

7.1 ข้อนำเสนอเบื้องต้น

สมัยก่อนนั้น เป้าหมายของการเขียนโปรแกรมคือความถูกต้อง (correctness), มีประสิทธิภาพ (efficiency) และมีความคิดสร้างสรรค์ (creativity) ทั้ง 3 หัวข้อนี้ ความถูกต้องเป็นเป้าหมายของการเขียนโปรแกรมเพียงอย่างเดียวที่ยังคงยอมรับกันอยู่ เรื่องมีประสิทธิภาพกับมีความสำคัญน้อยลง เพราะมีการประดิษฐ์เครื่องคอมพิวเตอร์ที่มีความเร็วในการทำงานสูงมาก และมีหน่วยความจำสำรองเก็บข้อมูล (virtual memory) ส่วนเรื่องความคิดสร้างสรรค์เป็นสิ่งที่ไม่เลวนักในตัวเอง ซึ่งเนื่องมาจากความฉลาดในตัวของแต่ละบุคคล แต่ก็มีบ่อยครั้ง ที่ทำให้ผลลัพธ์ออกมาเสียหาย

สิ่งแวดล้อมของการเขียนโปรแกรมในทุกวันนี้ มีการกำหนดเป้าหมายใหม่ ความถูกต้องยังคงมีความสำคัญเป็นอันดับแรก อย่างไรก็ตาม การบำรุงรักษา (maintainability) ซึ่งหมายถึงง่ายต่อการแก้ไข และอ่านได้ง่าย (readability) ซึ่งหมายถึงความชัดเจนของโปรแกรมถูกนำมาแทนที่ ประสิทธิภาพและความคิดสร้างสรรค์ของโปรแกรม สิ่งเหล่านี้เป็นลักษณะเฉพาะของโปรแกรมที่เราต้องการ

7.2 เขียนโปรแกรมโดยไม่ใช้คำสั่ง GO TO

Edsger W. Dijkstra เขียนเรื่อง "GO TO statement considered harmful" ในหนังสือ Communication of the ACM, vol 11, no 3 march 1968, p147 ไว้ว่า เป็นเวลาหลายปีที่เขาได้เฝ้าสังเกตว่า คุณภาพของโปรแกรมเมอร์ คือการลดฟังก์ชันของจำนวนการใช้คำสั่ง GO TO ในโปรแกรมที่เขาเขียน จนเมื่อไม่นานมานี้เอง เขาค้นพบว่า ทำไมการใช้คำสั่ง GO TO จึงได้ให้ผลลัพธ์เช่นนั้น เขาให้ข้อแนะนำว่า ควรจะขจัดคำสั่ง GO TO ออกไปให้หมดจากภาษาเขียนโปรแกรมระดับสูง ทุกภาษา ยกเว้นภาษาเครื่อง

Donald E. Knuth เขียนเรื่อง “GOTO is a four-letter word” เขาบอกว่า ไม่ควรให้หมีคำสั่ง GO TO ในการเขียนโปรแกรม

7.3 วัตถุประสงค์ของการเขียนโปรแกรมโครงสร้าง

เพื่อให้มีความเชื่อถือได้ (reliability) การบำรุงรักษา (maintenance) และความยืดหยุ่น (flexibility) สิ่งเหล่านี้เป็นปัจจัยสำคัญของการออกแบบโปรแกรม ในอดีตนั้นเมื่อเราจบการเขียนโปรแกรม และนำเอาไปใช้ มีหลายสิ่งที่เป็นสาเหตุทำให้ต้องมีการปรับปรุง (modify) โปรแกรม ซึ่งการปรับปรุงโปรแกรมส่วนมากแล้วเป็นเรื่องยาก เพราะว่าโปรแกรมนั้นเขียนไว้ไม่ดี (poorly designed and documented) และการเขียน สื่อความหมายเฉพาะผู้เขียนคนแรกเท่านั้นเอง ไม่สามารถเข้าใจได้โดยผู้อื่น

การเขียนโปรแกรมโครงสร้างได้พัฒนาขึ้นมาเพื่อแก้ไขสิ่งเหล่านี้ และปัญหาเบื้องต้นอื่น ๆ การเขียนโปรแกรมโครงสร้างเป็นสัญลักษณ์หรือภาษาซึ่งจะให้ความเชื่อมั่นแก่โปรแกรมเมอร์ ในการออกแบบโปรแกรม ไม่ใช่เพียงแต่เขียนโปรแกรม

7.4 คำจำกัดความ

Edward Yourdon บอกว่า การเขียนโปรแกรมโครงสร้าง เป็น “เซตของกฎเกณฑ์และข้อจำกัดของการเขียนโปรแกรม ทำให้โปรแกรมนั้นอยู่ในรูปแบบที่เหมาะสมเป็นการจัดสิ่งซึ่งไม่ใช่วัตถุประสงค์ของเรา, การอ่านโปรแกรมยาก และความยุ่งยากที่จะนำไปสู่หยาบ (bug) แต่เป็นการเพิ่มการทดสอบและการบำรุงรักษา”

