

บทที่ 4

ขบวนการสะสม (Cumulative process)

- 4.1 โปรแกรมตัวอย่าง
- 4.2 คำสั่งในภาษาฟอร์แทรน
 - 4.2.1 คำสั่ง DO
 - 4.2.2 คำสั่ง CONTINUE
 - 4.2.3 การโยกย้ายเข้าไปและออกจากลูป
 - 4.2.4 การซ้อนกันของลูป
 - 4.2.5 สัญลักษณ์ในการเขียนแผนผังสายงานสำหรับคำสั่ง DO
 - 4.2.6 ความสัมพันธ์ระหว่างรายชื่อตัวแปรและรายการ format code
- 4.3 โปรแกรมตัวอย่างเพิ่มเติม
 - 4.3.1 การคำนวณค่า $N!$ (N factorial)
 - 4.3.2 ความเบี่ยงเบนมาตรฐาน (Standard deviation)
 - 4.3.3 การคำนวณค่าไซน์ของ X (sine(x))
 - 4.3.4 คอกเบซทบต้น

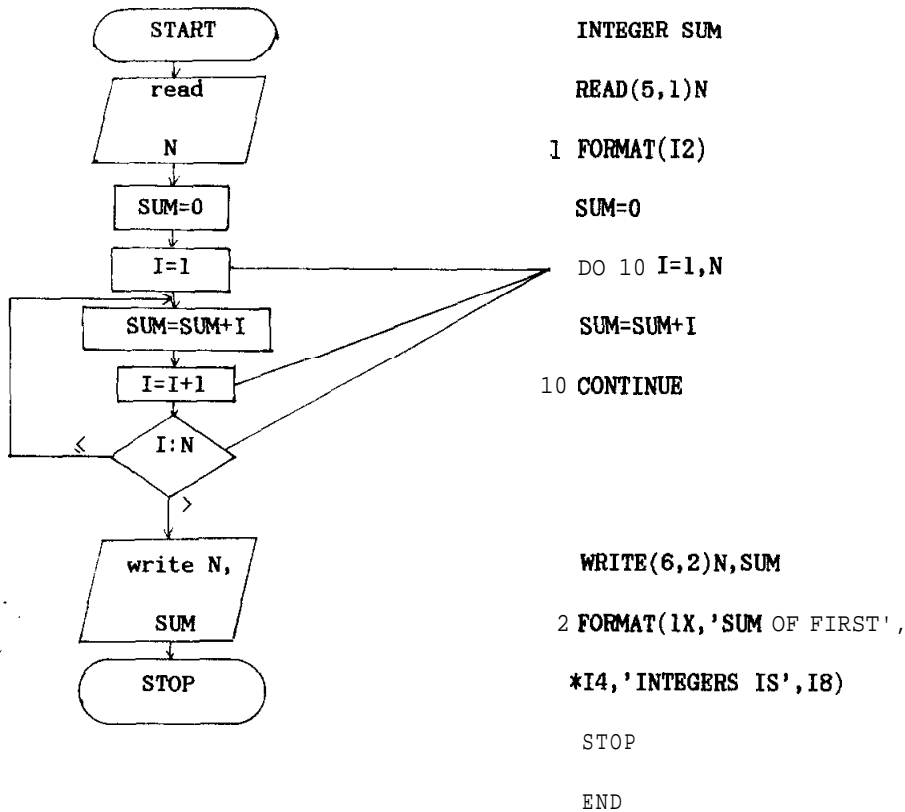
แบบฝึกหัดที่ 4

บทที่ 4

ขบวนการสะสม (Cumulative process)

4.1 โปรแกรมตัวอย่าง

จงเขียนโปรแกรมเพื่อหาผลบวกของเลขจำนวนเต็ม N ตัวแรก โดยให้อ่านค่า N จากบัตร (โปรแกรมใช้คำสั่ง DO เป็นคำสั่งสร้างเลข 1,2,...,N)



คำสั่งใหม่จากตัวอย่างนี้คือ คำสั่ง DO และคำสั่ง CONTINUE

4.2 คำสั่งในภาษาฟอร์แทรน

4.2.1 คำสั่ง DO

คำสั่ง DO นั้นใช้ในการควบคุมการทำงานของลูป วิธีการควบคุมก็คือทำให้ตัวแปรที่ใช้ควบคุมมีค่าเริ่มต้น (initial value) มีส่วนที่เปลี่ยน (increment) และมีค่าทดสอบ (tested value) ที่จะใช้ควบคุมการออกจากลูป

คำสั่ง DO จะกำหนดค่าเริ่มต้น ส่วนที่เปลี่ยน ค่าทดสอบของตัวแปรนี้ (counter หรือ index) และเรนจ์ของ DO ซึ่งทุกคำสั่งในเรนจ์ของ DO จะถูกทำซ้ำตามจำนวนครั้งที่ระบุไว้
ต่อไปนี้

DO เลขประจำคำสั่ง **ตัวแปร=ค่าเริ่มต้น,ค่าทดสอบ**[;ส่วนที่เปลี่ยน]

โดยที่ DO เป็นคีย์เวิร์ด

เลขประจำคำสั่ง คือเลขประจำคำสั่งของคำสั่งสุดท้ายของลูป

ตัวแปร เป็นตัวแปรชนิด integer บางครั้งเราเรียกว่า ตัวแปรของคำสั่ง DO (DO variable) ซึ่งใช้เป็นตัวนับจำนวนครั้งที่คำสั่งในลูปจะถูกทำซ้ำ

ค่าเริ่มต้น เป็นค่าคงที่จำนวนเต็มบวกหรือตัวแปรชนิด integer ซึ่งจะเป็นค่าเริ่มต้นของตัวแปร (สำหรับฟอร์แทรนคอมพิวเตอร์เลขบางตัว ค่าเริ่มต้นอาจอยู่ในรูปนิพจน์เลขคณิตชนิด integer หรือ real ก็ได้)

