

บทที่ 6

แถวลำดับแบบ 1 มิติ (One-dimensional array)

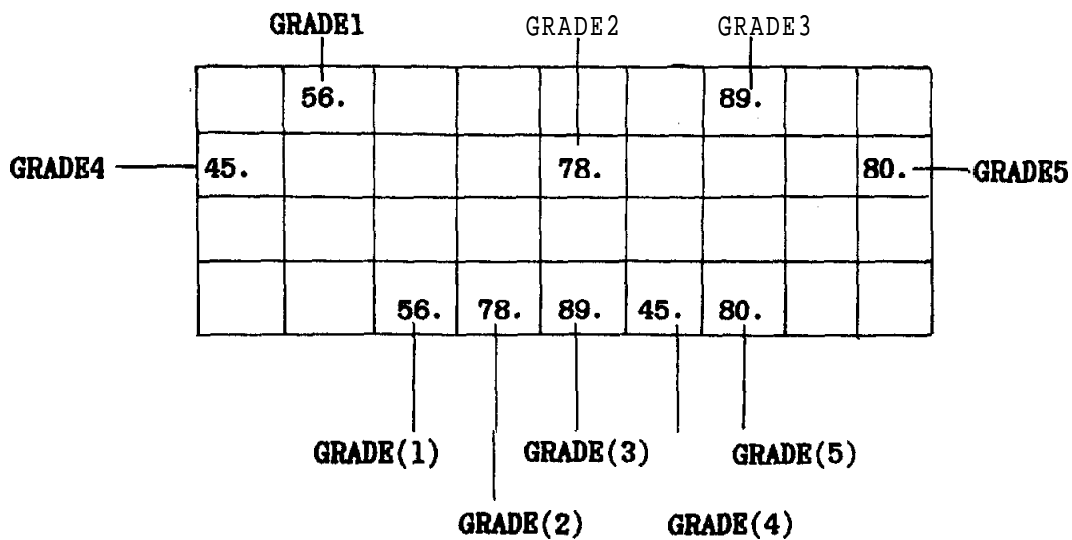
- 6.1 โปรแกรมตัวอย่าง
- 6.2 คำสั่งในภาษาฟอร์แทรน
 - 6.2.1 คำสั่ง DIMENSION
 - 6.2.2 การกำหนดค่าเริ่มต้นและคัดลอกค่าที่มีอยู่แล้วให้แก่แถวลำดับ
 - 6.2.3 การหาผลบวกสะสมโดยใช้สมาชิกของแถวลำดับ
 - 6.2.4 การรวมแถวลำดับ
 - 6.2.5 การค้นหาสิ่งที่ต้องการในแถวลำดับ
 - 6.2.6 การนำข้อมูลเข้า/ออกจากแถวลำดับโดยใช้คำสั่ง DO และ Implied DO
 - 6.2.7 คำสั่ง DATA กับแถวลำดับ
- 6.3 โปรแกรมตัวอย่างเพิ่มเติม
 - 6.3.1 การหาสมาชิกตัวที่มีค่าสูงสุดของแถวลำดับ
 - 6.3.2 การใช้ค่าจากตาราง
 - 6.3.3 ตารางแจกแจงความถี่
 - 6.3.4 กราฟแท่ง
 - 6.3.5 การเขียนกราฟ
- 6.4 คำสั่งในภาษาฟอร์แทรนเพิ่มเติม
 - 6.4.1 การกำหนด NAMELIST
 - 6.4.2 การใช้ format code ข้ำ

แบบฝึกหัดที่ 6

ผลการวิ่งไปรแกม	56.0	-13.6
	78.0	8.4
	89.0	19.4
	45.0	-24.6
	80.0	10.4

6.1.2 วิธีที่ 2

อีกวิธีหนึ่งที่เราทำได้และควรทำสำหรับปัญหานี้คือการกำหนดแถวลำดับชั้นเพื่อเก็บคะแนน 5 ค่า แถวลำดับคือที่ภายในหน่วยความจำที่อยู่ต่อเนื่องกันใช้เก็บชุดของข้อมูล เราระบุสมาชิกของแถวลำดับได้โดยใช้ชื่อของแถวลำดับและเลขประจำตำแหน่งที่อยู่ของมันในแถวลำดับซึ่งเรียกว่าคราซน์ล่าง (subscript) คราซน์นี้จะสัมพันธ์กับสมาชิกตัวแรกหรือตำแหน่งของสมาชิกตัวแรกของแถวลำดับตัวอย่างเช่น ตัวแรก ตัวที่ 4 ตัวที่ 5 เป็นต้น เราอาจตั้งแถวลำดับชื่อ GRADE ซึ่งมีสมาชิก 5 ตัวคือ GRADE(1), GRADE(2), GRADE(3), GRADE(4) และ GRADE(5) แถวลำดับ GRADE จะใช้ที่ 5 ที่ต่อเนื่องกันในหน่วยความจำหลัก รูปต่อไปนี้แสดงการเก็บข้อมูลทีอ่านจากบัตรเข้าไปเก็บไว้ในหน่วยความจำหลักตามวิธีในข้อ 6.1.1 และ 6.1.2



C DEVIATION FOR **FIVE** GRADES USING ARRAY

DIMENSION GRADE@)

(สำหรับที่ 5 สำหรับแถวลำดับ GRADE)

SUM=0

DO 10 I=1,5

READ(5,6)GRADE(I)

(เราเรียก I ว่ากรณีนี้ล่าง)

6 FORMAT(F5.1)

SUM=SUM+GRADE(I)

10 CONTINUE

AVE=SUM/5.

(เมื่อออกจากลูป $SUM = \sum_{i=1}^5 GRADE(i)$)

DO 30 I=1,5

DIF=GRADE(I)-AVE

WRITE(6,7)GRADE(I),DIF

เมื่อ I=1, DIF=GRADE(1)-AVE,
I=2, DIF=GRADE(2)-AVE,...

7 FORMAT(T10,F5.1,3X,F5.1)

30 CONTINUE

STOP

END

6.2 คำสั่งในภาษาฟอร์แทรน

6.2.1 คำสั่ง DIMENSION

รูปทั่วไป

DIMENSION ชื่อแถวลำดับ 1 (ลิมิต 1)[,ชื่อแถวลำดับ 2 (ลิมิต 2),...]

โดยที่

DIMENSION เป็นคีย์เวิร์ด

ชื่อแถวลำดับ ตั้งตามหลักการตั้งชื่อ ชนิดของแถวลำดับ (integer หรือ real) นั้น

ถูกกำหนดโดยชื่อของแถวลำดับเช่นเดียวกับตัวแปร นั่นคือถ้าชื่อขึ้นต้นด้วยตัวอักษร I-N สมาชิก

แต่ละตัวของแถวลำดับเป็นตัวแปรชนิด integer นอกนั้นเป็นตัวแปรชนิด real

ลิมิต 1, ลิมิต 2, ... เป็นเลขจำนวนเต็มที่ไม่มีเครื่องหมาย ใช้ออกค่าสูงสุดของจำนวนที่สำรองไว้ให้แถวลำดับแต่ละแถวลำดับ (ลิมิตจะมีค่าสูงสุดเท่าใดขึ้นอยู่กับขนาดของหน่วยความจำหลักของคอมพิวเตอร์ที่เราใช้) ที่ ๆ สำรองไว้แน่นอนอาจไม่ถูกใช้ทั้งหมด ครรชนของแถวลำดับสามารถแปรค่าได้จาก 1 ถึงลิมิตที่กำหนดไว้ในคำสั่ง DIMENSION และจะมีค่าเกินลิมิตนั้นไม่ได้ แถวลำดับที่จะใช้ในโปรแกรมหนึ่ง ๆ จะต้องถูกกำหนดไว้ในคำสั่ง DIMENSION (หรือคำสั่งอื่นที่ใช้กำหนดแถวลำดับได้ด้วย) ก่อน ในคำสั่ง DIMENSION หนึ่ง ๆ เรากำหนดแถวลำดับก็แถวลำดับก็ได้ เช่น

DIMENSION X(6),Z(20), JS(107)

ซึ่งเป็นการกำหนดว่า X และ Z เป็นแถวลำดับชนิด real ส่วน JS เป็นแถวลำดับชนิด integer X มีสมาชิกได้ถึง 6 ตัว Z มีสมาชิกได้ถึง 20 ตัว และ JS เก็บเลขจำนวนเต็มได้ถึง 107 จำนวน

แถวลำดับ X

X(1)	X(2)	X(3)	X(4)	X(5)	X(6)
------	------	------	------	------	------

แถวลำดับ z

Z(1)		...		Z(6)	Z(7)	Z(20)
------	--	-----	--	------	------	-------	-------

แถวลำดับ JS

JS(1)	JS(2)	...		JS(6)	JS(7)	JS(20)	..	JS(107)
-------	-------	-----	--	-------	-------	-------	--------	----	---------

ตัวอย่าง คำสั่งที่ไม่ถูกต้อง

DIMENSION A(3.)

