h.
54         ADD I-3 TO M-1.
55         MOVE I-2 TO O-1.
56         MOVE M-1 TO O-4.
57         IF I-3 > 0 GO TO PRO-7.
49         COMPUTE O-3 = I-3 * (.1).
60         WRITE REC-OUT FKOM L-X AFTEK 2.
61         GO TO PRO-3.
PRO-7.
62         MOVE I-3 TO O-2.
63         WRITE REC-OUT FROM L-X AFTER 2.
64         GO TO PRO-3.
PRO-X.
65         COMPUTE INT = (M-I-1000) * 0.01.
66         ADD INT TO M-1.
67         MOVE INT TO O-2.
68         MOVE M-1 TO O-4.
69         MOVE "INTEREST PAID" TO O-1.
70         WRITE REC-OUT FROM L-X AFTER 2.
71         MOVE SPACES TO L-8.
72         GO TO PRO-S.
E-D-F.
73         CLOSE FILE-IN FILE-OUT.
74         STOP KUN.

```

**YOU-TRUST-ME COMPANY
88 RAMKHAMHAENG ROAD
HUAMARK, BANGKOK**

TO : AA AAA

111 SURIWONG ROAD

ITEM	DEPOSITS	WITHDRAWAL	BALANCE
OPENING BALANCE		15,000.00	15,000.00
PAY CHECK		10,000.00	25 ,000.00
INTEREST PAID		240.00	25,240.00

**YOU-TRUST-ME COMPANY
88 RAMKHAMHAENG ROAD
HUAMARK, BANGKOK**

TO : BB BBB

55 ASOKE ROAD

ITEM	DEPOSITS	WITHDRAWAL	BALANCE
OPENING BALANCE	30,000.00		30,000.00
MACYS		15,000.00	15,000.00
CASH WITHDRAWAL		11.500.00	3,500.00
INTEREST PAID	25.00		3,525.00

YOU-TRUST-ME COMPANY
88 RAMKHAMHAENG ROAD
HUAMARK, BANGKOK

TO : cc ccc

888 RAMKHAMHAENG ROAD

ITEM	DEPOSITS	WITHDRAWAL	BALANCE
OPENING BALANCE	95,000.00		95,000.00
MACYS		79,000.00	16,000.00
PAY CHECK	25,000.00		41,000.00
CASH WITHDRAWAL		20,000.00	21,000.00
INTEREST PAID	200.00		21,200.00

แบบฝึกหัด

1. บริษัทอุตสาหกรรมไทยพลาสติก จำกัด ได้ผลิตสินค้าตัวอย่างออกมา 5 ชนิด ต้นทุนในการผลิต ประกอบด้วย วัตถุดิบ ค่าแรงงาน และค่าใช้จ่ายสำหรับอุปกรณ์การผลิต ซึ่งตัวเลขต้นทุนการผลิตต่อหน่วยมีดังนี้

ชนิดของสินค้า	วัตถุดิบ	ค่าแรงงาน	ค่าใช้จ่ายสำหรับ อุปกรณ์การผลิต
A	\$4.00	\$6.00	\$6.00
B	\$2.50	\$7.00	\$3.50
C	\$2.00	\$7.50	\$7.00
D	\$1.75	\$3.45	\$3.45
E	\$0.75	\$7.00	\$3.50

แผนกวิเคราะห์ข้อมูลของบริษัทได้ทำการทดลองหลายวิธี เพื่อที่จะหาวิธีเลือก ซึ่งใช้ต้นทุนต่ำที่สุด ซึ่งมี 3 วิธี และได้ตัวเลขจำนวนสินค้าที่ผลิตได้ทั้งหมดในแต่ละวิธีดังนี้

ชนิดของสินค้า	วิธีที่ 1	วิธีที่ 2	วิธีที่ 3
A	1,000	2,000	3,000
B	1,000	500	1,000
C	2,000	1,000	500
D	6,000	5,000	4,000
E	8,000	8,000	9,000

ให้นักศึกษาเขียน flowchart และโปรแกรม

- คำนวณหาต้นทุนการผลิตสินค้าทั้ง 5 ชนิดในแต่ละวิธี
- ต้นทุนผลิตรวมในแต่ละวิธี
- วิธีใดที่จะผลิตสินค้าทั้ง 5 ชนิดนี้ โดยมีต้นทุนการผลิตต่ำที่สุด