ค่าทดสอบ เป็นค่าคงที่จำนวนเต็มบวกหรือตัวแปรชนิด integer ซึ่งค่าของตัวแปรนี้จะถูกเปรียบเทียบกับค่าทดสอบว่าถ้ามันมีค่ามากกว่าค่าทดสอบ เราจะออกจากลูป (สำหรับฟอร์แทรนคอมพิวเตอร์เลขบางตัว ค่าทดสอบอาจอยู่ในรูปนิพจน์เลขคณิตชนิด integer หรือ real ก็ได้)

ส่วนที่เปลี่ยน จะมีหรือไม่มีก็ได้ เป็นค่าคงที่จำนวนเต็มบวกหรือตัวแปรชนิด integer ซึ่งเป็นค่าที่จะบวกเพิ่มขึ้นแก่ตัวแปรหลังจากทำงานในลูปแล้วแต่ละครั้ง ถ้าไม่มีส่วนที่เปลี่ยนไว้ ส่วนที่เปลี่ยนเท่ากับ 1

ตัวอย่าง

คำสั่ง- DO ที่ถูกต้อง

DO 10 J=1,303

DO 5 I=4,1000,66

DO 4 KSUM=M1,M2,M3

DO 6 LL=NSTAR,MITY

DO 8 K=10,M,4

ตัวอย่าง

คำสั่ง DO ที่ไม่ถูกต้อง และเหตุผลที่ทำให้ไม่ถูกต้อง

DO 3 (X)=I,3,4

ตัวแปรต้องเป็นค่าแปรชนิด integer

DO 4 I=1,20,(−1)

ส่วนที่เปลี่ยนต้องเป็นบวก

DO 6 (Z=M+X),(−3),T*2

ตัวแปรชนิด และใช้ฟังก์ชันเลขคณิตไม่ได้

DO 6 K=I,J,(TER)

ส่วนที่เปลี่ยนต้องเป็น integer

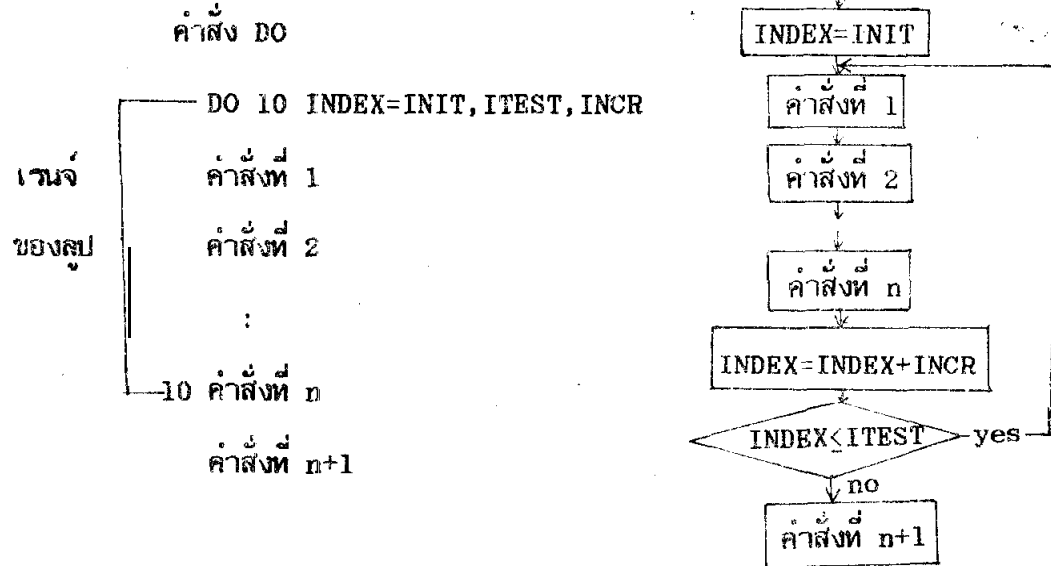
DO 8 L=(1.1),10,(−.1)

ต้องเป็นเลขจำนวนเต็ม

DO 9 K=(I+1),10

ใช้ฟังก์ชันเลขคณิตไม่ได้

หมายเหตุ ในคำสั่ง DO ที่ไม่ถูกต้องบางคำสั่งนั้น สำหรับฟอร์แทรนคอมพิวเตอร์บางตัวอาจใช้ได้ ผู้ใช้ต้องตรวจสอบกับหนังสือคู่มือการเขียนภาษาฟอร์แทรนของเครื่องนั้น ๆ



คำสั่งในแนวของลูปจะถูกทำอย่างน้อย 1 ครั้ง คำสั่งในลูปเป็นคำสั่งปฏิบัติการ คำสั่งสุดท้ายของลูปต้องไม่เป็นคำสั่ง STOP คำสั่ง IF เลขคณิต คำสั่ง GO TO คำสั่ง COMPUTED GO TO คำสั่ง DO อื่น ๆ หรือคำสั่ง END แต่อย่างไรก็ตามเป็นคำสั่ง IF ตรวจจับได้

จำนวนครั้งที่คำสั่งในลูปจะถูกทำซ้ำ (N)

$$N = (ITEST - INIT) / INCR + 1$$

ตัวอย่าง DO 16 I=3,100,2
 ISQR=I*I
 SQROOT=I**.5
 15 WRITE(6,1)I,ISQR,SQROOT
 1 FORMAT(2X,I3,I7,F6.2)

ซึ่งจะสร้างตารางหาค่ากำลังสองและรากที่ 2 ของเลข 3,5,7,...,99

ตัวอย่าง READ(5,1)N
 DO 15 K=1,N

16 _____

ในคำสั่ง DO ไม่ระบุส่วนที่เปลี่ยน ดังนั้นส่วนที่เปลี่ยนมีค่า=1 คำสั่งในแขนงของลูปจะถูกทำ N ครั้ง
สิ่งที่ไม่ให้ใช้ในคำสั่ง DO