3. ใช้เป็นลิมิตไม่ได้ต้องใช้เลขจำนวนเต็ม

DIMENSION A(N)

N ไม่ใช่เลขจำนวนเต็ม (ในโปรแกรมย่อยคำสั่งนี้ใช้ได้)

คำสั่ง DIMENSION เป็นคำสั่งเฉพาะ ซึ่งต้องใส่ไว้ก่อนคำสั่งปฏิบัติการในโปรแกรม

ในโปรแกรมหนึ่ง ๆ จะมีคำสั่ง DIMENSION ก็คำสั่งก็ได้ เช่น

```

{
  DIMENSION A(10),B(3)
  DIMENSION S(4)
  DIMENSION JP(10)
}

```

เหมือนกับ DIMENSION A(10),B(3),S(4),JP(10)

การกำหนดแถวลำดับอาจกำหนดในคำสั่งกำหนดชนิดได้ด้วย เช่นคำสั่ง INTEGER,

REAL, DOUBLE PRECISION

ตัวอย่าง การกำหนดแถวลำดับโดยใช้คำสั่งกำหนดชนิด

```

REAL A(336),K(32)

INTEGER GRADE(30),JSUM(30)

DOUBLE PRECISION B(5),LT(6),M

```

ตัวอย่าง { DIMENSION A(100),B(60),I(45),C(50)
 INTEGER A,B
 REAL I

ซึ่งเหมือนกับ DIMENSION d(50)
 INTEGER A(100),B(60)
 REAL I(45)

ตัวอย่าง การกำหนดแถวลำดับไม่ถูกต้อง

```

| INTEGER A(100)
{ DIMENSION A(100)

```

ความเป็น

```

{ INTEGER A
  DIMENSION A(100)

```

หรือ INTEGER A(100)

ควรรชนีล่าง (subscript)

เราใช้ควรรชนีล่างเพื่อระบุสมาชิกที่ต้องการจากแถวลำดับ ควรรชนีจะอยู่ในวงเล็บตามหลังชื่อของแถวลำดับ ซึ่งอาจเป็นเลขจำนวนเต็ม ตัวแปรชนิด integer หรือนิพจน์ ทั้งนี้คอมพิวเตอร์บางระบบอาจจะกำหนดรูปแบบตายตัวของนิพจน์ที่จะใช้เป็นควรรชนีล่างไว้ด้วย

ตัวอย่าง ถ้า v เป็นตัวแปร c และ k เป็นเลขจำนวนเต็มที่ไม่มีเครื่องหมาย

รูปของควรรชนี	ตัวอย่าง	ความหมาย
c	$A(3)$	สมาชิกตัวที่ 3 ของ A
v	$B(J)$	สมาชิกตัวที่ j ของ B
$k*v$	$COST(3*K)$	ถ้า k มีค่าเป็น $1, 2, \dots$ จะอ้างถึงสมาชิกตัวที่ $3, 6, \dots$ ของ $COST$
$k*v \pm c$	$L(10*ISUM-7)$	คำนวณหาค่าของนิพจน์ $10*ISUM-7$ สมมติเท่ากับ 8 จะอ้างถึงสมาชิกตัวที่ 8 ของ L

ในภาษาฟอร์แทรน 77 ควรรชนีล่างจะเป็นนิพจน์แบบ integer แบบ real หรือสมาชิกของแถวลำดับก็ได้

สมมติค่าของแถวลำดับ A ค่าของตัวแปร I และ R เป็นดังนี้

1.3	4.	8.9	3.6	2.9	4.3	1	5.
$A(1)$	$A(2)$	$A(3)$	$A(4)$	$A(5)$	$A(6)$	I	R

ตัวอย่างต่อไปนี้แสดงการใช้และความหมายของควรรชนีล่าง

รูปแบบของดรรรชนีล่าง	ตัวอย่าง	ความหมาย
ค่าคงที่	$A(4)$	A ตัวที่ 4 คือ 3.6
ค่าคงที่	$A(3.7)$	เมื่อควรรชนีล่างเป็นเลขจำนวนจริง มันจะตัดทศนิยมทิ้งก่อน นั่นคือตัดทศนิยมของ 3.7 ทิ้งเหลือ 3 ดังนั้น $A(3)$ คือ 8.9

รูปของครรชนี่ล่าง	ตัวอย่าง	ความหมาย
ตัวแปร	$A(I)$	ถ้า $I=1$, $A(1)=1.3$
ตัวแปร	$A(R)$	$R=5.$, $A(R)=A(5)=2.9$
นิพจน์	$A(24/R-2.3)$	คำนวณค่า $24/R-2.3$ ได้ 2.5 ดังนั้น ครรชนี่ล่างเท่ากับ $2 A(2)=4.$
ฟังก์ชันภายใน	$A(\text{SQRT}(4.))$	$\text{SQRT}(4.)=2$ ดังนั้น $A(2)=4.$
สมาชิกของแถวลำดับ	$A(A(4))$	$A(4)=3.6$ ครรชนี่ล่าง=3 ดังนั้น $A(3)=8.9$

สิ่งที่สำคัญคือการเข้าใจความแตกต่างของค่าของครรชนี่ล่างและค่าของสมาชิกตัวนั้น ๆ
ค่าของสมาชิก $A(3)$ อาจจะไม่มีอะไรเกี่ยวข้องกับเลข 3 เลย เช่น $A(3)=.0076$ เราอาจใช้
ครรชนี่ล่างตัวเดียวกันกับสมาชิกของแถวลำดับที่ต่างกันได้ เช่น $A(I), B(I)$ เป็นต้น

ในภาษาฟอร์แทรน 4 ตัวอย่างต่อไปนี้เป็นครรชนี่ล่างไม่ได้

- $A(-3)$ ผิดเพราะครรชนี่ล่างเป็นลบ
- $E(2*A(1)+1)$ นิพจน์ประกอบด้วยสมาชิกของแถวลำดับ
- $G(0)$ ครรชนี่ล่างเป็นศูนย์ไม่ได้
- $Z(3.)$ ครรชนี่ล่างเป็นเลขจำนวนจริงไม่ได้

สมาชิกของแถวลำดับเมื่อใช้ในคำสั่งใด ๆ ก็ใช้เช่นเดียวกับตัวแปรทั่ว ๆ ไป ตัวอย่าง
เช่น $X(4)=2.1*(X(1)*X(3)-\text{SQRT}(Y)/X(4))$

ตัวอย่างต่อไปนี้แสดงข้อผิดพลาดเนื่องจากค่าของครรชนี่ล่างเกินค่าสูงสุดที่ระบุไว้ใน
คำสั่ง DIMENSION สมมุติว่ามีค่าของแถวลำดับ A และ B อยู่แล้ว

DIMENSION A(100),B(10)

I=101

Y=A(I)

I=101,A(101) ใช้ไม่ได้ เพราะ 101>100

B(11)=3.

B(11) ใช้ไม่ได้ เพราะ 11>10

DO 10 I=1,11

10 B(I)=0

เมื่อ I=11,B(11) ใช้ไม่ได้

6.2.2 การกำหนดค่าเริ่มต้นและคัดลอกค่าที่มียู่แล้วให้แก่แถวลำดับ

ส่วนของโปรแกรมต่อไปนี้กำหนดค่าเริ่มต้นให้สมาชิกทุกตัวของแถวลำดับ A เป็นศูนย์ และกำหนดค่าเริ่มต้นให้สมาชิกทุกตัวของแถวลำดับ B เท่ากับค่าของตัวแปร X และคัดลอกค่าของแถวลำดับ D เก็บไว้ในแถวลำดับ SAVE

:

READ(5,5)N

5 FORMAT(I3)

DO 10 I=1,N

A(I)=0

B(I)=X

10 SAVE(I)=D(I)

ผลจากคำสั่งข้างต้นคือ $A(1)=0, A(2)=0, \dots, A(N)=0$

$B(1)=X, B(2)=X, \dots, B(N)=X$

$SAVE(1)=D(1), \dots, SAVE(N)=D(N)$

บางครั้งเราต้องการใช้แถวลำดับ C เพื่อเก็บผลบวกของ 2 แถวลำดับคือ A และ B โดยที่ $C(1)=A(1)+B(1), C(2)=A(2)+B(2), \dots, C(100)=A(100)+B(100)$

```
DO 15 I=1,100
15 C(I)=A(I)+B(I)
```

ถ้าเราต้องการกำหนดค่าเริ่มต้นของแถวลำดับ A และ B ดังนี้

$$A(1)=B(10)=1$$

$$A(2)=B(9)=2$$

$$A(3)=B(8)=3$$

$$A(10)=B(1)=10$$

เราสามารถสร้างเลขจำนวนเต็ม 1,2,...,10 โดยการใช้นิพจน์ของคำสั่ง DO ดังนี้

```
DO 20 I=1,10
  A(I)=I
  K=10-I+1 } หรือให้ B(10-I+1) = I
  B(K)=I    }
20 CONTINUE
```

I	1	2	...	10
K	10	9	...	1

การกลับค่าในแถวลำดับ สมมติว่าแถวลำดับ A มีสมาชิก N ตัว เราต้องการจะเปลี่ยนค่าของ A(1) กับ A(N), A(2) กับ A(N-1), A(3) กับ A(N-2),... เราอาจเขียนส่วนของโปรแกรมดังนี้

$L=N/2$ (เราจะเคลื่อนย้ายสมาชิก $(N/2)$ คู่ ถ้า N เป็นเลขคี่ตัวกลางจะคงเดิม)

DO 10 I=1,L

TEMP=A(I)

K=N-I+1 (K=N,N-1,N-2,N-3,...)

