

## บทที่ 3

### ขบวนการนับ(The Counting Process)

#### 3.1 เนื้อหาตัวอย่าง

#### 3.2 คำสั่งในภาษาฟอร์แทรน

##### 3.2.1 คำสั่งเฉพาะ (Specification statement)

###### 3.2.1.1 การกำหนดชนิดของตัวแปรอย่างชัดเจน

###### 3.2.1.2 คำสั่ง DATA

###### 3.2.1.3 คำสั่ง IMPLICIT

##### 3.2.2 การใช้เครื่องหมายทับ (/ slash)

##### 3.2.3 ตัวอักขระควบคุมการพิมพ์ (Carriage control character)

#### 3.3 ตัวอย่างเพิ่มเติม

แบบฝึกหัดที่ 3

# บทที่ 3

## ขบวนการนับ(The Counting Process)

### 3.1 ปกติตัวอย่าง

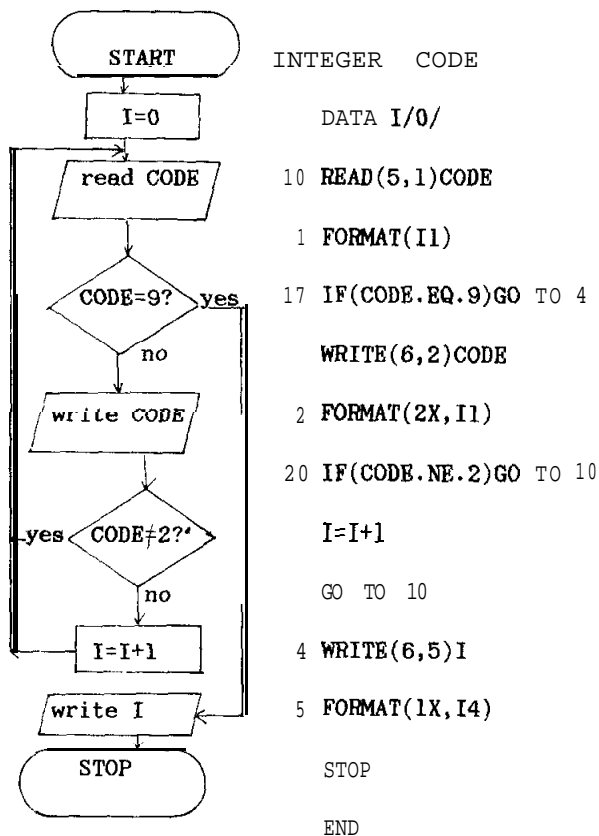
ในการสอบถามความเห็นเกี่ยวกับการจัดตั้งชมรม ๗ แห่งในมหาวิทยาลัย ผู้สอบถามได้ให้รหัสเกี่ยวกับความเห็นดังนี้

รหัส	ความหมาย
0	ต่อต้าน
1	เฉย ๆ
2	สนับสนุน

ความเห็นของนักศึกษาคนหนึ่งบันทึกเป็น 1 ระเบียบ (record) ให้ระเบียบที่มีรหัสพิเศษคือ 9 เป็นบัตรสุดท้าย (last card) ซึ่งไม่ใช่บัตรข้อมูล จึงเขียนโปรแกรมเพื่อนับจำนวนนักศึกษาที่สนับสนุนการจัดตั้งชมรมนี้

I จะใช้นับจำนวนนักศึกษาที่สนับสนุน  
อ่านรหัส

บัตรข้อมูลหมดหรือยัง?  
ถ้าบัตรหมดให้ไปพิมพ์จำนวนที่นับได้  
 แต่ถ้าบัตรไม่หมด ให้พิมพ์รหัสที่อ่าน  
เข้าไปและตรวจสอบว่าเป็นผู้สนับสนุน  
หรือไม่ ถ้าเป็นก็นับ แต่ถ้าไม่สนับสนุน  
ก็กลับไปอ่านข้อมูลใหม่



คำสั่งใหม่ในตัวอย่างนี้คือคำสั่ง INTEGER และคำสั่ง DATA คำสั่งทั้ง 2 เป็นคำสั่งที่เราเรียกว่าคำสั่งเฉพาะ (Specification statement) คำสั่งเฉพาะในภาษาฟอร์แทรนมีอีกหลายคำสั่งซึ่งจะได้อธิบายต่อไป