1. คำสั่งสุดท้ายของลูปต้องไม่เป็นคำสั่งเหล่านี้

คำสั่ง IF เลขคณิต

คำสั่ง DO

คำสั่ง STOP

คำสั่ง GO TO ทั้ง COMPUTED GO TO และ GO TO แบบไม่มีเงื่อนไข

คำสั่ง ไม่ปฏิบัติการ

ตัวอย่าง ทศกฏ

DO 15 I=1,N

:

13 _____

:

15 IF(I-15)13,17,13 ← คำสั่ง IF เลขคณิต

17 WRITE(6,1)I

2. กรรมนี้และคำสั่ง 3 คำของมันจะถูกเปลี่ยนโดยคำสั่งในสลับไม่ได้

DO 15 J=1,10,3

J=J+1 <--- ใช้ไม่ได้

15 S=S*J

3. ระวังการใช้ค่าของกรรมนี้ของคำสั่ง DO ถ้าการออกจากสลับเป็นไปอย่างปกติคือ ออกจากสลับเมื่อค่าของกรรมนี้เกินค่าทดสอบ นั่นคือค่าของกรรมนี้จะไม่เท่ากับค่าทดสอบ กรรมนี้ ไม่แนะนำให้เอาค่าของกรรมนี้ไปใช้

ตัวอย่าง DO 5 I=1,10] ----> คำสั่ง DO จะทำให้พิมพ์เลข 1,2,...,10
5 WRITE(6,6)I]

WRITE(6,6)I ----> I=11 ซึ่งถ้าต้องการใช้ค่า I=10 จะทำให้เกิดข้อผิดพลาดได้
การออกจากสลับอาจออกโดยคำสั่งควบคุมซึ่งอยู่ในสลับก็ได้ และค่าของกรรมนี้คือค่าของมัน
ก่อนออกจากสลับนั่นเอง

ตัวอย่าง DO 10 I=1,20

IF(I.GE.11)GO TO 5

10 X=X+1

:

5 WRITE(6,11)I ----> ค่าของ I ที่จะพิมพ์คือ 11

4.2.2 คำสั่ง CONTINUE

รูปทั่วไป

| | |
|----------------|----------|
| เลขประจำคำสั่ง | CONTINUE |
|----------------|----------|

คำสั่ง CONTINUE มักใช้เป็นคำสั่งสุดท้ายของสลับ เพื่อหลีกเลี่ยงการที่คำสั่งสุดท้ายของสลับ
จะเป็นคำสั่งที่ห้ามไว้ดังกล่าวข้างต้น คำสั่ง CONTINUE ปรากฏที่ใด ๆ ในโปรแกรมก็ได้ โดยจะ ไม่
ทำให้การปฏิบัติงานหยุดชะงักแต่อย่างใด

ตัวอย่าง

ไม่ถูกต้อง

```

C=0
DO 10 I=1,100
- 5- READ(5,1)GRADE
      IF(GRADE.GT.90)C=C+1
10 GO TO 5  ---> วนซ้ำไม่ได้
16 X=1

```

ถูกต้อง

```

C=0
DO 10 I=1,100
      READ(5,1)GRADE
      IF(GRADE.GT.90)C=C+1
10 CONTINUE
16 X=1

```

ตัวอย่าง

ไม่ถูกต้อง

```

C=0
DO 10 I=1,100
5 READ(5,1)GRADE<-----
      IF(GRADE.LE.90)GO TO 5
10 C=C+1

```

ถูกต้อง

```

C=0
DO 10 I=1,100
      READ(5,1)GRADE
      IF(GRADE.GT.90)C=C+1
10 CONTINUE

```

กรณีค่าของ I จะไม่เพิ่มขึ้นเพราะคำสั่งสุดท้ายของลูปไม่ถูกทำ

ตัวอย่าง

ไม่ถูกต้อง

```

C=0
1 DO 10 I=1,100
      READ (5,11) GRADE
      IF(GRADE.LE.90)GO TO 1
10 C=C+1

```

ทุกครั้งที่มี $GRADE \leq 90$ การทำงานจะกลับไปคำสั่งเลขที่ 1 คือคำสั่ง DO กรณีนี้จะทำให้ค่าของ I ถูกกำหนดค่าใหม่เป็น 1 การกระทำดังกล่าวจะเกิดลูปอนันต์ (infinite loop) ขึ้น

ตัวอย่าง ถ้าต้องการนับและพิมพ์คะแนนที่เกิน 90 คะแนน

ไม่ถูกต้อง

C=0

DO 10 I=1,100

READ(5,1)GRADE

IF(GRADE.GT.90)GO TO 5

→10 CONTINUE

5 C=C+1

WRITE(6,6)GRADE

GO TO 10

ถูกต้อง

C=0

DO 10 I=1,100

READ(5,1)GRADE

IF(GRADE.GT.90)GO 1-0 5

Go To 10

5 C=C+1

WRITE(6,6)GRADE

10 CONTINUE

การเขียน

C=0

DO 10 I=1,100

READ(5,1)GRADE

IF(GRADE.LE.90)GO TO 10

C=C+1

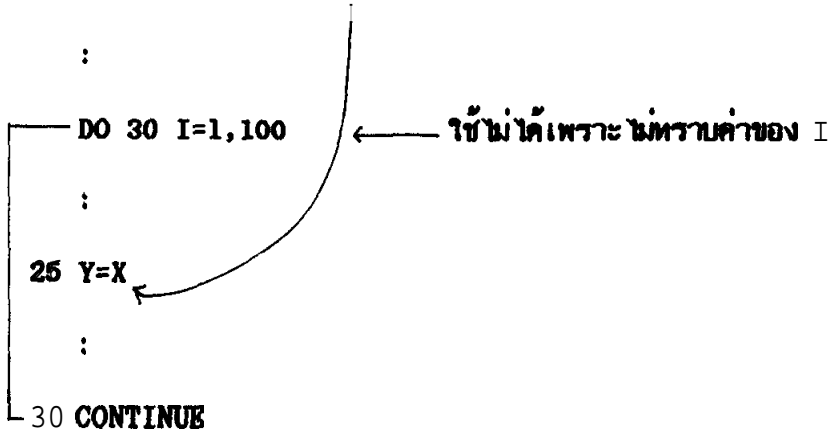
WRITE(6,6)GRADE

10 CONTINUE

4.2.3 การโยกย้ายเข้าไปและออกจากสปี

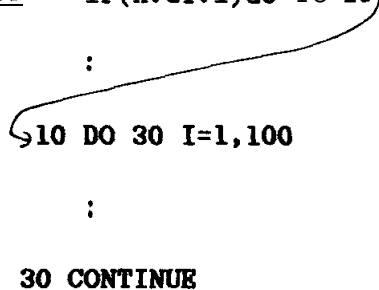
กรณี 1 ห้ามโยกย้ายจากคำสั่งนอกลูปเข้าไปยังคำสั่งใด ๆ ในแขนงของลูป