7.5 วิธีการออกแบบโปรแกรม (Methodology program design)

เมื่อไม่กี่ปีมานี้ การเขียนโปรแกรมโครงสร้าง ขอมรับกันว่า เป็นเทคโนโลยีของการเขียนโปรแกรมขั้นสูง, มีเครื่องมือ (tools) หลากอย่าง พัฒนาขึ้นมาเพื่อการนี้, โปรแกรมเมอร์จะใช้วิธีการเขียนโปรแกรมโครงสร้าง ดังนี้

(1) The Jackson method of detail program design

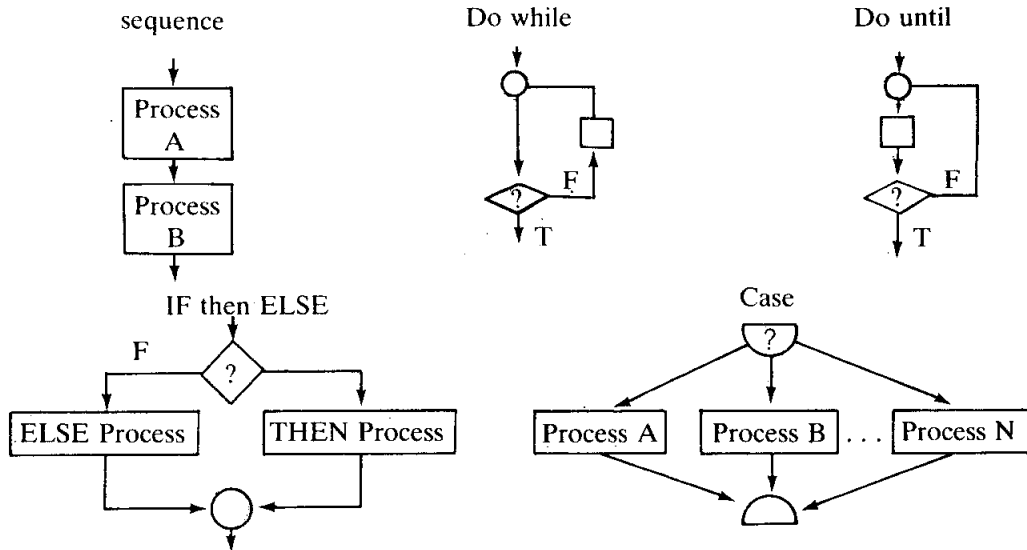
(2) Nassi-Shneiderman charts

An alternative to flowchart for design

(3) Top down development using a program design language.

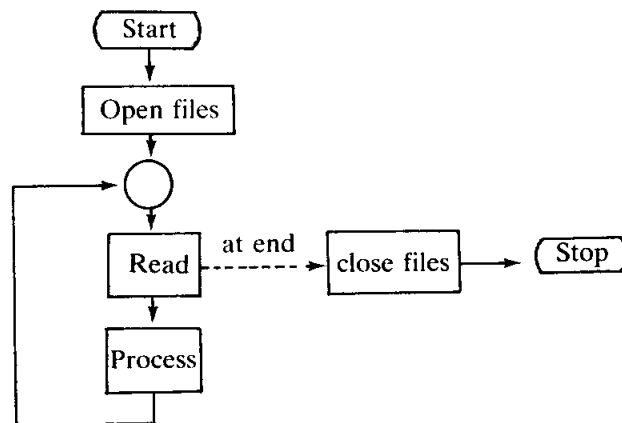
เทคนิคของแต่ละวิธีข้างต้นนี้ มีทั้งข้อดีและข้อไม่ดี แต่ละวิธีเป็นประโยชน์ในสถานการณ์อย่างใดอย่างหนึ่ง