A(I)=A(K)

A(K)=TEMP

10 CONTINUE

6.2.3 การหาผลบวกสะสมโดยใช้สมาชิกของแถวลำดับ

สมาชิกแถวลำดับ A คือ

10.	20.	30.	40.	50.
-----	-----	-----	-----	-----

ต้องการหา $A(1)+A(2)+A(3)+A(4)+A(5)=\sum_{i=1}^5 A(i)$

SUM=0

DO 10 I=1,5

SUM=SUM+A(I)

10 CONTINUE

การหาผลบวกเป็นไปดังนี้

I	SUM
	0
1	10.(0+10)
2	30.(10+20)
3	60.(30+30)
4	100.(60+40)
5	150.(100+50)

สมมติแถวลำดับ A และแถวลำดับ B ต่างก็มีสมาชิก N ตัว ต้องการหา

$$\sum_{i=1}^N A(i) + \sum_{i=1}^N B(i)$$

เราทำได้โดยการเขียนคำสั่งดังนี้

```
SUM=0
```

```
DO 10 I=1,N
```

```
    SUM=SUM+A(I)+B(I)
```

```
10 CONTINUE
```

6.2.4 การรวมแถวลำดับ

สมมติว่า A และ B ต่างก็เป็นแถวลำดับที่มีสมาชิก 10 ตัว เราต้องการสร้างแถวลำดับ C ให้มีสมาชิกตามลำดับดังนี้ A(1),B(1),A(2),B(2),...,A(10),B(10) เราทำได้โดยวิธีใดวิธีหนึ่งดังนี้

วิธี 1 DIMENSION A(10),B (10),C (20)

```
K=1
```

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

```
    C(K)=A(I)
```

```
    K=K+1
```

```
    C(K)=B(I)
```

```
    K=K+1
```

```
10 CONTINUE
```

วิธี 2

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

```
    C(2*I-1)=A(I)
```

```
    C(2*I)=B(I)
```

```
10 CONTINUE
```

๑๑๗
วิธีที่ 3

```
K=1  
DO 10 I=1,20,2  
    C(I)=A(K)  
    C(I+1)=B(K)  
K=K+1
```

10 CONTINUE

6.2.5 การค้นหาลำดับที่ต้องการในแถวลำดับ

สมมติแถวลำดับ A มีคะแนน 100 ค่าเก็บอยู่ เราต้องการจะนับจำนวนคนที่ได้คะแนนเกิน 53 คะแนน เราทำได้ดังนี้

```
K=0  
DO 10 I=1,100  
    IF(A(I).GT.53)K=K+1
```

10 CONTINUE

6.2.6 การนำข้อมูลเข้า/ออกจากแถวลำดับโดยใช้คำสั่ง DO และ Implied DO

วิธี 2 วิธีที่ใช้ในการอ่านข้อมูลเข้าไปเก็บในแถวลำดับ และเรียกสมาชิกของแถวลำดับออกมาพิมพ์คือ

- 1) ใช้คำสั่ง DO
- 2) ใช้ Implied DO

6.2.6.1 วิธีใช้คำสั่ง DO ในการนำข้อมูลเข้า/ออกจากแถวลำดับ

ตัวอย่าง DIMENSION A(10)

 DO 10 I=1,10

 READ(5,5)A(I)

 5 FORMAT(F5.0)

 10 CONTINUE:

ในหน่วยความจำหลัก

<u>ตัวอย่าง</u>	บัตรที่ 1	3.	A(1)	3.
	บัตรที่ 2	5.	A(2)	5.
	บัตรที่ 3	9.	A(3)	9.
			:	:
	บัตรที่ 10	15.	A(10)	15.

นั่นคือการอ่านบัตร 10 บัตรบัตรละ 1 ค่าเข้าไปเก็บในแถวลำดับ A

ตัวอย่าง DIMENSION IHR(10),RATE(10)

DO 12 I=1,10

READ(5,6) IHR(I),RATE(I)

6 FORMAT(I3,2X,F4.0)

12 CONTINUE

นั่นคือจะอ่านบัตร 10 บัตร ๆ ละ 2 ค่าเข้าไปเก็บในแถวลำดับ IHR และ RATE

ตัวอย่าง สมมติว่าแถวลำดับ SALES เก็บยอดขายประจำวันไว้ เราต้องการพิมพ์ยอดขายพร้อมกับวันที่ขายโดยย่อคนั้น ๆ ด้วย

DIMENSION SALES(7)

:

DO 10 I=1,7

WRITE(6,11)I,SALES(I)

11 FORMAT (1X, 12. 3X. F5. 1)

10 CONTINUE

ผลการทำงาน

1	101.1
2	200.0
3	150.2
4	300.0
5	400.8
6	502.4
7	500.0

ถ้าต้องการอ่านค่าไปเก็บในแถวลำดับ A โดยมีข้อมูลเกิน 1 จำนวนในบัตร 1 บัตร (เช่น 2 จำนวน) เราทำได้ดังนี้

```

DIMENSION A(10)

DO 10 I=1,5,2

    READ(5,8)A(I),A(I+1)

8 FORMAT(2F5.0)

10 CONTINUE
    
```

<u>บัตรข้อมูล</u>	บัตรที่	
	1	9.1 6.
	2	6. -2.
	3	3. 4.
	:	

แถวลำดับ A	
A(1)	9.1
A(2)	6.
A(3)	6.
A(4)	-2.
A(5)	3.
A(6)	4.
:	

ถ้ามีบัตรข้อมูล 5 บัตร ๆ ละ 6 ค่า เราต้องเขียนคำสั่ง READ ดังนี้

```
DIMENSION B(30)
```

```
DO 20 I=1,30,6
```

```
    READ(5,9)B(I),B(I+1),B(I+2),B(I+3),B(I+4),B(I+5)
```

```
  9  FORMAT(6F5.0)
```

```
20 CONTINUE
```

ซึ่งจะเห็นได้ว่าเราต้องเขียน B(I),...,B(I+5) ซึ่งเป็นการไม่สะดวก ดังนั้นเราจะใช้

Implied DO ให้เป็นประโยชน์

6.2.6.2 Implied DO

เราใช้ Implied DO ในคำสั่งนำข้อมูลเข้า/ออก และคำสั่ง DATA เท่านั้น

จากคำสั่ง READ(5,5)A(1),A(2),A(3),A(4),A(5),A(6) เราอาจเขียนแทน
ได้ด้วยสิ่งที่เรียกว่า Implied DO ดังนี้

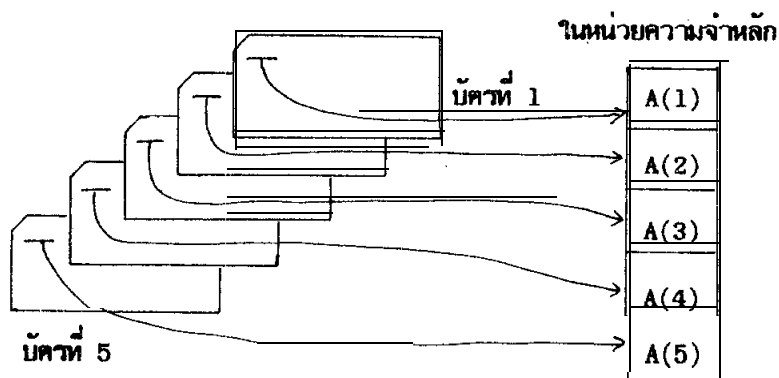
```
READ(5,5)(A(I),I=1,6,1)
```

```
หรือ READ(5,5)(A(I),I=1,6)
```

ในรายชื่อตัวแปรจะบอกเพียงว่าตัวแปรทั้งหมดในรายการมีตัวใดบ้างแต่ format code ในคำสั่ง
FORMAT จะเป็นตัวกำหนดว่าจะอ่านกี่ค่าจากบัตร 1 บัตรหรือจะพิมพ์กี่ค่าลงบน 1 บรรทัด

ตัวอย่าง READ(5,10)(A(I),I=1,5)

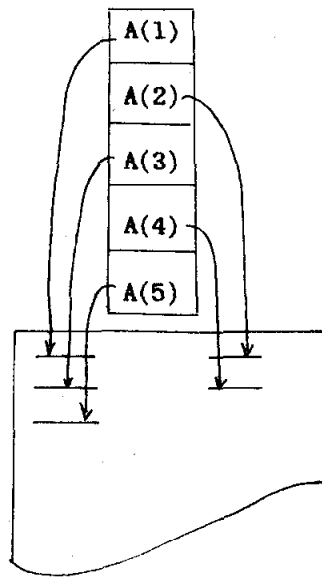
10 FORMAT(F5.0) → อ่านบัตรละ 1 ค่า



ตัวอย่าง WRITE(6,20)(A(I),I=1,5)