ตัวอย่าง IF(X.GT.1)GO TO 26



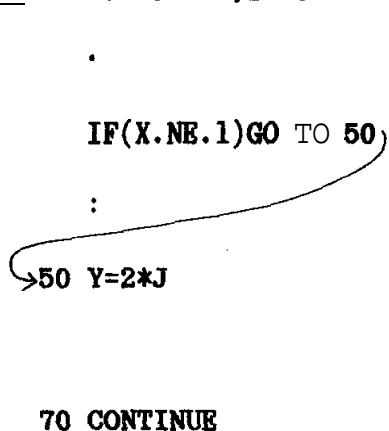
กรณี 2 การโยกย้ายเข้าสู่ลูปโดยเข้าที่คำสั่ง Do นั้นทำได้

ตัวอย่าง IF(X.GT.1)GO TO 10)



กรณี 3 การโยกย้ายระหว่างคำสั่งภายในแขนงของลูปด้วยกันใช้ได้

ตัวอย่าง DO 70 J=1,1000



กรณี 4 การโยกย้ายคำสั่งในลูปออกไปนอกลูปนั้นใช้ได้

ตัวอย่าง DC 10 JOE=1,10

```
IF(Y.EQ.6)GO TO 70
10 CONTINUE
70 DO 16 I=1,JOE
:
```

กรณี 5 การโยกย้ายจากคำสั่งในลูปไปยังคำสั่ง DO ของมันจะทำให้เกิดการกำหนดค่าเริ่มต้นของตัวแปรใหม่แทนที่จะเพิ่มค่าของตัวแปรตามส่วนที่เปลี่ยนค่า ๆ ไป ผลของตัวอย่างข้างล่างจะได้ลูปอนันต์

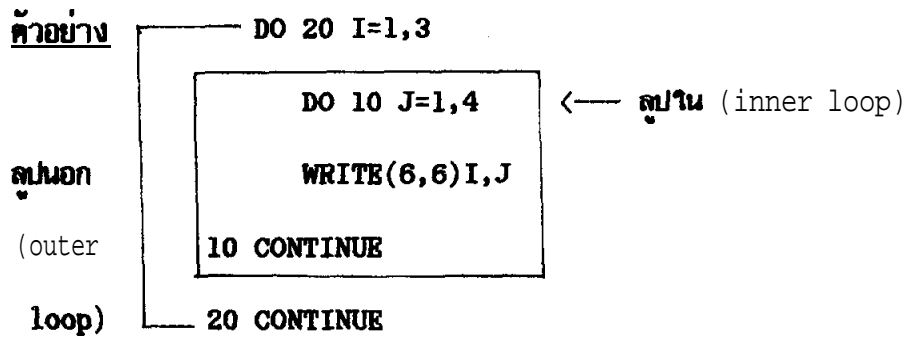
```
4 DO 10 I=1,10
X=I**2-7*I+12
IF(X.NE.0)GO TO 4 (ต้องให้ GO TO 10)
WRITE(6,1)I
10 CONTINUE
```

| I | X |
|---|---|
| 1 | 6 |
| 1 | 6 |
| 1 | 6 |
| : | : |
| : | : |

4.2.4 การซ้อนกันของลูป (nested loop)

การซ้อนกันของลูปเกิดขึ้นเมื่อมีเงื่อนไขของลูปหนึ่งอยู่เป็นส่วนหนึ่งของคำสั่งในเงื่อนไขของอีก

ลูปหนึ่ง



ผลกาารพิมพ์

| I | J |
|---|---|
| 1 | 1 |
| 1 | 2 |
| 1 | 3 |
| 1 | 4 |
| 2 | 1 |
| 2 | 2 |
| 2 | 3 |
| 2 | 4 |
| 3 | 1 |
| 3 | 2 |
| 3 | 3 |
| 3 | 4 |

คำสั่งข้างต้นอาจเขียน โดยที่ให้ลูปนอกและลูปในใช้คำสั่งสุดท้ายร่วมกัน ได้ดังนี้

```

DO 20 I=1,3
  DO 20 J=1,4
    WRITE(6,3)I,J
  20 CONTINUE

```

ตัวอย่าง

```

DO 6 J=6,10
  DO 7 K=6,12
    KPROD=J*K
    WRITE(6,1)J,K,KPROD
  7 CONTINUE
  WRITE(6,2)
5 CONTINUE
STOP
1 FORMAT(2X,I2,'*',I2,'=',I3)
2 FORMAT(/)
END