7.6 แนวความคิดของการเขียนโปรแกรมโครงสร้าง

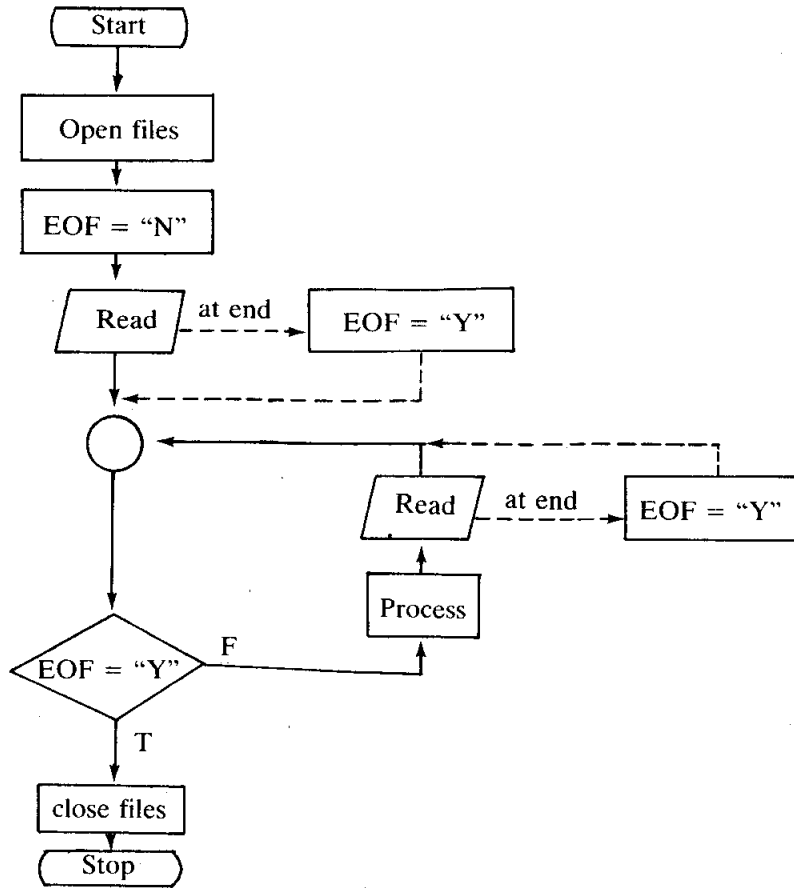


ทั้งหมดนี้เป็นกลุ่มของวิธีการเขียนมาตรฐาน เพื่อให้ประสบความสำเร็จในเป้าหมายของการเขียนโปรแกรมโครงสร้าง

ตัวอย่าง

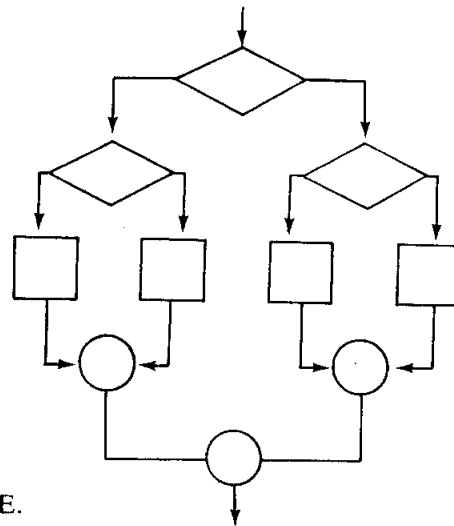


ตัวอย่าง



ตัวอย่าง การเขียน flowchart จาก 1 conditional sentence ที่กำหนดให้

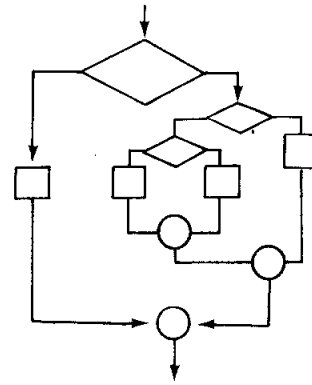
1. a) IF TOTAL PINS-DOWN = 10
 - IF BALL-NUMBER = 1
 - PERFORM PROCESS-STRIKE
 - ELSE
 - PERFORM PROCESS-SPARE
 - ELSE
 - IF BALL-NUMBER = 1
 - PERFORM PROCESS-FIRST-BALL
 - ELSE
 - PERFORM PROCESS-END-OF-FRAME.



b) IF PAYMENT-LATE OR NOT COMPLETE

```

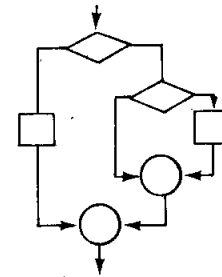
IF FIRST-TIME
  PERFORM GENTLE-REMAINDER
ELSE
  IF PREVIOUS-LETTER-SENT
    PERFORM THREATENING-LETTER
  ELSE
    PERFORM SECOND-REMAINDER
ELSE
  PERFORM THANK-YOU-LETTER.
  
```



c) IF DATE-FIELD NUMERIC

```

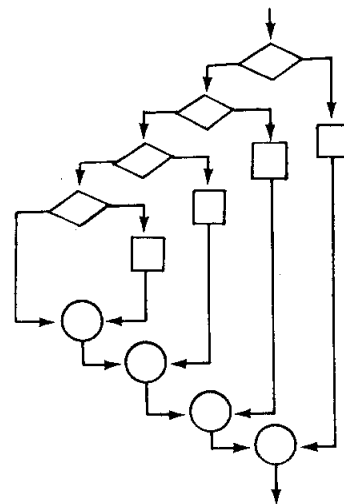
IF DATE-FIELD < 791125
  MOVE "DATE TOO SMALL" TO ERROR-MSG
  WRITE PRINT-LINE
ELSE
  NEXT SENTENCE
ELSE
  MOVE "DATE NOT NUMERIC" TO ERROR-MSG
  WRITE PRINT-LINE.
  