### 3.2 คำสั่งในภาษาฟอร์แทรน

#### 3.2.1 คำสั่งเฉพาะ (Specification statement)

คำสั่งเฉพาะเป็นคำสั่งที่ไม่ปฏิบัติการ (nonexecutable statement) คำสั่งเหล่านี้ต้องใส่ไว้ก่อนคำสั่งปฏิบัติการแรกของโปรแกรม คำสั่งเฉพาะเป็นคำสั่งซึ่งบอกคอมพิวเตอร์ให้จัดเตรียมสิ่งต่าง ๆ ให้พร้อมไว้เพื่อการทำงานตามคำสั่งปฏิบัติการจะได้เป็นไปอย่างเรียบร้อย เช่นคำสั่งในการกำหนดชนิดให้แก่ตัวแปรโดยใช้คำสั่ง TYPE หรือใช้คำสั่ง IMPLICIT คำสั่ง DATA เป็นต้น

##### 3.2.1.1 การกำหนดชนิดของตัวแปรโดยชัดแจ้ง (explicit)

การกำหนดชนิดของตัวแปรโดยใช้ตัวอักษรตัวแรกของชื่อเป็นตัวกำหนดเราเรียกว่าการกำหนดโดยปริยาย (implicit) เราอาจกำหนดโดยชัดแจ้งได้ด้วยการใช้คำสั่ง เช่น คำสั่ง INTEGER คำสั่ง REAL เป็นต้น

รูปทั่วไป

INTEGER	รายชื่อตัวแปร
REAL	รายชื่อตัวแปร

โดยที่ คำ INTEGER และ REAL เป็นคีย์เวิร์ด

รายชื่อตัวแปร นั้นอาจมีตัวแปรหนึ่งตัวหรือมากกว่าหนึ่งตัว ในกรณีมากกว่า 1 ตัว ให้ใช้เครื่องหมายจุลภาค (,) คั่น

ตัวแปรทุกตัวในรายชื่อตัวแปรที่ตามหลังคีย์เวิร์ด INTEGER จะมีชนิดเป็น integer ไม่ว่าจะปรากฏที่ใดในโปรแกรม แม้ว่าชื่อเหล่านั้นจะไม่เริ่มต้นด้วยตัวอักษร I-N ตัวแปรที่ตามหลังคีย์เวิร์ด REAL จะมีชนิดเป็น real

ตัวอย่าง                    INTEGER X, CODE1, QI, J

REAL I, J123, SUM

### 3.2.1.2 คำสั่ง DATA

คำสั่ง DATA ใช้ในการกำหนดค่าเริ่มต้น (initial value) ให้แก่ตัวแปร มีรูปทั่วไปคือ

DATA รายชื่อตัวแปร/รายการค่าคงที่/[,...]

โดยที่ รายชื่อตัวแปร คือรายชื่อตัวแปรซึ่งค้นด้วยเครื่องหมายจุลภาค (,)

รายการค่าคงที่ คือค่าคงที่ตั้งแต่ 1 ตัวขึ้นไปค้นด้วยเครื่องหมายจุลภาค

ตัวแปรแรกในรายชื่อจะคู่กับค่าคงที่ค่าแรกในรายการค่าคงที่

ตัวแปรตัวที่สองในรายชื่อจะคู่กับค่าคงที่ค่าที่สองในรายการค่าคงที่

ดังนั้นจำนวนตัวของตัวแปรและจำนวนตัวของค่าคงที่ตรงเท่ากัน และชนิดของตัวที่อยู่กันต้องเป็นชนิดเดียวกันด้วย

ตัวอย่าง DATA A,B/3.2,0./

ค่าของ A คือ 3.2 และค่าของ B คือ 0.

ตัวอย่าง DATA A,J,K/3.2,4/ ไม่ถูกต้องเพราะมีตัวแปร 3 ตัว แต่มีค่าคงที่ 2 ค่าเท่านั้น

ตัวอย่าง DATA CODE,LL/0,3.2/ ไม่ถูกต้องเพราะชนิดชนิดของตัวแปรและชนิดของค่าคงที่ไม่ถูกต้องในคู่ของมัน

ถ้าต้องการให้ CODE=3.2 และ LL=0 ต้องเขียน

DATA CODE,LL/3.2,0/

ตัวอย่าง DATA A,B/3.,0./,I,J/1,0/

↑ ต้องมีเครื่องหมาย , ถ้ามี รายชื่อตัวแปร/รายการค่าคงที่/ มากกว่า 1 ชุด

ดังนั้นค่าของ A คือ 3.