```

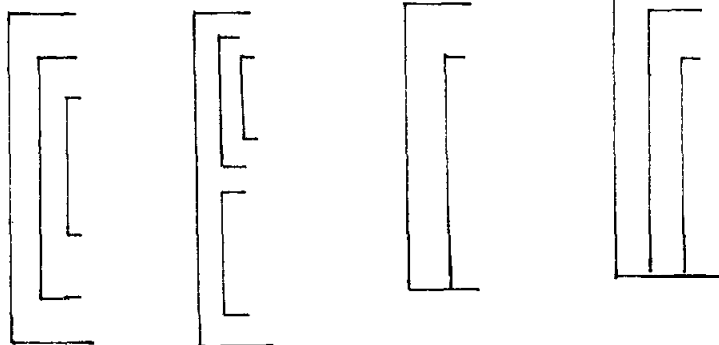
ผลที่ได้

```

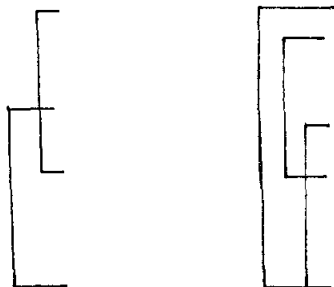
6* 6= 36
6* 7= 42
:
6*12= 72 ← เว้น 2 บรรทัด
7* 6= 42
:
7*12= 64
:
10* 6= 60 ← เว้น 2 บรรทัด
:
10*12=120

```

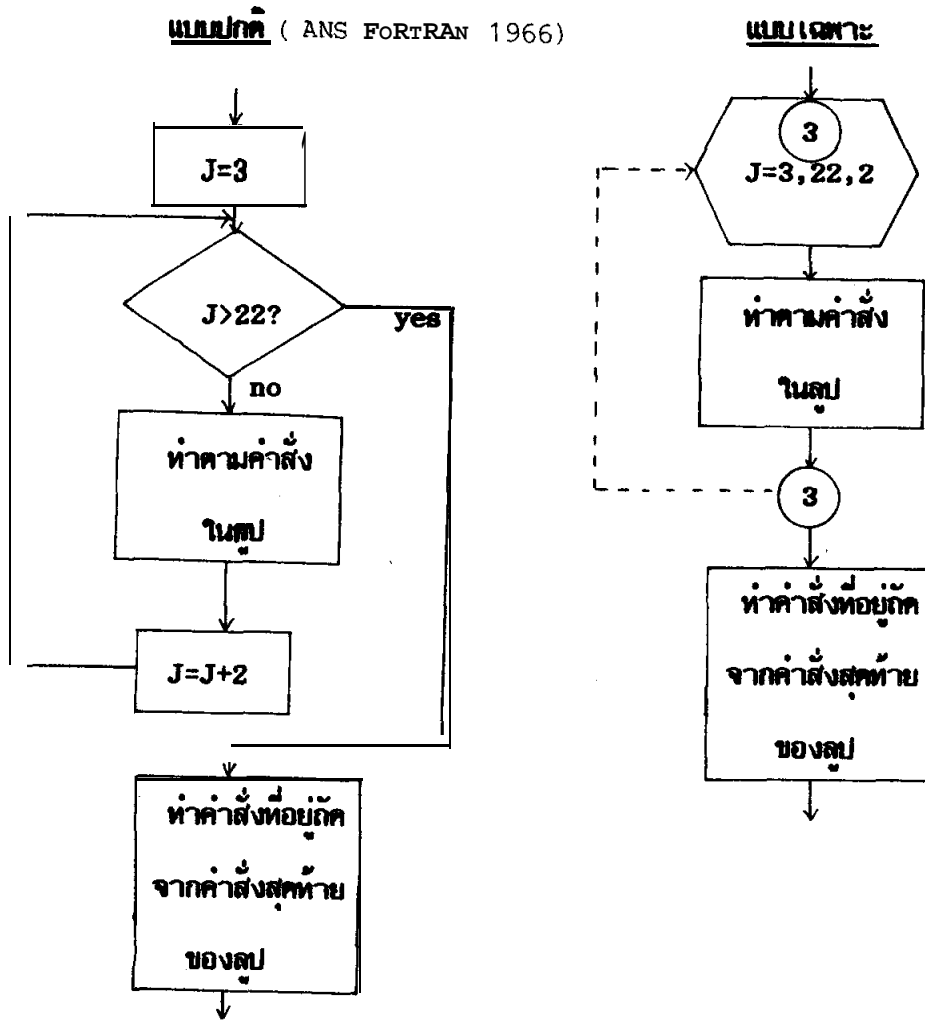
โครงสร้างของการซ้อนกันของสเปซที่ใช้ได้



โครงสร้างของการซ้อนกันของสเปซที่ใช้ไม่ได้



4.2.5 สัตถกณ์ในการเขียนแผนมิตายงานสำหรับคำสั่ง DO



4.2.6 ความสัมพันธ์ระหว่างรายชื่อตัวแปรและรายการ format code

ในคำสั่ง READ คำสั่ง WRITE และคำสั่ง FORMAT นั้น ถ้าจำนวนตัวแปรน้อยกว่าจำนวน format code format code ที่เกิดมาจะไม่ถูกใช้

ตัวอย่าง

READ(5,5)IX,IY

5 FORMAT(I3,I2,I4,I5)

ไม่ถูกใช้

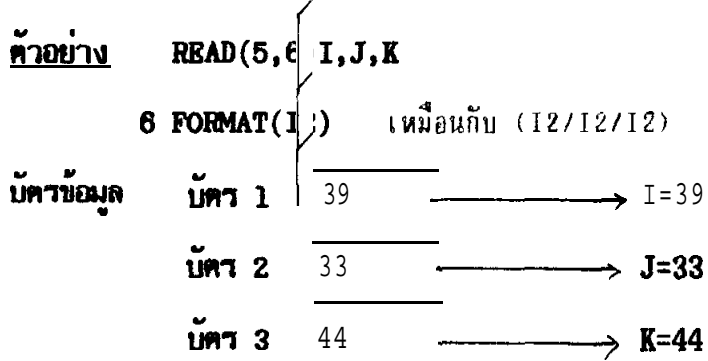
บัตรข้อมูล

| | | |
|-----|----|----|
| 123 | 45 | 67 |
|-----|----|----|

IX=123

IY=45

ถ้าจำนวนตัวแปรมีมากกว่า format code เมื่อใช้ format code หมดแล้วมันจะย้อนกลับมาใช้ format code ในวงเล็บเปิดแรกที่ยังเหลือ และจะมีการขึ้นระเบียน (record) ใหม่ เช่น ขึ้นบัตรใหม่ในการอ่านบัตรหรือขึ้นบรรทัดใหม่ในการพิมพ์

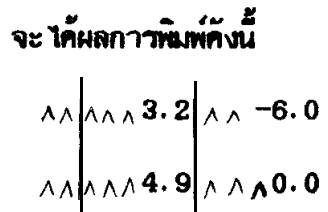


ตัวอย่าง WRITE(6,10)A,B,C,D

10 FORMAT(2X,2F6.1) เหมือนกับ (2X,2F6.1/2X,2F6.1)

ค่าที่กำหนดให้

- A=3.2
- B=-6.0
- C=4.9
- D=0.