```



d) IF CUSTOMER-TYPE = "C"

```

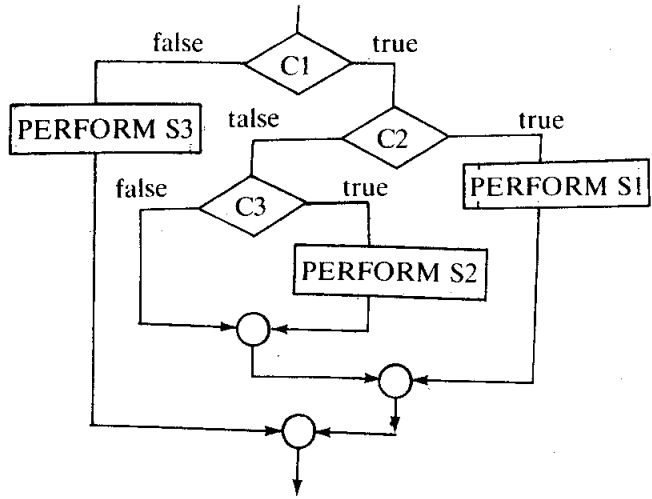
PERFORM CASH-ACCOUNT
ELSE
  IF CUSTOMER-TYPE = "T"
    PERFORM THIRDTY-DAY-ACCOUNT
  ELSE
    IF CUSTOMER-TYPE = "R"
      PERFORM REVOLVING-ACCOUNT
    ELSE
      IF CUSTOMER-TYPE = "S"
        PERFORM SPECIAL-ACCOUNT.
  
```



2. a) จงเขียน flowchart จากคำสั่งข้างล่างนี้

```

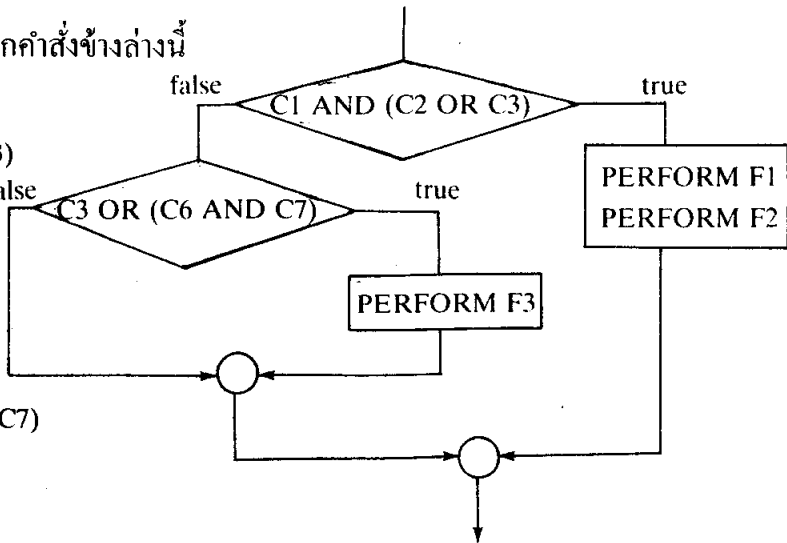
IF C1
  IF C2
    PERFORM S1
  ELSE
    IF C3
      PERFORM S2
    ELSE
      NEXT SENTENCE
ELSE
  PERFORM S3
  