ค่าของ B คือ 0.

ค่าของ I คือ 1

ค่าของ J คือ 0

แสดงการซ้ำไม่ได้หมายถึงการคูณ

ตัวอย่าง DATA I,J,K/3\*0/,A,B,C,D/4\*(-4.1)/

หมายความว่า I,J,K แต่ละตัวค่ามีค่า=0

A,B,C,D แต่ละตัวค่ามีค่า=-4.1

คำสั่งข้างต้นจะ เหมือนกับคำสั่ง

DATA I,J,K/0,0,0/,A,B,C,D/-4.1,-4.1,-4.1,-4.1/

เนื่องจากคำสั่ง DATA เป็นคำสั่งไม่ปฏิบัติการ เราไม่สามารถใช้คำสั่ง DATA เพื่อกำหนดค่าใหม่ให้แก่ตัวแปรระหว่างการวิ่งโปรแกรม ในการกำหนดค่าใหม่ให้แก่ตัวแปรเราจะใช้คำสั่งในการกำหนดค่าหรืออ่านค่าใหม่จากตัวกลาง

### 3.2.1.3 คำสั่ง IMPLICIT

ในโปรแกรมที่ใช้ตัวแปรทั้งชนิด integer และชนิด real จำนวนมากจะดูไม่สะดวกในการกำหนดชนิดให้ตัวแปรทั้งหมดโดยชัดเจน การใช้คำสั่ง IMPLICIT จะทำให้โปรแกรมเมอร์สามารถกำหนดได้เองว่า ถ้าตัวแปรนั้นขึ้นต้นด้วยตัวอักษรใดจะมีชนิดใด เช่น

IMPLICIT REAL (I)

ซึ่งจะทำให้ตัวแปรทุกตัวที่ขึ้นต้นด้วยตัวอักษร I มีชนิดเป็น real เราสามารถใช้เลขของตัวอักษรในการกำหนดด้วยเช่น

IMPLICIT INTEGER (A-D)

ซึ่งจะทำให้ตัวแปรใด ๆ ที่ขึ้นต้นด้วย A, B, C หรือ D มีชนิดเป็น integer รูปทั่วไปของคำสั่งคือ

IMPLICIT ชนิด(a[,a]...), ชนิด(a[,a]...),...

โดยที่ชนิด (type) อาจเป็น INTEGER REAL หรือชนิดอื่น ๆ ซึ่งจะกล่าวถึงในตอนต่อไป

a อาจเป็นตัวอักษรตัวเดียวหรือเลขของตัวอักษรในรูป  $l_1-l_2$  ( $l_1$  และ  $l_2$

เป็นตัวอักษรเดี่ยว ๆ)

ตัวอย่าง IMPLICIT INTEGER(A,C-F),REAL(I-N)

IMPLICIT REAL (C),INTEGER(A,B)





บรรทัดที่ 1	เว้น	} เว้น 2 บรรทัด
บรรทัดที่ 2	เว้น	
บรรทัดที่ 3	^^XX XXX	
บรรทัดที่ 4	เว้น	} เว้น 2 บรรทัด
บรรทัดที่ 5	เว้น	
บรรทัดที่ 6	^XX.XX	

3.2.3 ตัวอักษรควบคุมการพิมพ์ (Carriage control character)