4.3 โปรแกรมตัวอย่างเพิ่มเติม

4.3.1 การคำนวณค่า N! (Nfactorial)

N! อ่านว่า N แฟกทอเรียล ถ้า N เป็นเลขจำนวนเต็มบวกแล้ว

$$N! = N(N-1)(N-2) \dots 2.1$$

```

INTEGER PROD
PROD=1
READ(5,5)N
5 FORMAT(I2)
DO 2 I=1,N
PROD=PROD*I
2 CONTINUE
WRITE(6,4)N,PROD
4 FORMAT(1X,' FACTORIAL ',I3,' IS ',I16)
STOP
END

```

ค่าแฟกทอเรียลเพิ่มขึ้นอย่างรวดเร็ว คอมพิวเตอร์หลาย ๆ ระบบไม่สามารถหาค่าเกิน 17! ได้

4.3.2 ส่วนเบี่ยงเบนมาตรฐาน (Standard deviation)

สูตรทั่วไปในการคำนวณความเบี่ยงเบนมาตรฐานของคะแนน n คะแนน

x_1, x_2, \dots, x_n คือ

$$sd = \sqrt{\frac{n(x_1^2 + x_2^2 + \dots + x_n^2) - (x_1 + x_2 + \dots + x_n)^2}{n(n-1)}}$$

จงเขียนโปรแกรมเพื่ออ่านคะแนน 30 คะแนนจากบัตรข้อมูล (บัตร 1 บัตรมีคะแนน

หนึ่งตัว) คำนวณค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐาน

```

SUM=0
SUMSQ=0
DO 5 K=1,30
READ(5,1)GRADE
SUM=SUM+GRADE

```

```
SUMSQ=SUMSQ+GRADE**2
```

```
5 CONTINUE
```

```
N=30
```

```
DEV=((N*SUMSQ-SUM**2)/(N*(N-1)))**.5
```

```
AV=SUM/N
```

```
WRITE(6,20)DEV,AV
```

```
20 FORMAT(T9,'STANDARD DEVIATION=',F5.2//
```

```
*T9,'AVERAGE=',F5.1)
```

```
STOP
```

```
1 FOIWT(F5.1)
```

```
END
```

4.3.3 การคำนวณค่าไซน์ของ X (sine(x))

สูตรในการประมาณค่าไซน์ของ X โดยที่ X เป็นมุมเรเดียน คือ

$$\sin(x) = \frac{x^1}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots - \frac{x^n}{n!}$$

จงเขียนโปรแกรมเพื่อประมาณค่า sin(x) จากสูตรข้างต้น

วิธีที่ 1

```
READ(5,3)X
```

```
3 FORMAT(F3.0)
```

```
XSQ=X*X
```

```
S=0
```

```
IFACT=1
```

```
DO 10 I=1,6
```

```
S=S+X/IFACT
```

```
X=X*XSQ
```

```
10 IFACT=-IFACT*(2*I)*(2*I+1)
```


WRITE(6,6)S

6 FORMAT(4X,F8.4)

STOP

END

ข้อ 2

DATA K,SUM,N/1,0.,1/

READ(5,5)X

5 FORMAT(F3.0)

X=-X

20 SUM=SUM+(-X)**N/K

N=N+2

K=K*(N-1)*N

IF(N.LE.11)GO TO 20

WRITE(6,6)SUM

6 FORMAT(F6.4)

STOP

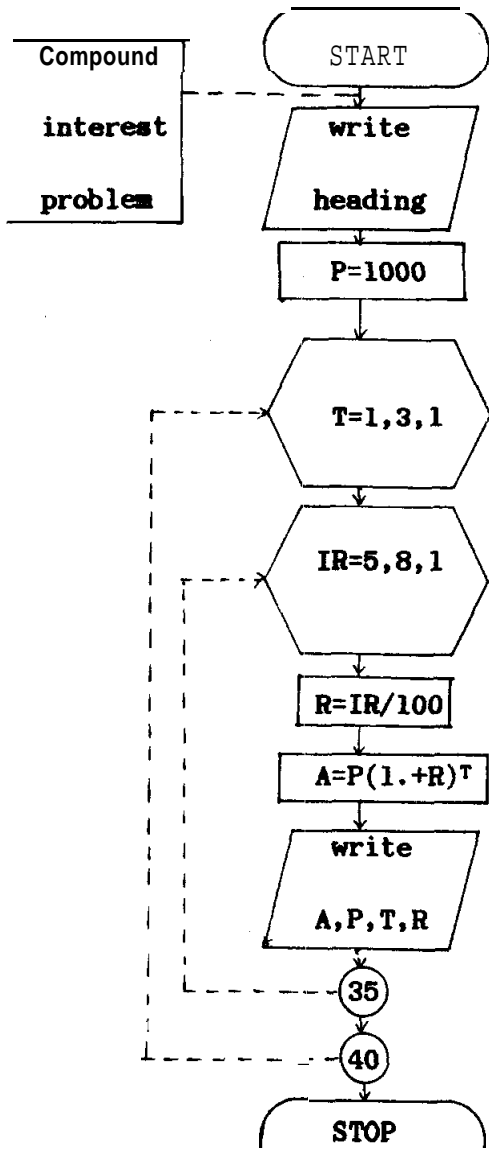
END

4.3.4 ดอกเบี้ยทบต้น

เงินต้น P ฝากประจำเป็นเวลา T ปี ด้วยอัตราดอกเบี้ย R ยอดเงินจากการคำนวณดอกเบี้ยทบต้นเป็น A ซึ่งหาได้จากสูตร

$$A=P(1+R)^T$$

จงเขียนโปรแกรมเพื่อคำนวณเงินที่จะมีในบัญชี (A) ถ้ากำหนดเงินต้น \$1000 ด้วยอัตราดอกเบี้ย 5%, 6%, 7% และ 8% และฝากเป็นเวลา 1, 2, 3 ปี
แผนภูมิสายงานและโปรแกรมเพื่อแก้ปัญหาหาคือ