```



b) จงเขียน flowchart จากคำสั่งข้างล่างนี้

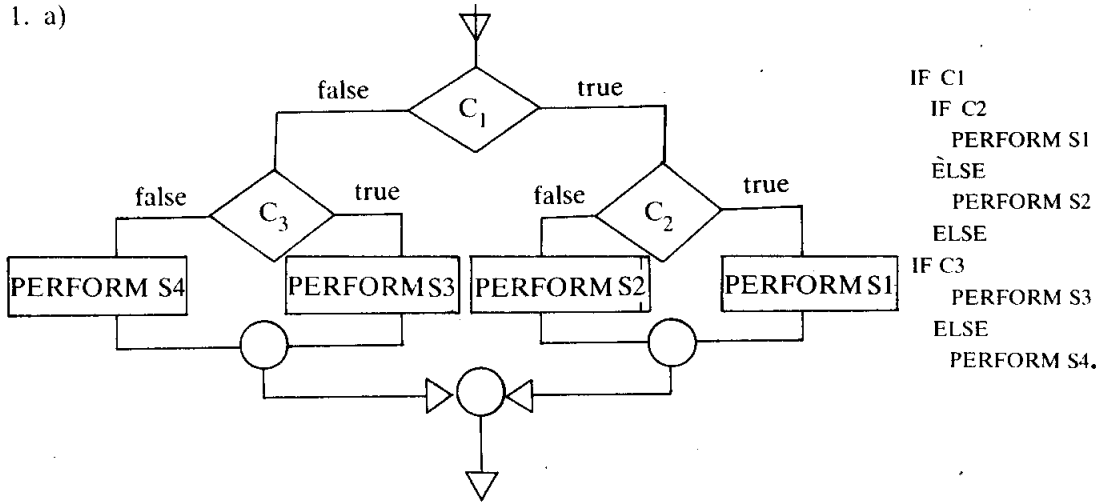
```

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



ตัวอย่าง การเขียน 1 conditional sentence จาก flowchart ที่กำหนดให้

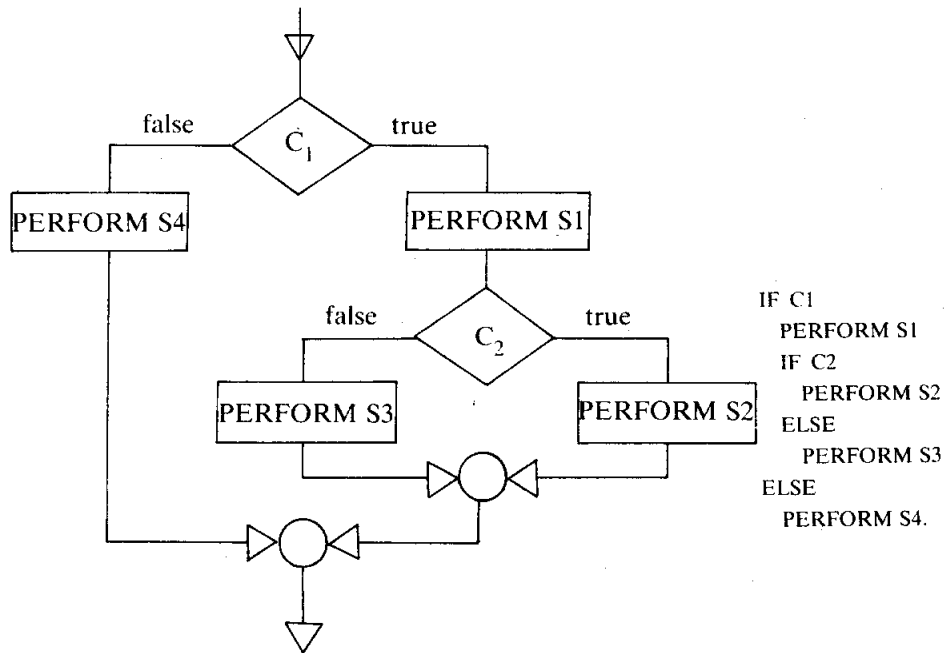