20 FORMAT(2X,2F6.0) ----> พิมพ์บรรทัดละ 2 ค่า

ในหน่วยความจำหลัก

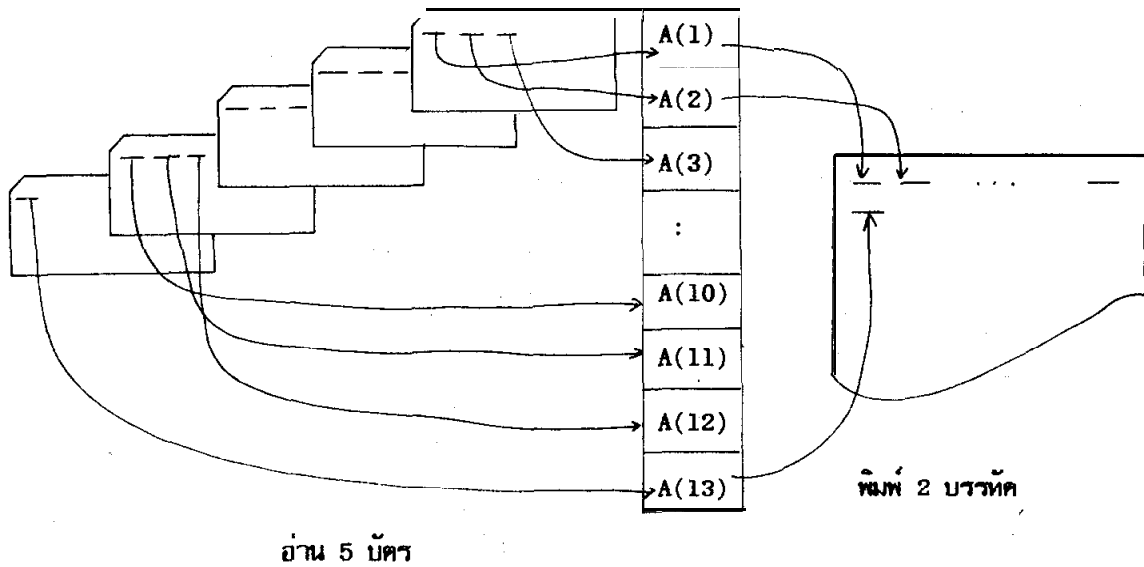


ตัวอย่าง READ(5,10)(A(I),I=1,13)

10 FORMAT(3F5.0) ----> อ่านบัตรละ 3 ค่า

WRITE(6,20)(A(I),I=1,13)

20 FOFMAT(12F6.0) ----> พิมพ์บรรทัดละ 12 ค่า



ตัวอย่าง READ(5,5)(SALES(I),I=1,7)

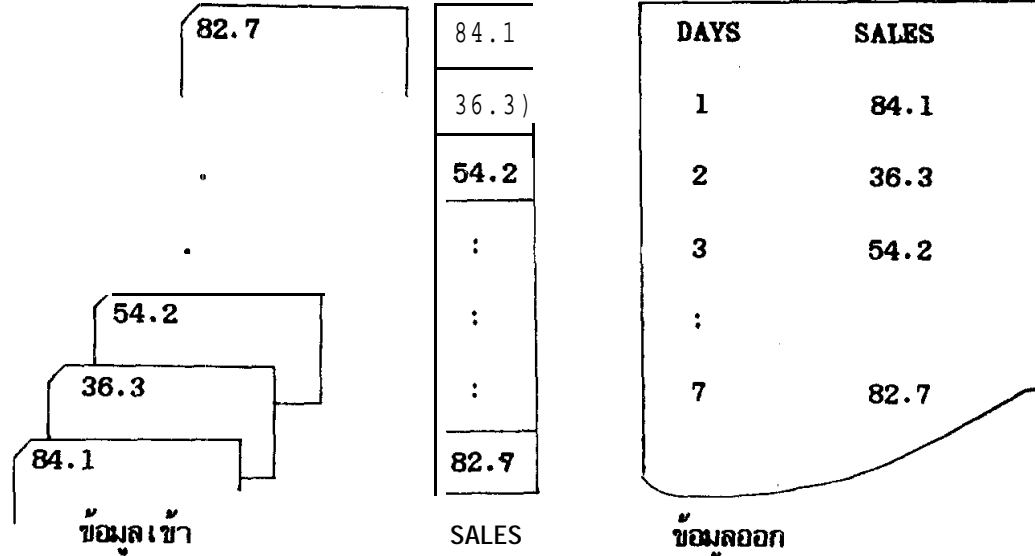
5 FORMAT(F4.0)

---> อ่านทีละ 1 ค่า

WRITE(6,6)(I,SALES(I),I=1,7)

6 FORMAT(T5,I2,T10,F5.1)

---> พิมพ์รายทีละ 2 AI



จากคำสั่ง WRITE

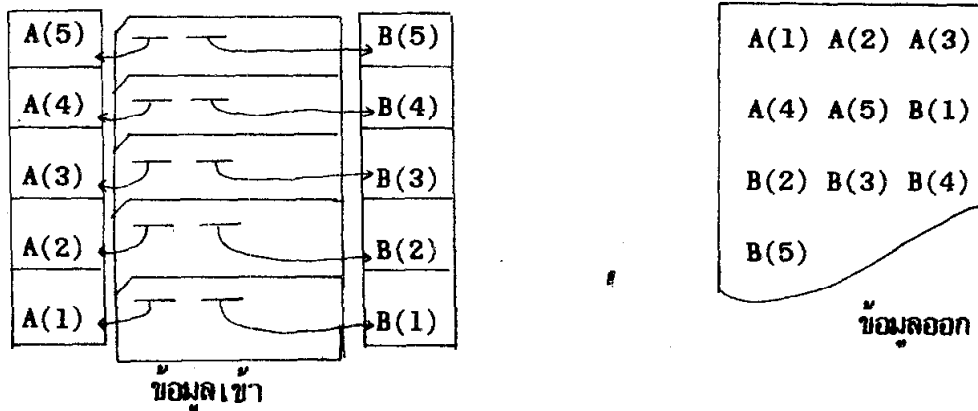
(I,SALES(I),I=1,7) คือ 1,SALES(1),2,SALES(2),...,7,SALES(7)

ตัวอย่าง READ(5,10)(A(I),B(I),I=1,5)

10 FOFtMAT(2F5.0)

WRITE(6,20)(A(I),I=1,5),(B(I),I=1,5)

20 FORMAT(3F6.0)



ถ้าใช้คำสั่ง WRITE(6,30)(A(I),B(I),I=1,5)

30 FORMAT(3F6.0)

จะได้ผลกาพิมพ์ดังนี้

A(1)	B(1)	A(2)
B(2)	A(3)	B(3)
A(4)	B(4)	A(5)
B(5)		

ตัวอย่าง ในการอ่านข้อมูลอักขระ เช่นข้อมูลอักขระบันทึกในบัตร 80 คอลัมน์ 1 บัตร เราอาจเขียนคำสั่ง READ ดังนี้ (สมมติให้ DOUBLE PRECISION มีขนาด 8 ไบต์)

DOUBLE PRECISION A(10)

READ(5,1)(A(I),I=1,10)

1 FORMAT(10A8)

บัตรข้อมูล

CS 215 F	ORTRAN P	ROGRAMMI	NG RAMKH	AMHAENG...
----------	----------	----------	----------	------------

ผลการอ่าน

A(1) = C S 2 1 5 F

A(2) = O R T R A N P

A(3) = R O G R A M M I

:

ตัวอย่าง ถ้าต้องการพิมพ์หัวตารางที่หน้าใหม่

PART NO.	MACHINE 1	MACHINE 2 . . .	MACHINE 9
----------	-----------	-----------------	-----------

คำว่า MACHINE ต้องพิมพ์ซ้ำ 9 ครั้ง และเลข 1,2,...,9 อาจสร้างขึ้นโดยใช้ implied DO

WRITE(6,1)(I,I=1,9)

1 FORMAT('1',T10,'PART NO.',4X,9('MACHINE',I2,2X))

สิ่งที่เข้าใจผิดกันมาก

```
DO 5 I=1,9
```

```
READ(5,6)A(I)
```

```
6 FORMAT(3F5.3)
```

```
5 CONTINUE
```

หลายท่านอาจคิดว่าคำสั่งข้างต้นจะใช้บัตรข้อมูล 3 บัตรเท่านั้น แต่ความจริงนั้นจะต้องใช้บัตรข้อมูลถึง 9 บัตร ถึงแม้ว่าบัตรแต่ละบัตรจะมี 3 ค่า แต่การอ่านจะอ่านเพียงบัตรละค่า เพราะเมื่อค่า I มีค่า 1 ค่าเรามี A(1) เพียงตัวเดียว

ถ้าต้องการอ่าน 3 ค่าจากบัตรแต่ละบัตร และมีบัตร 3 บัตร เราจะเขียนคำสั่งอ่านดังนี้

```
READ(5,6)(A(I),I=1,9)
```

```
6 FORMAT(3F5.3)
```

Nested DO list

เราอาจเขียน implied DO ซ้อนกันได้ทำนองเดียวกับในคำสั่ง DO

ตัวอย่าง จากคำสั่ง DO 6 I=1,3

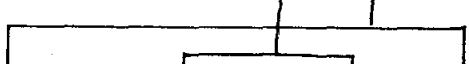
```
WRITE(6,5)A(I),B(I),(M(J),J=1,4)
```

```
5 FORMAT(1X,2F5.0,4I5)
```

```
6 CONTINUE
```