C COMPOUND INTEREST PROBLEM

INTEGER T

WRITE(6,1)

1 FORMAT(T3,'AMOUNT',T10,
* 'PRINCIPAL',T20,'YEARS',
*T30,'INTEREST')

P=1000

DO 40 T=1,3

DO 35 IR=5,8,1

R=IR/100.

A=P*(1.+R)**T

WRITE(6,2)A,P,T,R

2 FORMAT(T2,F7.2,T11,F5.0,T23,
*I1,T32,F3.2)

35 CONTINUE

40 CONTINUE

STOP

END

ਅੰਕ ੧੧

| AMOUNT | PRINCIPAL | YEARS | INTEREST |
|---------------|------------------|--------------|-----------------|
| 1060.00 | 1000. | 1 | .05 |
| 1060.00 | 1000. | 1 | .06 |
| 1070.00 | 1000. | 1 | .07 |
| 1080.00 | 1000. | 1 | .08 |
| 1120.60 | 1000. | 2 | .05 |
| 1123.60 | 1000. | 2 | .06 |
| 1144.90 | 1000. | 2 | .07 |
| 1188.40 | 1000. | 2 | .08 |
| 1157.82 | 1000. | 3 | .05 |
| 1191.01 | 1000. | 3 | .06 |
| 1225.04 | 1000. | 3 | .07 |
| 1259.71 | 1000. | 3 | .08 |

แบบฝึกหัดที่ 4

1. DO loop ต่อไปนี้ข้อใดถูกต้อง ถ้าผิดเหตุใดจึงผิด

1.1) DO 10 I=1,5

10 CONTINUE

1.3) DO 30 K=1,10

DO 40 K=1,40

40 CONTINUE

30 CONTINUE

1.5) DO 50 I-1=2,6

:

50 CONTINUE

1.7) DO 60 I=1,5

DO 70 J=1,10

60 CONTINUE

70 X=1

1.9) DO 90 L=1,5

DO 91 K=2,8

S=S+G

90 CONTINUE

1.2) DO 20 IJ=2,6.5

20 CONTINUE

1.4) DO 15 I=I,6

15 CONTINUE

1.6) DO 60 I=1,5

DO 70 J=1,10,-1

70 CONTINUE

60 CONTINUE

1.6) DO 60 A=I,N

60 CONTINUE

1.10) DO 21 M=10,6,1

21 CONTINUE

| | |
|--|--|
| <p>1.11) Do 5 K=1,5 : 5 IF(L-80)3,3,2</p> | <p>1.12) DO 20 L=8,1 DO 30 K=1,3 L=L+1 30 CONTINUE 20 CONTINUE</p> |
| <p>1.13) Do 5 I=1,3 IF(Q.EQ.6)GO TO 5 Do 5 K=1,10 SUM=SUM+J 5 CONTINUE</p> | <p>1.14) DO 6 I=1,10 J=I*I+1 6 CONTINUE</p> |
| <p>1.15) DO 8 K=1,6 READ(5,5)K,K1 SUM1=SUM1+K1 8 SUM2=SUM2+K</p> | <p>1.16) DO 10 I=1,17 15 X=X+SUM IF(X.GT.10)GO TO 17 10 CONTINUE 17 IF(SUM.NE.J)GO TO 15</p> |
| <p>1.17) DO 9 J=40,60 DO 9 K=1,3 9 CONTINUE</p> | <p>1.18) Do 85 I=1,4 : CONTINUE 85 DO 90 J=3,6,1 90 X=10.3</p> |

2. จงแสดงผลการพิมพ์จากคำสั่งต่อไปนี้

2.1) DO 5 I=1,10,2

5 WRITE(6,1)I

1 FORMAT(2X,I3)

2.3) DO 8 K=4,4

8 WRITE(6,3)K

3 FORMAT(2X,I3)

2.2) DO 7 J=6,18,3

2 FORMAT(2X,I3)

7 WRITE(6,2)J

2.4) DO 6 I=1,3

K=10**I

6 WRITE(6,5)K

5 FORMAT(I5)

3. ค่าของ KOUNT ที่จะถูกพิมพ์เท่ากับเท่าใด (นั่นคือจำนวนครั้งที่คำสั่งที่ 3 ถูกทำ)

3.1) KOUNT=0

DO 20 I=1,3

DO 20 J=1,4

DO 20 K=1,10

3 KOUNT=KOUNT+1

20 CONTINUE

WRITE(6,2)KOUNT

2 FORMAT(I4)

3.2) KOUNT=0

DO 33 I=2,20,3

DO 33 J=3,17,5

3 KOUNT=KOUNT+1

33 CONTINUE

WRITE (6, 2) KOUNT

```

3.3)      KOUNT=0

          DO 10 IO=2,8,2

          DO 10 IM=10,2,1

          DO 10 II=1,4,2

          3  KOUNT=KOUNT+1

          10  CONTINUE

          WRITE(6,2)KOUNT

```

4. จงบอกสิ่งผิดที่จะพบตอนแปลโปรแกรม (Compile-time-error)

```

15 DO 10 I=1,5

      WRITE(6,1)I

      IF(I.GT.5)GO TO 15

10  CONTINUE

```

5. ส่วนของโปรแกรมต่อไปนี้จะคำนวณค่าเฉลี่ยของคะแนน 10 ค่าหรือไม่

```

5.1)      SUM=0

          DO 10 I=1,10

          READ(5,1)GRADE

          10 SUM=SUM+GRADE

          AVER=SUM/I

          WRITE(6,6)AVER