1. a)



```

IF C1
  IF C2
    PERFORM S1
  ELSE
    PERFORM S2
  ELSE
    IF C3
      PERFORM S3
    ELSE
      PERFORM S4.
    
```

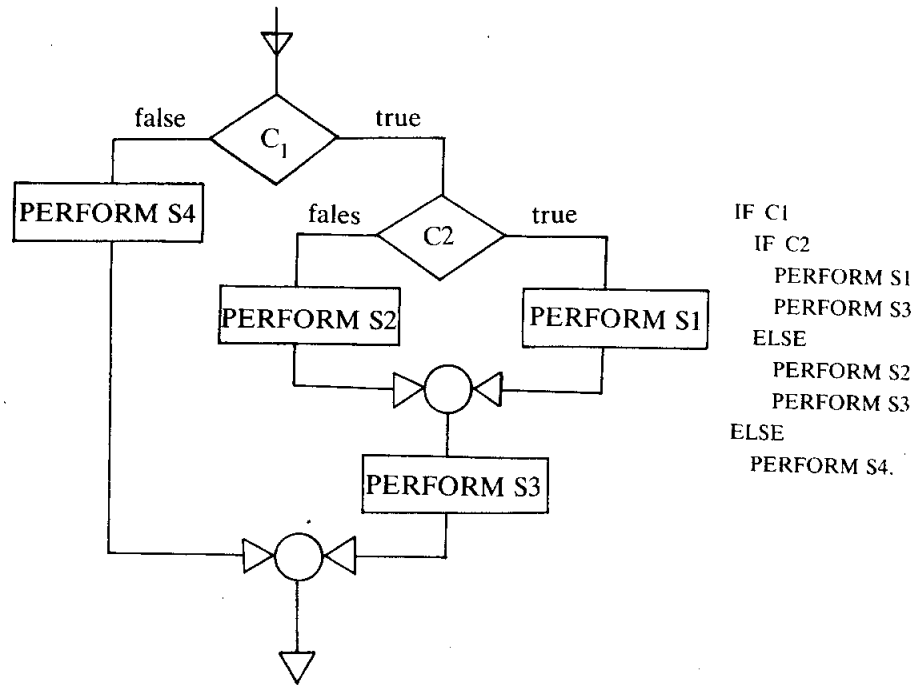
b)



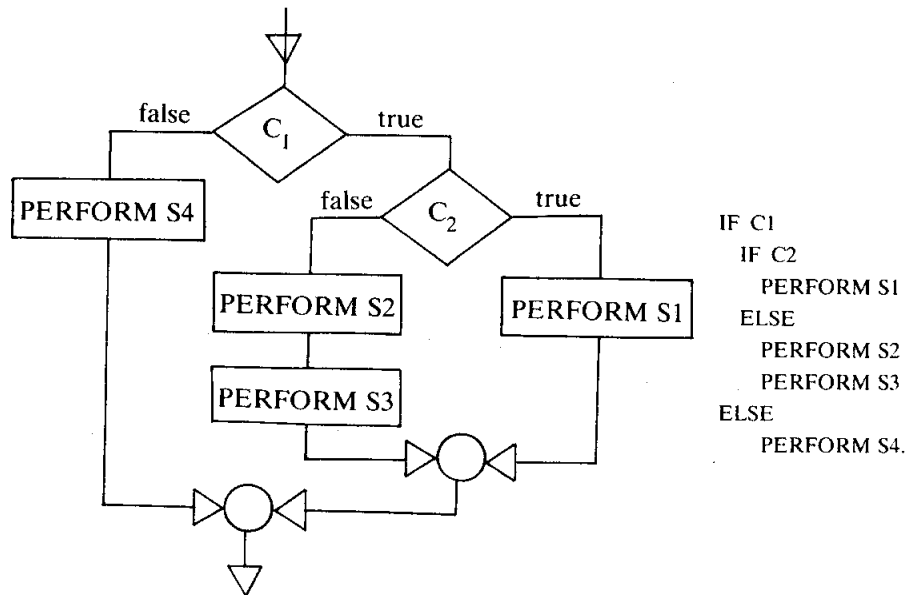
```