เราอาจใช้ Nested DO list ดังนี้

```
WRITE(6,5)(A(I),B(I),(M(J),J=1,4),I=1,3)
```



```
5 FORMAT(1X,2F5.0,4I5)
```

การทำงานเป็นดังนี้

$$\left\{ \begin{array}{l} I=1, J=1, 2, 3, 4 \\ I=2, J=1, 2, 3, 4 \\ I=3, J=1, 2, 3, 4 \end{array} \right.$$

ผลกาารพิมพ์

A(1)	B(1)	M(1)	M(2)	M(3)	M(4)
A(2)	B(2)	M(1)	M(2)	M(3)	M(4)
A(3)	B(3)	M(1)	M(2)	M(3)	M(4)

6.2.7 คำสั่ง DATA กับแถวลำดับ

ตัวอย่าง REAL A(5)

DATA A(1),A(2),A(3),A(4),A(5)/1.,2.1,-3.,4.,5.6/

หรือ DATA (A(I),I=1,5)/1.,2.1,-3.,4.,5.6/

หรือ DATA A/1.,2.1,-3.,4.,5.6/

หมายถึงสมาชิกทุกตัวของแถวลำดับ A ที่กำหนดไว้ในคำสั่งที่ใช้กำหนดแถวลำดับ ในที่นี้คือ A ทั้ง 5 ตัว

ตัวอย่าง ต้องการกำหนดให้สมาชิก 100 ตัวแรกของแถวลำดับ B ต่างก็มีค่าเป็นศูนย์

DIMENSION B(200)

DATA (B(I),I=1,100)/100*0./

ในกรณี B(101),...,B(200) ยังไม่ได้ถูกกำหนดค่าใดๆ ทั้งสิ้น (undefined)

ตัวอย่าง ต้องการกำหนดให้สมาชิกทั้ง 200 ตัวของแถวลำดับ B มีค่าเป็น 1

DIMENSION B(200)

DATA (B(I),I=1,200)/200*1./

หรือ DATA B/200*1./

ตัวอย่างอื่น ๆ ในการใช้แถวลำดับ

ตัวอย่าง DIMENSION A(10)

READ(5,10)(A(I),I=1,10) หรือ READ(5,10)A

10 FORMAT(F6.0)

ตัวอย่าง การใช้ชื่อแถวลำดับโดยไม่ใช่บรรทัดล่างในการนำข้อมูลเข้าออกในตอมักระวัง

เช่น

50.	2.5	
20.	3.	
40.	5.5	
HRS	RATE	

ต้องการ

40.	5.5
20.	3.
50.	2.5
HRS	RATE

จากคำสั่ง DIMENSION HRS(3),RATE(3)

READ(5,3)HRS,RATE

3 FORMAT(F3.0,F5.2)

จะให้ผลการอ่าน

HRS RATE

ซึ่งไม่ตรงตามความต้องการ

40.	3.
5.5	50.
20.	2.5

๕ ๕

หนึ่งเพราะ READ(5,3)HRS,RATE

๕

หนึ่งเหมือนกับ READ(5,3)(HRS(I),I=1,3),(RATE(I),I=1,3)

หรือ

READ(5,3)HRS(1),HRS(2),HRS(3),RATE(1),RATE(2),RATE(3)

คำสั่งที่จะตรงตามต้องการคือ

DIMENSION HRS(3),RATE(3)

READ(5,3)(HRS(I),RATE(I),I=1,3)

3 FORMAT(F3.0,F5.2)

คำสั่ง READ(5,3)(HRS(I),RATE(I),I=1,3) เหมือนกับ

READ(5,3)HRS(1),RATE(1),HRS(2),RATE(2),HRS(3),RATE(3)

ตัวอย่าง

WRITE(6,15)A,(K(J),J=1,3),(X(I),B(I),I=1,9)

รายการตัวแปรคือ A,K(1),K(2),K(3),X(1),B(1),X(2),B(2),...,X(9),B(9)

6.3 โปรแกรมตัวอย่างเพิ่มเติม

6.3.1 การหาสมาชิกตัวที่มีค่าสูงสุดของแถวลำดับ

```
DIMENSION G(10)
```

```
:
```

```
XLARG=G(1)
```

```
DO 5 I=2,10
```

```
IF(XLARG.LT.G(I))XLARG=G(I)
```

```
5 CONTINUE
```

```
WRITE(6,4)XLARG
```

```
:
```

ถ้าต้องการหาสมาชิกตัวที่มีค่าสูงสุดและตำแหน่งของค่าสูงสุดนั้นด้วย

```
DIMENSION G(10)
```

```
XLARG=G(1)
```

```
MAX=1
```

```
DO 5 I=2,10
```

```
IF(XLARG.GE.G(I))GO TO 5
```

```
XLARG=G(I)
```

```
MAX=I
```

```
5 CONTINUE
```

```
WRITE(6,4)MAX, XLARG
```

```
4 FORMAT(1X,'THE BIGGEST G IS G(',I2,') = ',F5.2)
```

```
:
```

6.3.2 การใช้ค่าจากตาราง

สมมติว่าการคำนวณค่าขนส่งสินค้าซึ่งเท่ากับน้ำหนัก(ปอนด์)คูณกับค่าส่ง/ปอนด์ และอัตราค่าส่งของ/ปอนด์นั้นขึ้นอยู่กับโซนของที่หับของปลายทาง ตารางแสดงอัตราดังกล่าวคือ

รหัสโซน : IZ	ค่าขนส่ง/ปอนด์ : R
1	.5
2	.75
3	1.05
4	1.25
5	1.40
6	1.70

โปรแกรมเพื่ออ่านโซนที่หับของปลายทางและน้ำหนักของของที่ส่ง (W) และพิมพ์โซนน้ำหนักและค่าส่งของมัน ๆ คือ

```
REAL R(6) หรือ DIMENSION R(6)
READ(5,5)(R(I),I=1,6)
5 FORMAT(6F5.2)
1 READ(5,6,END=60) IZ,W
6 FORMAT(I3,F5.2)
COST=W*R(IZ)
WRITE(6,11) IZ,R(IZ),W,COST
11 FORMAT(3X,I1,3F6.2)
GOTO1
60 STOP
END
```

		R(1)	R(2)	R(6)
บัตรข้อมูลเข้า	บัตรที่ 1	.50	.75 1.70
	บัตรที่ 2	2	6.25	
	บัตรที่ 3	5	10.0	
		IZ (โชน)	W (น้ำหนัก)	

6.3.3 ตารางแจกแจงความถี่

ถ้าคะแนนมีค่าจาก 1-100 และบัตร 1 บัตรบันทึกคะแนนของนักศึกษาคนหนึ่ง ๆ สมมติว่าไม่ทราบจำนวนนักศึกษา เราต้องการนับจำนวนนักศึกษาที่ได้คะแนนต่าง ๆ กัน จะใช้สมาชิก 100 ตัวของแถวลำดับ K ในการนับสิ่งที่ต้องการ โดยเริ่มจากการกำหนดค่าของสมาชิกทั้ง 100 ตัวให้มีค่าเริ่มต้นเป็นศูนย์ก่อน ถ้าคะแนนที่อ่านเป็น 1 จะทำให้ K(1) มีค่าเพิ่มขึ้น 1 ถ้าคะแนนที่อ่านเป็น 50 จะทำให้ K(50) มีค่าเพิ่มขึ้น 1 เป็นต้น

C FREQUENCY TABLE (ONE-WAY TABLE)

DIMENSION K(100)

DATA K/100*0/

4 READ(5,5,END=70)IG

5 FORMAT(I3)

WRITE(6,6)IG

6 FORMAT(I6)

K(IG)=K(IG)+1

GO TO 4

70 WRITE(6,7)

7 FORMAT(1X,'GRADES FREQUENCY'//)

Do 80 I=1,100

IF(K(I).EQ.0)GO TO 80

WRITE(6,8)I,K(I)

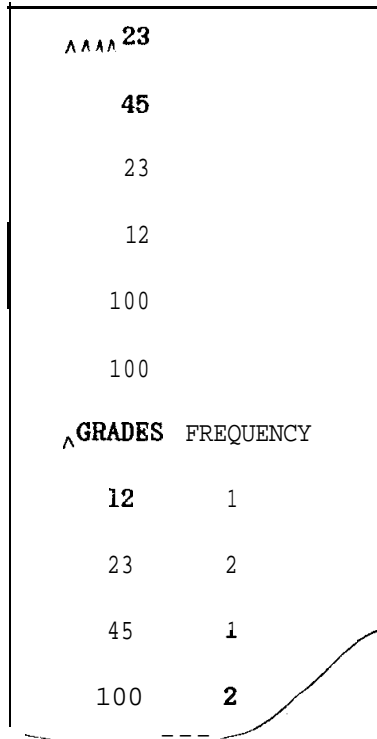
8 **FORMAT(15,18)**

80 CONTINUE

STOP

END

ตัวอย่างข้อมูลออก

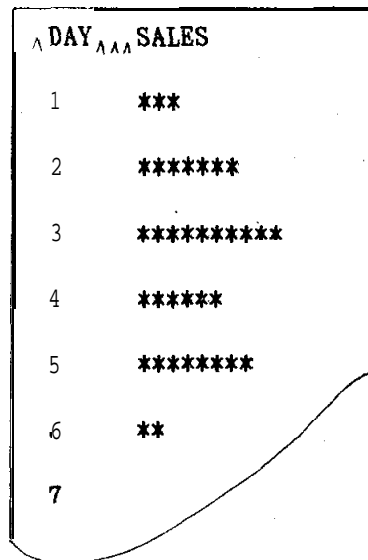


8.3.4 กราฟแท่ง (Bar graph)