```

```

5.2)      SUM=0

          DO 15 I=1,10

          READ(5,1)GRADE

          SUM=SUM+GRADE

          15 AVER=SUM/I

          WRITE(6,6)AVER

```

```

5.3)    SUM=0

        Do 25 I=1,10

        READ(5,1)GRADE

25      SUM=SUM+GRADE

        AVER=SUM/10.

        WRITE(6,6)AVER

```

6. คำสั่ง READ และคำสั่ง FORMAT แต่ละคำสั่ง จะต้องใช้จำนวนลเข้าที่บัตร

- 6.1) READ(5,5)A,B,C
- a) 5 FORMAT(F5.1,F5.1,F5.1)
 - b) 5 FORMAT(F5.1,F5.1)
 - c) 5 FORMAT(4F5.1)
 - d) 5 FQRMAT(F5.1)

- 6.2) READ(5,6)I,J,K,L,M,N
- a) 6 FORMAT(I1, I2, I2)
 - b) 6 FORMAT(I1, I2, I3, I4)
 - c) 6 FORMAT(I1, I2)
 - d) 6 FORMAT(6I3)

- 6.3) READ(5,7)A, I, D
- a) 7 FORMAT(F5.1, I3, F4.1)
 - b) 7 FORMAT(F5.1/I2)
 - c) 7 FORMAT(F5.1, I2)
 - d) 7 FORMAT(F5.1/I2/F4.1)

7. จงบอกสิ่งที่มีอย่างน้อย 6 ที่ในโปรแกรมต่อไปนี้

```

INTEGER A,B,STAT,STAT10,STAT100,X

```

```

DATA X,Y,A,I/4*0./

```


DATA PI,E/3.14159,2.71828/

READ(5,10)K,L,N

10 FORMAT(3I5.0)

IF(K.EQ.L)THEN L=L+1

GO TO (10,20,30,30)L

20 STOP

30 WRITE(6,25)L

25 FORMAT(I6)

DO 100 I=1,10

DO 300 J=1,3

77 M=I*J

WRITE(6,200)M

100 CONTINUE

300 CONTINUE

IF(I*J*L)GO TO 77

STOP

END

8. จงเขียนโปรแกรมเพื่อหาค่าของ $3+5+7+\dots+225+227$ (ไม่ให้ใช้ข้อมูลเข้าจากคีย์กลาง)

9. จงเขียนโปรแกรมเพื่อหาค่าของ $1-2+3-4+5-\dots-100$ (ไม่ให้ใช้ข้อมูลเข้าจากคีย์กลาง)

10. จงเขียนโปรแกรมเพื่ออ่านค่าของ N แล้วคำนวณผลบวกกำลังสองของเลขคู่ N เทอมแรก

ตัวอย่างเช่น N=4 ต้องการหาผลบวกของ $2^2+4^2+6^2+8^2$

11. จงเขียนโปรแกรมเพื่อคำนวณค่าของ $\pi/4$ โดยใช้สูตร

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots$$

การประมาณครั้งแรกคือ 1

การประมาณครั้งที่ 2 คือ $1 - \frac{1}{3}$

การประมาณครั้งที่ 3 คือ $1 - \frac{1}{9} + \frac{1}{5}, \dots$

ให้หยุดการประมาณเมื่อการประมาณ 2 ครั้งติดต่อกันมีค่าต่างกันน้อยกว่า .01

12. จงเขียนโปรแกรมเพื่อพิมพ์ตารางสูตรคูณแม่ 2 ถึง 12 และกำหนดให้พิมพ์ผลดังนี้

MULTIPLICATION

TABLES

2* 2 = 4

2* 3 = 6

:

2*12 = 24

:

12* 2 = 24

:

12*12 = 144

13. โรงพยาบาลแห่งหนึ่งเก็บข้อมูลของคนที่ยื่นคำขออนุญาตให้เมื่อโรงพยาบาลต้องการ

สังเกตว่ามีคนพิมพ์รายชื่ออยู่ 2000 คนแต่ละคนมีข้อมูลอยู่จำนวน 1 ปีเศษ ซึ่งเก็บรูปแบบดังนี้

| ฟิลด์ | ลำดับ | รายการข้อมูล | ลักษณะของข้อมูล |
|-------|-------|--|-----------------|
| 1 | 1-6 | รหัสประจำตัว (CODE) | XXXXXX |
| 2 | 7-38 | ชื่อ-นามสกุล | |
| 3 | 39-45 | หมายเลขโทรศัพท์ | xxxxxxx |
| 4 | 46 | กลุ่ม ion : 1=เลือกกลุ่ม A 2=เลือกกลุ่ม B 3=เลือกกลุ่ม AB 4=เลือกกลุ่ม โอ | X |
| 5 | 47-48 | อายุ | xx |
| 6 | 49-54 | น้ำหนัก (nn.) | xxx. xx |

| รหัส | สภพ | รายการข้อมูล | ลักษณะของข้อมูล |
|------|-------|--------------|-----------------|
| 7 | 55-57 | ความสูง (ม.) | xxx |

จงเขียนโปรแกรมเพื่อเลือกและพิมพ์ข้อมูลของคนที่มีเลือดกลุ่มโอ นับจำนวนคนที่มีเลือด
 กลุ่มโอทั้งหมดและหาหนักเฉลี่ยของคนกลุ่มนี้
 กำหนดรูปแบบของกาพิมพ์ดังนี้

HOSPITAL XYZ

BLOOD TYPE O

| คนที่ ↓ NUMBER | CODE | NAME | TEL.NO. | AGE | WEIGHT (LG.) | HEIGHT (C.M.) |
|----------------------|------|------|---------|-----|-----------------|------------------|
| 1 | | | : | : | : | : |
| 2 | : | : | : | : | : | : |

TOTAL = XXXX PERSONS

AVERAGE WEIGHT = xxx.xx KILOGRAMS