IF C1
  PERFORM S1
  IF C2
    PERFORM S2
  ELSE
    PERFORM S3
  ELSE
    PERFORM S4.
    
```

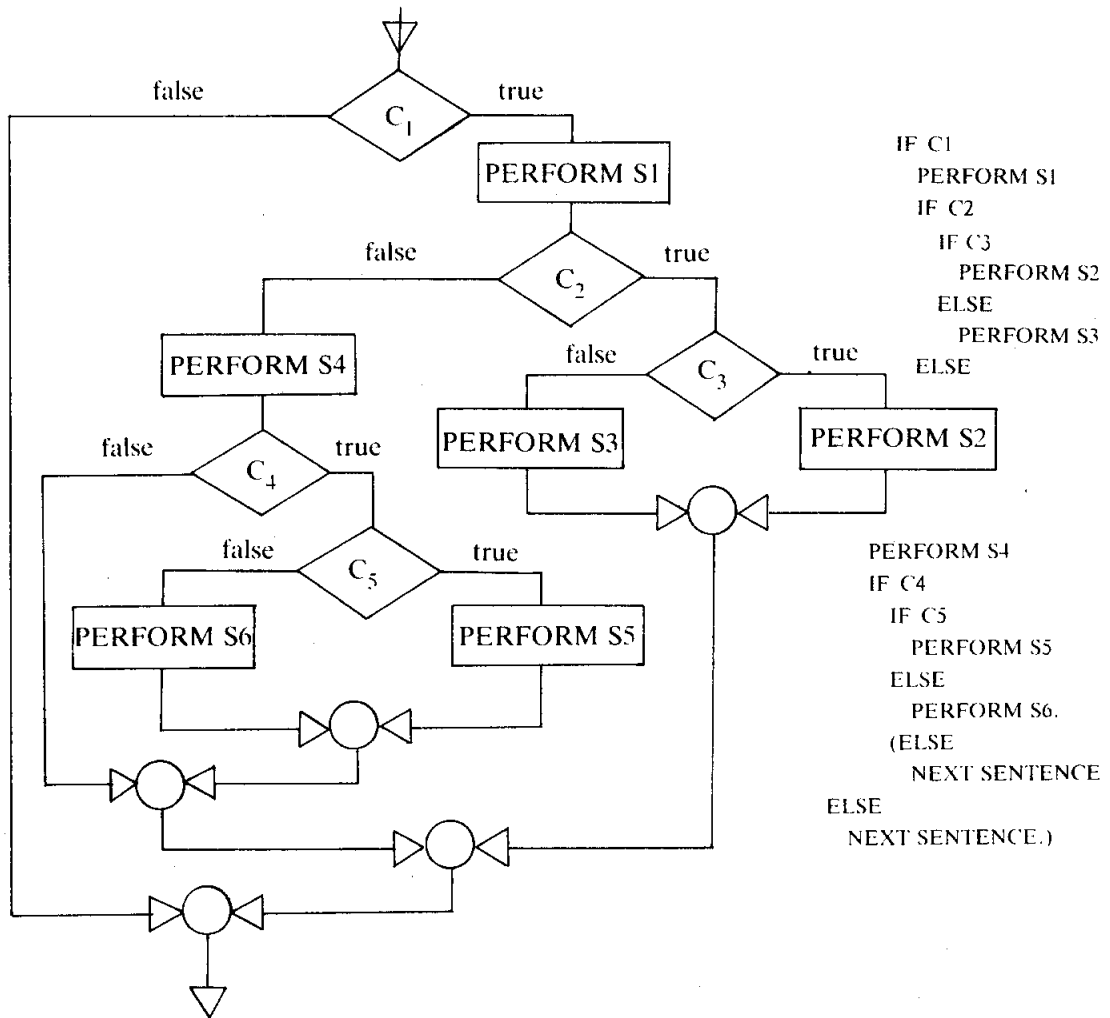
c)



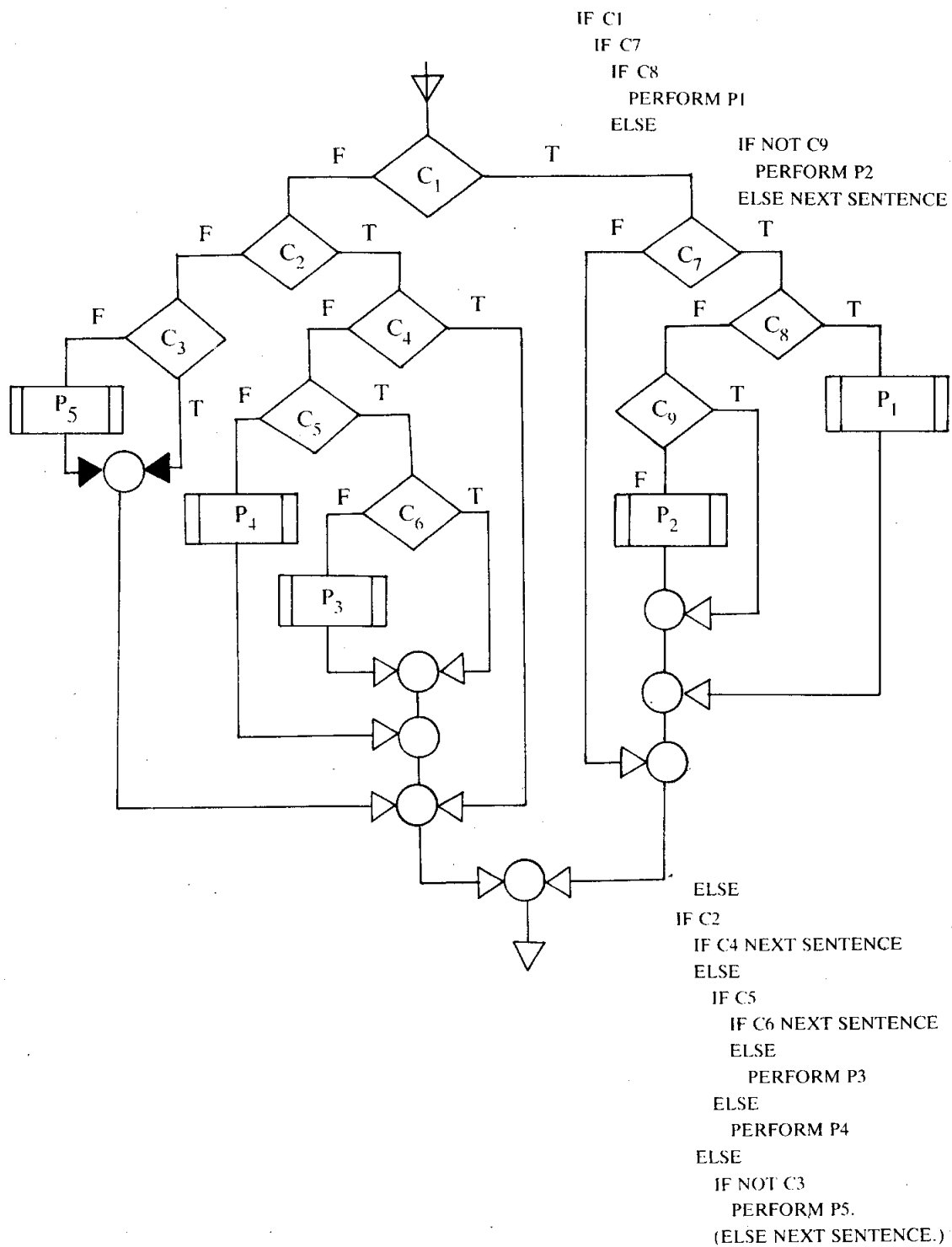
d)



2. จงเขียน 1 conditional sentence จาก flowchart ข้างล่างนี้

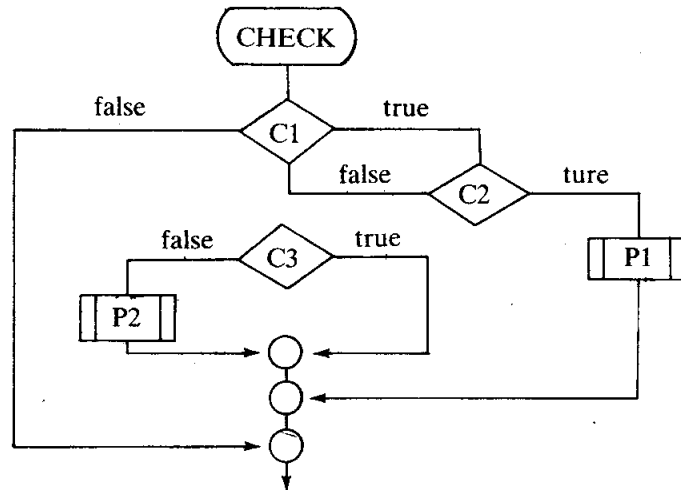


3. จงเขียน 1 conditional sentence จาก flowchart ข้างล่างนี้ เมื่อ $C_1, C_2 \dots C_9$ เป็นชื่อมีเงื่อนไข (condition names) และ P_1, P_2, \dots, P_5 เป็นชื่อพารากราฟ (paragraph names)

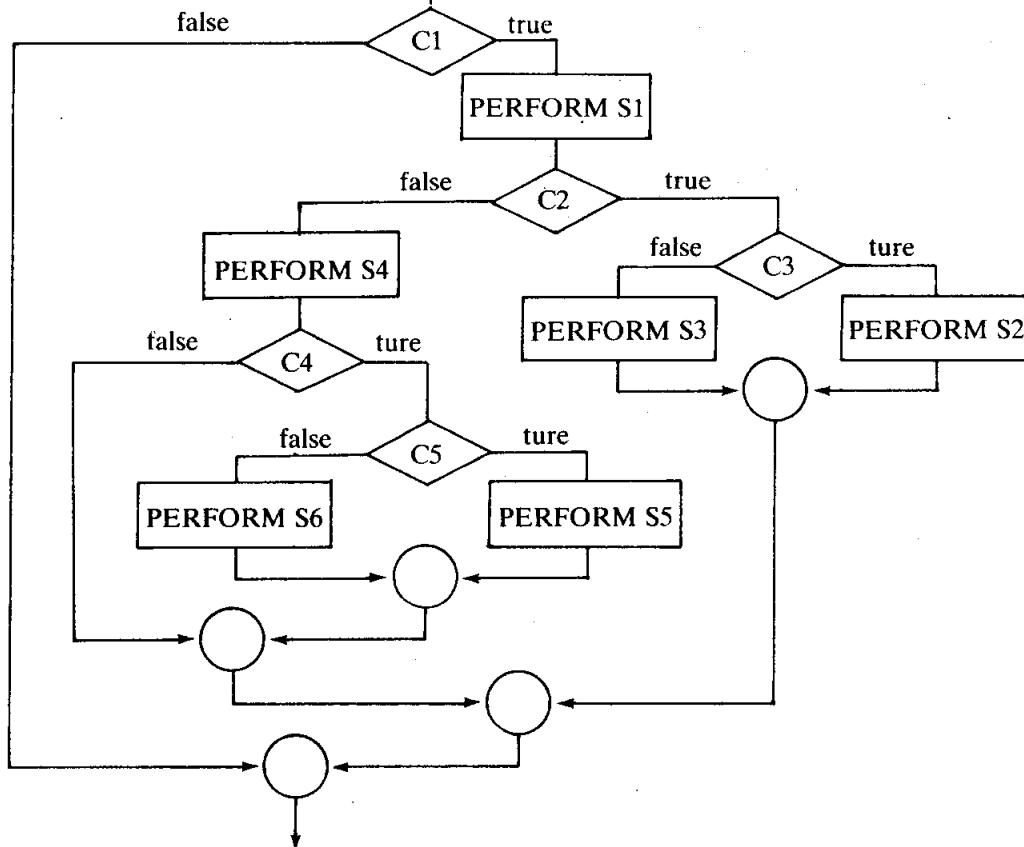


แบบฝึกหัด

1. จาก flowchart ข้างล่างนี้ จงเขียน 1 conditional sentence เมื่อ C_1, C_2, C_3 เป็นชื่อเงื่อนไข (condition-names) และ $P_1, P_2, CHECK$ เป็นชื่อพารากราฟ (paragraph-names)



2. จงเขียน 1 conditional sentence จาก flowchart ข้างล่างนี้



2. Data description entries ข้างล่างนี้ เป็นข้อมูลของผู้สมัครเข้าทำงานแต่ละคน

- 02 EDUCATION PICTURE 99.
 - 88 H-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.

บริษัทต้องการพนักงานเฉพาะที่มีคุณสมบัติข้อใดข้อหนึ่งใน 3 ข้อต่อไปนี้

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-REQUIREMENT