สมมติยอดขายในแต่ละวันเป็นดังนี้

Day	Sales
1	3
2	7
3	10
4	6
5	8
6	2
7	0

ต้องการพิมพ์



เราเขียนโปรแกรมได้ดังนี้

```
DATA STAR/'*'/
WRITE(6,1)
1 FORMAT('1DAY',T8,'SALES')
DO 20 IDAY=1,7
READ(5,2)KSALES
2 FORMAT(I2)
IF(KSALES.NE.0)GO TO 15
WRITE(6,3)IDAY
3 FORMAT(T2,I1)
GO TO 20
15 WRITE(6,4)IDAY,(STAR,J=1,KSALES)
4 FORMAT(T2,I1,T8,20A1)
20 CONTINUE
STOP
END
```

6.3.5 การเขียนกราฟ

ตัวอย่างการเขียนกราฟของฟังก์ชัน $y=x^2+x-6$, $x=-4$ (.) 3.6 และของ $y=8(\sin x+2)$, $x=0$ (.) 7.6

```
DIMENSION ILINE(28),JLINE(28)
DATA IBLANK,ISTAR/' ','*'/
WRITE(6,9)
9 FORMAT('1',T4,'X',T9,'Y',T35,'X',T40,'Y')
XP=-4
XS=0
```

18 YP=XPSXP+XP-6

YS=SIN(XS)

DO 15 I=1,28

ILINE(I)=IBLANK

15 JLINE(I)=IBLANK

JP=YP+7.25

JS=(YS+2)*8

ILINE(JP)=ISTAR

JLINE(JS)=ISTAR

WRITE(6,11)XP,YP,ILINE,XS,YS,JLINE

11 FORMAT(2F5.1,2X,28A1,T32,2F5.1,28A1)

XP=XP+.4

XS=XS+.4

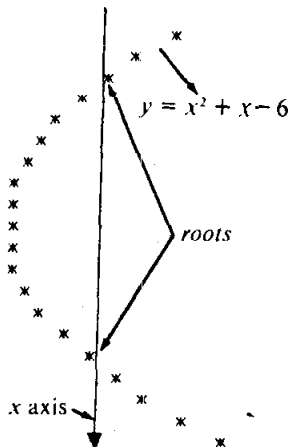
IF(XP.LE.3.6)GO TO 18

STOP

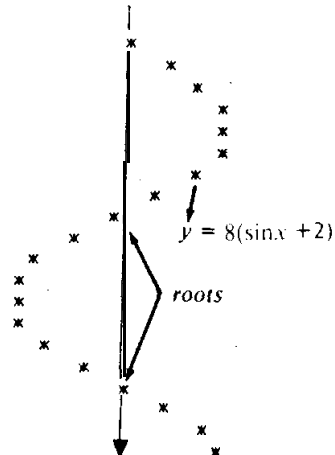
END

ผลการทำงาน

X	Y
-4.0	6.0
-3.6	3.4
-3.2	1.0
-2.8	-1.0
-2.4	-2.6
-2.0	-4.0
-1.6	-5.0
-1.2	-5.8
-.8	-6.2
-.4	-6.2
.0	-6.0
.4	-5.4
.8	-4.6
1.2	-3.4
1.6	-1.8
2.0	-.0
2.4	2.2
2.8	4.6
3.2	7.4
3.6	10.6



X	Y
0.0	0.0
.4	.4
.8	.7
1.2	.9
1.6	1.0
2.0	.9
2.4	.7
2.8	.3
3.2	-.1
3.6	-.4
4.0	-.8
4.4	-1.0
4.8	-1.0
5.2	-.9
5.6	-.6
6.0	-.3
6.4	.1
6.8	.5
7.2	.8
7.6	1.0



6.4 คำสั่งในภาษาฟอร์แทรนเพิ่มเติม

6.4.1 การกำหนด NAMELIST

การกำหนด NAMELIST ทำให้คำสั่ง READ และคำสั่ง WRITE ไม่จำเป็นต้องใช้คำสั่ง FORMAT ได้

รูปทั่วไปคือ

```
NAMELIST /ชื่อ1/รายการ1/[ชื่อ2/รายการ2/...]
```

โดยที่ ชื่อ1 เป็นชื่อของรายชื่อตัวแปรในรายการ1 ชื่อ1ตั้งตามหลักการตั้งชื่อตัวแปร รายการ1 เป็นรายชื่อของตัวแปรหรือแถวลำดับ รายชื่อค้นด้วยเครื่องหมายจุลภาค (,) ตัวแปรและแถวลำดับอาจปรากฏในรายการต่าง ๆ มากกว่า 1 ครั้งได้

ตัวอย่าง การใช้ NAME1 ในคำสั่ง READ/WRITE แทนที่เลขประจำคำสั่ง

```
WRITE(6,NAME1) หมายความว่าให้พิมพ์ค่าของตัวแปรที่อยู่ใน NAME1
```

คำสั่งระบุที่อ้างถึงตัวแปรที่อยู่ใน NAMELIST จะต้องใส่ไว้ก่อนคำสั่ง NAMELIST

NAMELIST และคำสั่ง READ

```
CHARACTER*10 PART
```

```
INTEGER TABLE(3)
```

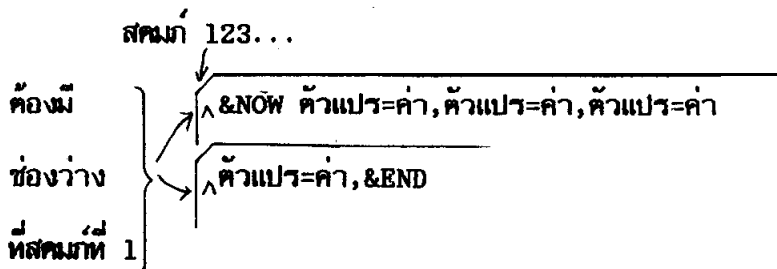
```
LOGICAL PULSE
```

```
NAMELIST /NOW/TABLE,PULSE,PART,F/
```

```
NAMELIST /THEN/PULSE,PART/
```

```
READ(5,NOW)
```

บัตรข้อมูลเข้าต้องมีรูปแบบดังนี้



ตัวอย่าง READ(5, NOW)

บรรทัดที่ 2 TABLE=3, -2, 10, &END

บรรทัดที่ 1 &NOW PULSE=.TRUE., PART='BATTERIES'

ผลการอ่านและการเก็บค่าในหน่วยความจำหลัก

PULSE=.TRUE. PART=BATTERIES F ไม่ได้ถูกอ่าน

TABLE(1)=3 TABLE(2)=-2 TABLE(3)=10

ให้สังเกตว่าข้อมูลบรรทัดข้อมูลเข้านั้นไม่ต้องเรียงลำดับเดียวกันในคำสั่ง NAMELIST และไม่จำเป็นว่าตัวแปรทุกตัวต้องถูกกำหนดค่าในบรรทัดข้อมูลเข้า เช่น ตัวแปร F ถูกกำหนดใน

คำสั่ง NAMELIST แต่มันไม่ถูกอ่านค่า

NAMELIST กับคำสั่ง WRITE

ถ้าเราเขียนคำสั่งต่อจากข้างต้นด้วย

WRITE(6, NOW)

WRITE(6, THEN)

ค่าของตัวแปรในรายการใน NOW และ THEN จะถูกพิมพ์ตามลำดับที่ปรากฏใน NAMELIST และในการพิมพ์จะพิมพ์ชื่อตัวแปรหรือชื่อแถวลำดับไว้พร้อมกับค่าที่แสดงด้วย ตัวอย่างของข้อมูลออกอาจเป็นดังนี้

&NOW

TABLE=3, -2, 10 ,PULSE=T ,PART=BATTERIES ,F= ,&END

&THEN

PULSE=T ,PART=BATTERIES ,&END

6.4.2 การใช้ format code ข้าง

ในกรณีจำนวนตัวแปรในรายชื่อตัวแปร (n ตัว) นี้นั้นมากกว่าจำนวน format code สำหรับข้อมูล (m ตัว) (เช่น I, F, A, E, D, L) ในคำสั่ง FORMAT หลังจากอ่านหรือพิมพ์ m ตัวแล้ว การอ่านและการพิมพ์จะเริ่มขึ้นที่ระเบียบใหม่ (บรรทัดหรือบรรทัดใหม่) และการอ่าน format code จะเริ่มใหม่ที่วงเล็บเปิดขวาสุดเพื่อจัดการกับตัวแปร (n-m) ตัวที่เหลืออยู่ในรายชื่อตัวแปร