ตัวอย่าง

I=5

J=6

K=7

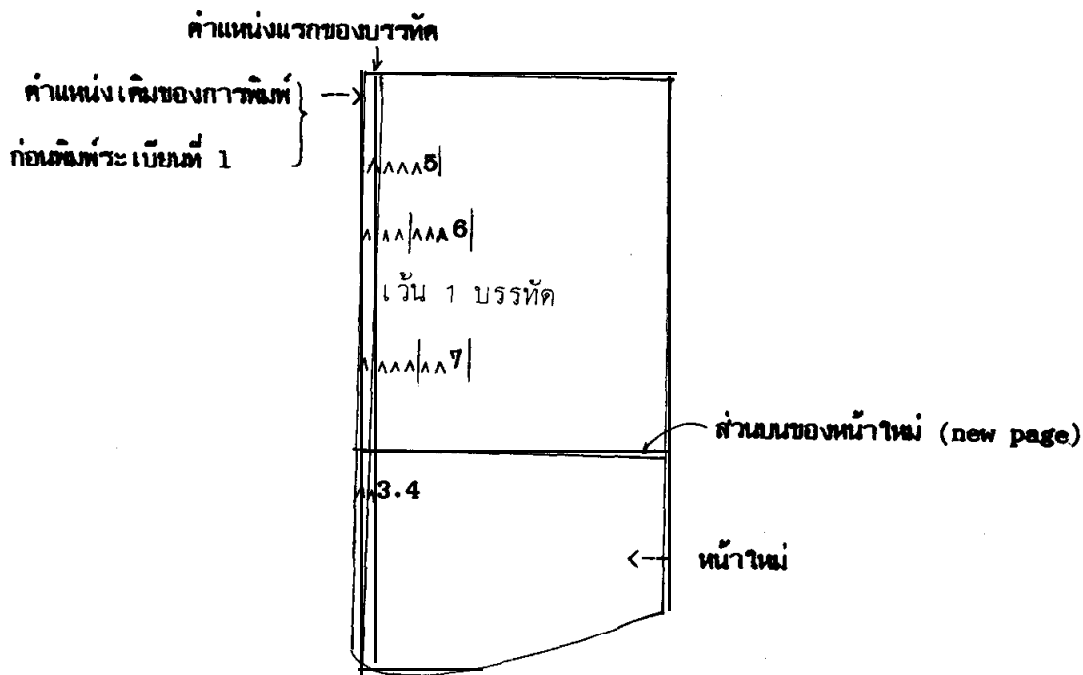
A=3.43

WRITE(6,1)I,J,K,A

1 FORMAT(1X,I4/' ',2X,I4/'0',3X,I3/'1',1X,F3.1)

ระเบียนที่ 1    ระเบียนที่ 2    ระเบียนที่ 3    ระเบียนที่ 4

ผลกาพิมพ์คือ





ตำแหน่งแรกของทศบรรทัดของกระดาษพิมพ์จะ ไม่ใช้ในการแสดงผล แต่ใช้ใส่ตัวอักษรควบคุมการพิมพ์ โดยจะใส่ไว้ในคำสั่ง FORMAT ดังตัวอย่างข้างต้นในรูป 'X' หรือ 1HX ก็ได้ โดยที่ X คือตัวอักษรควบคุมการพิมพ์ ซึ่งอาจเป็นตัวใดตัวหนึ่งดังต่อไปนี้

X : ตัวอักษรควบคุมการพิมพ์	การควบคุมการพิมพ์
ช่องว่าง ( blank )	ขึ้นบรรทัดใหม่(new line)
0	เว้น 1 บรรทัด (skip a line)
1	ขึ้นหน้าใหม่ (new page)
+	พิมพ์บนบรรทัดเดิม (โดยจะกลับมาเริ่มจากตำแหน่งแรกของบรรทัดก่อนพิมพ์)

คำสั่งต่อไปนี้จะ เหมือนกับคำสั่งในตัวอย่างข้างต้น (ใช้ H-format)

```
WRITE(6,1)I,J,K,A
```

```
1 FORMAT(1X,I4/1H_,2X,I4/1H0,3X,I3/1H1,1X,F3.1)
```

### 3.3 ตัวอย่างเพิ่มเติม

จากแฟ้มข้อมูลหนึ่งซึ่งประกอบด้วย 120 ระเบียน ระเบียนหนึ่ง ๆ มี 3 รายการข้อมูล คือเลขประจำตัว อายุ และเพศ ของสมาชิก 1 คนของชมรม ๆ หนึ่งในเมือง จงเขียนโปรแกรมเพื่อคำนวณหาเปอร์เซ็นต์ของชายที่มีอายุตั้งแต่ 60 ปีขึ้นไปในชมรมนี้