ตัวอย่าง WRITE(6,5)I,J,K,L,M,N



5 FORMAT(T5,I1,I2,(1X,I2,I2))

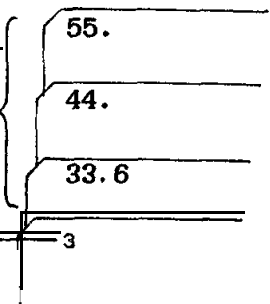
↑
วงเล็บเปิดขวาสุด

ตัวอย่าง READ(5,10)N,(A(I),I=1,N)

10 FORMAT(I2/(5X,F5.0))

↑
วงเล็บเปิดขวาสุด

ค่าในหน่วยความจำหลัก



N=3

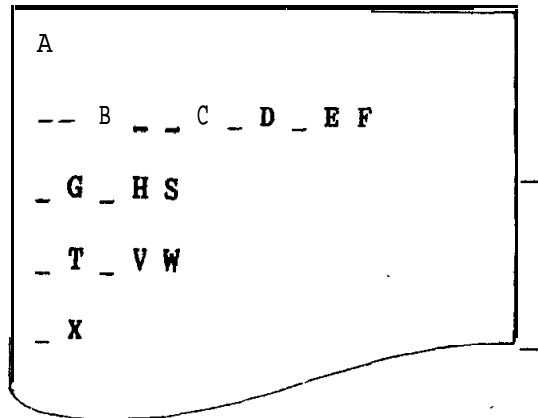
A(1)=33.6

A(2)=44.

A(3)=55.

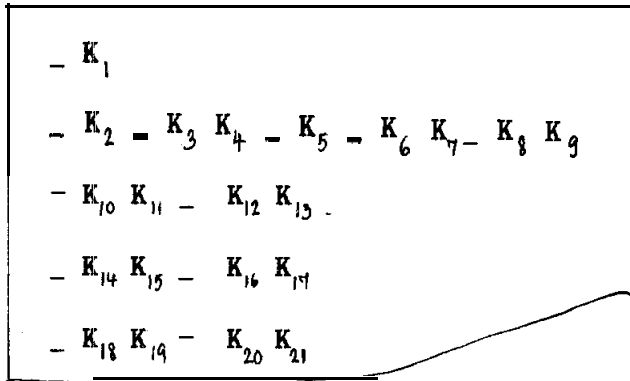
ตัวอย่าง WRITE(6,5)A,B,C,D,E,F,G,H,S,T,V,W,X

5 FORMAT(F5.0/2(2X,F3.0),2(1X,F3.0),F3.0)



ตัวอย่าง WRITE(6,11)(K(I),I=1,21)

11 FORMAT(1X,I1/1X,2(I1,1X,I1),2(1X,I1,I1))

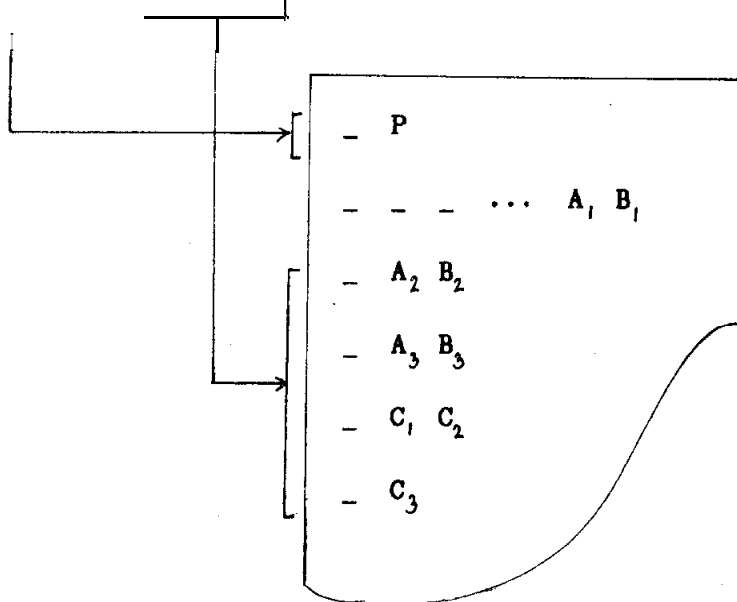


FORMAT ที่ 11 เหมือนกับ

(1X, I1/1X, I1, 1X, I1, I1, 1X, I1, 1X, I1, I1, 1X, I1, I1/1X, I1, I1, 1X, I1, I1/
 1X, I1, I1, 1X, I1, I1/1X, I1, I1, 1X, I1, I1)

ตัวอย่าง WRITE(6, 12)P, (A(I), B(I), I=1, 3), (C(I), I=1, 3)

12 FORMAT(1X, F5.1/T20, (1X, 2F5.0))



FORMAT ที่ 12 เหมือนกับ

(1X, F5.1/T20, 1X, 2F5.0/1X, 2F5.0/1X, 2F5.0/1X, 2F5.0/1X, 2F5.0)

ตัวอย่าง ปัญหาที่แสดงการใช้ format code ซ้ำ

ในการพิมพ์ตารางเรียน บัณฑิต 1 จะมีตารางเรียนของนักศึกษาคนหนึ่งในลำดับหนึ่ง
 บัณฑิตใบแรกมีตารางวันจันทร์ (เหมือนกับของวันพุธและศุกร์) และบัณฑิตที่ 2 มีตารางวันอังคาร
 (เหมือนกับของวันพฤหัสบดี) ต้องการเขียนโปรแกรมเพื่อพิมพ์ตารางเรียนของนักศึกษาแต่ละคน
 ตามรูปแบบดังนี้

TIME	M	T	W	TH	F
8	DP101		DP101		DP101
9		FH100		FH100	
10	MS312		MS312		MS312
11		BY101		BY101	
12	EN202		EN202		EN202
13					
14					

TOTAL CLASS HOURS IS 13

บัณฑิตแต่ละบัณฑิตแบ่งเป็น 7 พิลด์ ซึ่งลำดับของพิลด์จะแสดงลำดับของเวลาเรียนตั้งแต่

8.00-14.00 น.

เรียนวันอังคาร, พฤหัสบดี

เรียนวันจันทร์, พุธ, ศุกร์

เวลาเรียน

		FH100		BY101		
DP101		MS312		EN202		
8	9	10	11	12		


```

DOUBLE PRECISION WFM(7), TTH(7), BLANK

DATA BLANK/' '/

5 READ(5,1,END=8)(WFM(I),I=1,7),(TTH(J),J=1,7)

1 FORMAT(7A5)
  I=0
  DO 10 J=1,7

    IF(WFM(J).NE.BLANK)I=I+3

    IF(TTH(J).NE.BLANK)I=I+2

10 CONTINUE

    WRITE(6,2)(J,(WFM(J-7),TTH(J-7),L=1,2),WFM(J-7),J=8,14),I

2 FORMAT('1'//T8,'TIME',T20,'M',T30,'T',...,T60,'F'//
*7(T9,I2,T13,5A10/),T25,'TOTAL CLASS HOURS IS',I3)
GO TO 5
8 STOP

END

```

พิมพ์ 7 บรรทัด คูณให้ขึ้นบรรทัดใหม่

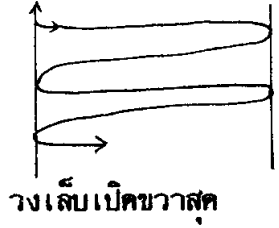
ถ้าไม่ต้องการนับจำนวนชั่วโมงที่เรียนในสัปดาห์คือไม่ต้องกาาพิมพ์ 'TOTAL CLASS HOURS IS' เราจะตัดคำสั่ง I=0 และคำสั่ง DO 10 J=1,7 ถึง 10 CONTINUE ออกแล้วเขียนคำสั่ง WRITE ใหม่ดังนี้

```

WRITE(6,2)(J,(WFM(J-7),TTH(J-7),L=1,2),WFM(J-7),J=8,14)

2 FORMAT('1'//T8,'TIME',T20,'M',...//(T9,I2,T13,5A10))

```



แบบฝึกหัดที่ 6

1. จงบอกจำนวนบัตร/บรรทัดที่จะถูกอ่าน/พิมพ์ จากคำสั่งต่อไปนี้

1.1) DO 10 I=1,5

READ(5,6)A(I)

6 FORUAT(3F5.0)

10 CONTINUE

1.2) DO 10 J=1,9

READ(5,6)A(J),B(J)

6 FORMAT(F5.3)

10 CONTINUE

1.3) DO 10 K=1,6,2

READ(5,7)(A(I),I=1,K)

7 FORMAT(3F5.0)

10 CONTINUE

1.4) REAL A(4),J

DO 5 I=1,4

5 WRITE(6,6)J,A

6 FORMAT(1X,2F4.1)

2. จงเปลี่ยน implied DO ให้เป็นคำสั่ง DO

2.1) READ(5,5)(A(I),I=1,6)

5 FORMAT(2F5.0)

2.2) READ(5,6)(A(I),K(I),I=2,9,2)

6 FORMAT(2(F5.1,I2))

3. จากค่าในแถวลำดับ A,B และค่าของ K ที่กำหนดให้ จงแสดงผลการพิมพ์ (ระบุตำแหน่งของการพิมพ์ให้ชัดเจน) จากคำสั่ง WRITE เมื่อคำสั่ง FORMAT เปลี่ยนไป

แถวลำดับ A

1.5	-3.2	3.	4.8	.34
-----	------	----	-----	-----

แถวลำดับ B

-1.	2.	3.
-----	----	----

ค่าคงที่ K=3

3.1) WRITE(6,11)(I,I=1,5)

a) 11 FORMAT(I1)

b) 11 FORMAT(I2,I2)

c) 11 FORMAT(1X,20I1)

d) 11 FORMAT(4X,2I2,('+',2I1))

5.1) INTEGER C

REAL X(5)

C=1

3 READ(5,1)X(C)

IF(C.EQ.5)GO TO . . .