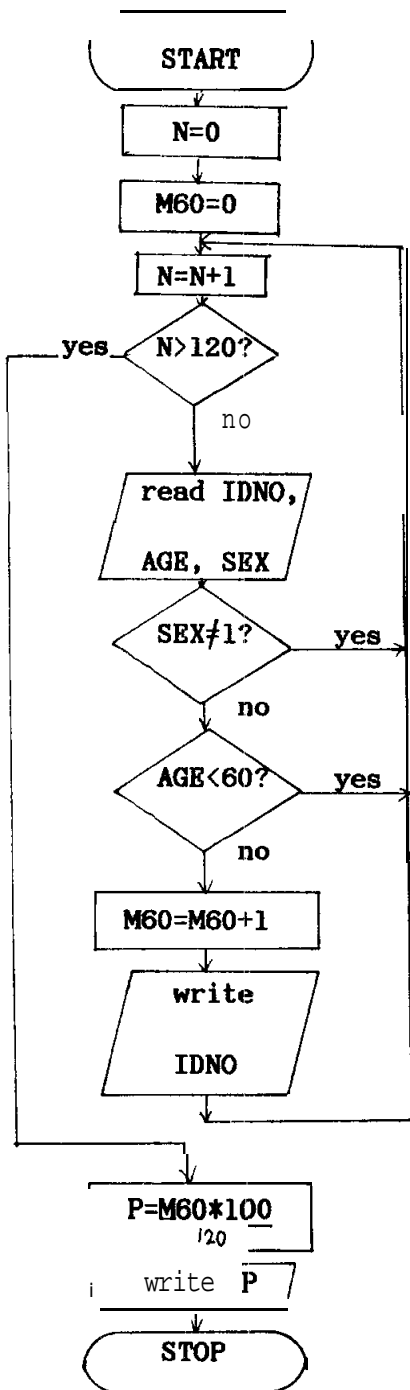
ตัวแปรซึ่งใช้เก็บรายการข้อมูลคือ

IDNO เก็บเลขประจำตัว

AGE เก็บอายุ

SEX เก็บเพศ โดยที่ SEX=1 คือชาย

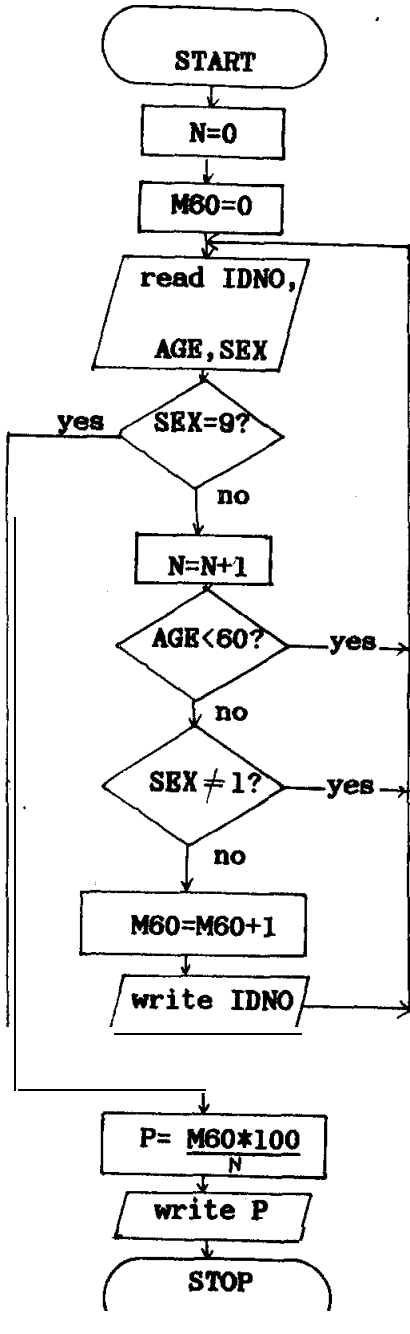
SEX=2 คือหญิง



```

INTEGER AGE,SEX
DATA N,M60/2*0/
11 N=N+1
IF(N.GT.120)GO TO 6
READ(5,1) IDNO, AGE, SEX
1 FORMAT(I8,I3,I2)
IF(SEX.NE.1)GO TO 11
IF(AGE.LT.60)GO TO 11
M60=M60+1
WRITE(6,4) IDNO
4 FORMAT(5X,I8)
Go To 11
6 P=(M60*100.)/120.
WRITE(6,7)P
7 FORMAT(10X,F6.2)
STOP
END
  
```

จากโปรแกรม ถ้าเราไม่ทราบจำนวนระเบียบในแฟ้มข้อมูล เราจะควบคุมการอ่านบัตร โดยกำหนดบัตรใบสุดท้ายขึ้น (ทำนองเดียวกับข้อ 3.1) ในที่นี้ให้บัตรใบสุดท้ายมีค่าของ SEX=9 ส่วนในฟิลด์อื่น ๆ เป็นช่องว่าง ดังนั้นแผนภูมิสายงานและโปรแกรมจะเป็นดังนี้



```

INTEGER AGE, SEX

DATA N/0/, M60/0/

3 READ(5, 1) IDNO, AGE, SEX
1 FORMAT(18, I3, I2)
IF(SEX.EQ.9)GO TO 6

N=N+1
IF(AGE.LT.60)GO TO 3
IF(SEX.NE.1)GO TO 3
M60=M60+1
WRITE(6, 4) IDNO
4 FORMAT(5X, I8)
GOTO 3
6 P=(M60*100.)/N
WRITE(6, 7)P
7 FORMAT(10X, F6.2)

STOP

END
  
```

### แบบฝึกหัดที่ 3

1. จงแสดงบัตรข้อมูลเข้าสำหรับคำสั่ง READ เมื่อคำสั่ง FORMAT เปลี่ยนไป

READ(5,5)I,J

- 1.1) 5 FORMAT(I4/I3)
- 1.2) 5 FORMAT(/I4, I3)
- 1.3) 5 FORMAT(/I4/I3)
- 1.4) 5 FORMAT(//I4, I3)
- 1.5) 5 FORMAT(//I4//I3)

2. จงแสดงรูปของผลการพิมพ์ตามคำสั่ง WRITE เมื่อคำสั่ง FORMAT เปลี่ยนไป

WRITE(6,6)I,K

- 2.1) 6 FORMAT(/1X, I2//1X, I3)
- 2.2) 6 FORMAT('0', I2/1X, I3)
- 2.3) 6 FORMAT(//T2, I2/1X, I3)
- 2.4) 6 FORMAT(//1X, I2//1X, I4)
- 2.5) 6 FORMAT(/'0', I2/'0', I3)

3. จากคำสั่งต่อไปนี้ จะต้องอ่านบัตรกับด

- 3.1) READ(5,6)X, I, Y  
6 FORMAT(F5.1/I3/F5.1)
- 3.2) READ(5,7)I, X, K  
7 FORMAT(I1, F1.1, I2, F2.2, F3.3)
- 3.3) READ(5,8)I, J, K  
8 FORMAT(I2)

4. สมมติว่ามีสามเหลี่ยมมุมฉาก 5 รูป แต่ละรูปมีมุม C เป็นมุมฉาก เวทราบค่าด้าน a และ c ดังนี้

รูปที่	1	2	3	4	5
a	3.26	3.00	4.15	8.91	2.28
c	9.47	5.00	12.52	10.25	3.49

(ให้ใช้ input card 5 บัตร แต่ละบัตรมีค่า a และ c ของแต่ละรูป)

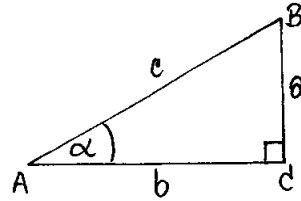
$$a^2 + b^2 = c^2 \quad \text{ดังนั้น} \quad b = \sqrt{c^2 - a^2}$$

จงเขียนโปรแกรมเพื่อคำนวณหาค่าต่อไปนี้

$$\sin \alpha = a/c$$

$$\cos \alpha = b/c$$

$$\tan \alpha = a/b$$



แต่ละรูปให้พิมพ์ตามรูปแบบดังนี้

TRIANGULAR NO. \_\_\_\_

A = \_\_\_\_\_ B = \_\_\_\_\_ C = \_\_\_\_\_

SIN(ALPHA) = \_\_\_\_\_

COS(ALPHA) = \_\_\_\_\_

TAN(ALPHA) = \_\_\_\_\_