C=C+1

GOTO 3

1 FORMAT(F6.1)

5.2) INTEGER C

REAL X(5)

C=1

3 READ(5,1)X(C)

IF(C.EQ.5)GO TO . . .

C=C+2

GOTO 3

1 FORMAT(F6.1)

5.3) REAL X(3),Y(2)

READ(5,1)X,Y

1 FORMAT(4F6.1)

5.4) REAL X(4),Y(3)

READ(5,1)X

READ(5,1)Y

1 FORMAT(4F6.1)

5.5) REAL X(3),Y(3)

READ(5,1)(X(I),Y(I),I=1,3)

1 FORMAT(4F6.1)

5.6) REAL X(3),Y(3)

READ(5,1)(X(I),I=1,3),(Y(I),I=1,3)

1 FORMAT(4F6.1)

5.7) REAL X(5)

READ(5,1)X(5)

READ(5,1)X(1),X(4)

X(3)=X(1)*5

X(2)=X(1)+X(5)

1 FORMAT(F6.1)

5.8) REAL X(16)

READ(5,1)(X(I),A,I=1,4)

1 FORMAT(3F6.1)

6. คำสั่งอ่านจะอ่านกี่บรรทัดและคำสั่งพิมพ์จะพิมพ์กี่บรรทัด

6.1) READ(5,5)(A(I),I=1,5),(B(J),J=1,3)

5 FORMAT(3F5.2)

8. จากโปรแกรม 4 โปรแกรม จงหา execution error หรือ run-time error

8.1) DIMENSION A(100)

I=1

x=4

A(J)=X**2+2*X+I

STOP

END

8.2) DIMENSION A(100)

I=1

x=4

A(I)=X**A(J)

STOP

END

8.3) DIMENSION A(100)

WRITE(6,5)A

READ(5,5)(A(I),I=1,110)

5 FORMAT(10F5.0)

STOP

END

8.4) DIMENSION A(18)

DO 10 I=1,5

READ(5,7)A(I),A(I+1)

7 FORMAT(2F5.0)

10 CONTINUE

STOP

END

9. จากคำสั่งที่กำหนดให้ จงระบุตำแหน่งในบัตรหรือตำแหน่งพิมพ์บนกระดาษของตัวแปรในรายชื่อตัวแปร

9.1) INTEGER M(20)

WRITE(6,5)(N,L,(M(J),J=1,4),K,K=1,2)

5 FORMAT(1X,3I3/(1X,4I2))

9.2) REAL T(10)

READ(5,6)A,B,(T(I),I=1,5)

a) 6 FORMAT(3F5.1)

b) 6 FORMAT(3F5.1/F5.1)

c) 6 FORMAT(F5.1/(2F5.1))

9.3) WRITE(6,4)(A(I),I=1,7)

4 FORMAT(1X,2F4.0/1X,2(F3.1,1X)/2(1X,F3.0))

10. จงหาข้อผิดพลาดอย่างน้อย 5 แห่งในโปรแกรมต่อไปนี้

```

REAL A(10),B(20),IBIG
DIMENSION A(10),C(3)
DATA C/4,5,6/
20 READ(5,1)(A(I),I=1,10)
DO 3 I=1,10
IF(A(I).GE./IBIG)IBIG=A(I)
IF(A(I).LT ISMALL)ISMALL=A(I)
30 CONTINUE
IF(IBIG=0)GOTO 20
STOP
END

```

11. จากแฟ้มข้อมูลของนักศึกษาที่ลงทะเบียนเรียนวิชา CS 215 จำนวน 300 คน แต่ละคนมีข้อมูลที่บันทึกอยู่ในบัตร 1 บัตร ซึ่งมีรูปแบบดังนี้

ฟิลด์	สตริง	รายการข้อมูล
1	1-7	รหัสประจำตัว
2	8-39	ชื่อ-นามสกุล
3	40	คณะ 1=นิติศาสตร์ : 5=วิทยาศาสตร์ 6=รัฐศาสตร์ 7=เศรษฐศาสตร์
4	41	เพศ 1=ชาย 2=หญิง
5	42-43	อายุ

จงเขียนโปรแกรมเพื่อ

- 1) นับจำนวนนักศึกษาหญิงคณะวิทยาศาสตร์ และพิมพ์รายชื่อของแต่ละคนด้วย
- 2) คำนวณหาอายุเฉลี่ยของนักศึกษาในข้อ 1)
- 3) นับจำนวนนักศึกษานในแต่ละคณะ

กำหนดรูปแบบการพิมพ์ผลดังนี้

CS 215

TABLE 1

FEMALES IN FACULTY OF SCIENCE

NUMBER	NAME	AGE
AVERAGE AGE = _____		

TABLE 2

NUMBER OF STUDENTS CLASSIFIED BY FACULTY

FACULTY	NO. OF STUDENTS
LAW	_____
BUSINESS	_____
:	:
ECONOMICS	_____
TOTAL	_____

12. จงเขียนโปรแกรมเพื่อคำนวณหา Sample linear regression equation และ Analysis of variance table สำหรับข้อมูลต่อไปนี้

$n=12$

X	1.3	3.1	2.1	1.3	-5.2	-6.2	-2.1	-5.2	-3.9	-2.1	1.2	.9
Y	5.1	6.9	6.1	4.9	.8	.1	2.3	1.1	1.2	2.1	4.2	4.2

กำหนดให้บันทึกค่า x ทั้ง 12 ค่าไว้ในบัตรที่ 1 และค่า y ทั้ง 12 ค่าในบัตรที่ 2

กำหนดรูปแบบของกาพิมพ์ผล

SAMPLE LINEAR REGRESSION

$$Y = (a) + (b)x$$

ANALYSIS OF VARIANCE

SOURCE	D.F.	SUM OF SQUARES	MEAN SQUARE	F-RATIO
REGRESSION	(1)	(SSR)	(MSR=SSR)	(F)
ERROR	(n--2)	(SSE)	(MSE)	
TOTAL	(n--1)	(SSY)		

กำหนดสูตร

$$\bar{x} = \frac{\sum_{i=1}^m x_i}{n}, \quad SSX = \sum x_i^2 - (\sum x_i)^2/n$$

$$\bar{y} = \frac{\sum_{i=1}^m y_i}{n}, \quad SSY = \sum y_i^2 - (\sum y_i)^2/n$$

$$SXY = \sum x_i y_i - (\sum x_i)(\sum y_i)/n, \quad \sum = \sum_{i=1}^m$$

$$b = SXY/SSX, \quad a = \bar{y} - b\bar{x}$$

$$SSR = b * SXY, \quad SSE = SSY - SSR$$

$$MSR = SSR, \quad MSE = SSE/(n-2)$$

$$F = MSR/MSE$$

13. เงินเดือนของพนักงานบริษัท XYZ ขึ้นอยู่กับชนิดของงาน จำนวนปีที่ทำงานกับบริษัท ระดับความรู้ ความเอาใจใส่ในการทำงาน เงินฐาน (base pay) ของทุกคนเท่ากันหมดคือ 3000 บาท และบวกด้วยจำนวนเงินตามเปอร์เซ็นต์ของเงินฐาน ตามรายการต่อไปนี้

1) ชนิดของงาน (TYPE OF JOB)	%	2) ระดับความรู้ (EDUCATION)	%
1	0	1 (ประกาศนียบัตร)	0
2	5	2 (ประกาศนียบัตรชั้นสูง)	10
3	15	3 (ปริญญาตรี)	15
4	25	4 (ปริญญาโท)	25
5	50	5 (ปริญญาเอก)	50

3). จำนวนปีทำงาน (NO.OF YEARS) %	4) ความสามารถและเอาใจใส่งาน(MERIT) %
0-10 5	0 (ไม่มี) 0
แต่ละปีถัดไป 4	1 (ดี) 10
	2 (ดีมาก) 25

จงเขียนโปรแกรมเพื่ออ่านข้อมูลที่ป้อนเข้ามาเป็นตัวเลขที่แสดง 4 รายการข้างต้นของพนักงานแต่ละคน กำหนดให้พิมพ์ข้อมูลออกมาในรูปแบบต่อไปนี้

COMPANY XYZ

SALARY REPORT

^d
 คนที่
 ↓
 NUMBER TYPE OF JOB NO.OF YEARS EDUCATION MERIT PERCENT TO SALARY

BE ADDED

:

:

: