

บทที่ 5 ข้อมูล

- 5.1 ปัญหาตัวอย่าง
- 5.2 วิธีการเก็บและจัดการกับข้อมูล
 - 5.2.1 วิธีการเก็บเลขจำนวนภายในหน่วยความจำ
 - 5.2.2 ข้อมูลอักขระและ A-format code
 - 5.2.3 คำสั่ง CHARACTER
 - 5.2.4 เลขจำนวนจริงในรูปเลขชี้กำลังและ E-format code
 - 5.2.5 เลขจำนวนจริงชนิด Double precision และ D-format code
 - 5.2.6 ข้อมูลตารางและ L-format code
 - 5.2.7 ข้อมูลเชิงซ้อน

แบบฝึกหัดที่ 5

บทที่ 5 ข้อมูล

5.1 ปัญหาตัวอย่าง

ระเบียบของนักศึกษาแต่ละคนจะประกอบด้วยชื่อและคะแนนสอบไล่ของนักศึกษานั้น
จึงเขียนโปรแกรมเพื่อกำหนดเกรดตัวอักษร (A-F) ตามเกณฑ์ข้างล่าง และกำหนดให้พิมพ์
ผลลัพธ์ในหน้าจอเดียวกับที่แสดงไว้ด้วย

คะแนน	เกรด
คะแนน ≥ 90	A
$80 \leq$ คะแนน < 90	B
$60 \leq$ คะแนน < 80	C
$50 \leq$ คะแนน < 60	D
คะแนน < 50	F

ตัวอย่าง การพิมพ์ผล

MALL	83	B
USA	83	B
ORADA	33	F
ONJIRA	91	A
SAWAT	92	A
HIGHEST SCORE WAS OBTAINED BY SAWAT		
SCORE WAS OBTAINED BY OTHER STUDENT(S)		

ข้อความนี้จะพิมพ์เมื่อนักศึกษาได้คะแนนสูงสุดอย่างน้อย 2 คน มิฉะนั้นจะไม่พิมพ์

DOUBLE PRECISION NAME, I HOLD

LOGICAL P

DATA A,B,C,D,F/'A','B','C','D','F'/

P=.TRUE.

AMAX=0.

15 READ(5,5,END=6)NAME, SCORE

5 FORMAT(A8, F5.1)

K=SCORE/10.+1.

K1=K-5

GO TO (1,2,2,3,4,4),K1

GRADE=F

Go TO 10

1 GRADE=D

Go To 10

2 GRADE=C

Go To 10

3 GRADE=B

Go To 10

4 GRADE=A

10 WRITE(6,11)NAME, SCORE, GRADE

11 FORMAT(T10, A8, 3X, F6.1, 2X, A1)

IF(AMAX-SCORE)13,14,15

14 P=.FALSE.

Go To 17

		K1
A	K=10,11	6,5
B	K=9	4
C	K=8,7	3,2
D	K=6	1
F	K=1,2,3,4,5	คำว่า

13 **AMAX=SCORE**

P=.TRUE.

17 **IHOLD=NAME**

GO TO 15

6 WRITE(6,7) IHOLD

7 FORMAT(T10, 'HIGHEST GRADE WAS OBTAINED BY', 1X, A8)

IF(P)GO TO 9

WRITE(6,8)

8 FORMAT(T10, 'SCORE WAS OBTAINED BY OTRRR STUDENT(S)')

9 STOP

END

จากตัวอย่างข้างต้นมีสิ่งใหม่ ๆ 3 สิ่งคือ

1. คำสั่ง **DOUBLE PRECISION**
2. คำสั่ง **LOGICAL**
3. A-format. code

5.2 วิธีการเก็บและจัดการกับข้อมูล

ชนิดของข้อมูลที่จะกล่าวถึงคือ

1. ข้อมูลเชิงตัวเลข (Numeric data)
2. ข้อมูลอักขระ (Character data)
3. ข้อมูลตรรกะ (Logical data)
4. ข้อมูลเชิงซ้อน (Complex number)

5.2.1 วิธีการเก็บเลขจำนวนภายในหน่วยความจำ

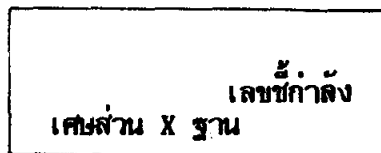
5.2.1.1 เลขจำนวนเต็ม (Integer constant)

คอมพิวเตอร์ส่วนมากเก็บเลขจำนวนเต็มในระบบฐานสอง เลขจำนวนเต็มหนึ่งจำนวนไว้ที่เก็บในหน่วยความจำเท่ากับ 1 คอมพิวเตอร์เวิร์ด (word) ซึ่งประกอบด้วย กีบิต (bit) นั้นขึ้นอยู่กับระบบคอมพิวเตอร์ ทั้งนี้อาจมีตั้งแต่ 16-60 บิต ทั้งนี้ตัวอย่างข้างล่างนี้ แสดงค่าสูงสุดและต่ำสุดของเลขจำนวนเต็มขึ้นอยู่กับระบบคอมพิวเตอร์

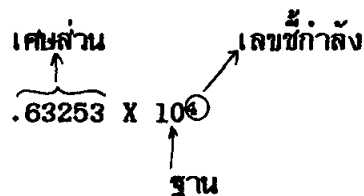
คอมพิวเตอร์	จำนวนบิตใน 1 เวิร์ด	ค่าสูงสุดของเลข จำนวนเต็ม	จำนวนหลักของ เลขฐานสิบ
IBM 1130	16	$2^{15}-1$	5
IBM 370/30XX/43XX	32	$2^{31}-1$	10
VAX 11/750, Micro Vax II	48	$2^{38}-1$	15
Burroughs 6700	48	$2^{38}-1$	15
CDC Cyber 72	60	$2^{59}-1$	18

5.2.1.2 เลขจำนวนจริง (Real constant)

เลขจำนวนจริงเก็บอยู่ในรูปที่เรียกว่า floating point ซึ่งอยู่ในรูป



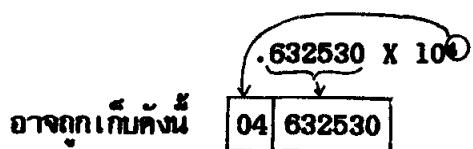
ตัวอย่างเช่นในระบบเลขฐานสิบ เลขจำนวน 6325.3 อาจเขียนในรูป



ดังนั้นในการเก็บเลขจำนวนจริงจึงมีการแบ่งเวิร์ดออกเป็น 2 ส่วนคือส่วนหนึ่งเพื่อเก็บ

เลขชี้กำลัง (exponent) และอีกส่วนหนึ่งเพื่อเก็บเศษส่วน (fraction)

ตั้งใหม่เลขจำนวน 6325.3 ซึ่งเขียนได้ในรูป



พิสัย (range) ของเลขจำนวนจริงที่จะเก็บได้ขึ้นอยู่กับจำนวนบิตที่แบ่งเป็นแต่ละส่วนใน 2 ส่วนของที่ใช้เก็บเลขจำนวนจริงหนึ่งจำนวน ทั้งนี้ขึ้นอยู่กับคอมพิวเตอร์ ดังตัวอย่างในตารางข้างล่างนี้

คอมพิวเตอร์	จำนวนบิต	พิสัยของเลขที่ก่าตั้ง	จำนวนตัวเลขในส่วนของแต่ละส่วน (จำนวนตัวเลขที่สำคัญ)
IBM 1130	32	$10^{38}-10^{-39}$	7
IBM 370/30XX/43XX	32	$10^{75}-10^{-78}$	7
Burroughs B 6700	48	$10^{67}-10^{-47}$	11
CDC Cyber 72	60	$10^{308}-10^{-308}$	15

5.2.1.3 ตัวเลขที่สำคัญ (Significant digit)

ตัวเลขที่สำคัญคือตัวเลขที่ใช้แสดงเลขจำนวนใด ๆ นั้นเอง เลขศูนย์ที่นำหน้าเลขจำนวนและเลขศูนย์ที่อยู่ถัดจากจุดทศนิยมนั้นไม่มีนัยสำคัญ เช่น

- 1001.56 มีตัวเลขที่สำคัญ 6 ตัว
 - 00012.4 } มีตัวเลขที่สำคัญ 3 ตัว
 - .000315 }
- (.000315 = .315 X 10⁻³)

คอมพิวเตอร์เก็บเลขจำนวนและจัดกระทำกับตัวเลขที่สำคัญที่มีจำนวนแน่นอนคือ n ตัว

ซึ่งค่าของ n นั้นขึ้นอยู่กับเครื่องคอมพิวเตอร์ เลขจำนวนใด ๆ หรือผลลัพธ์จากการกระทำหนึ่ง ๆ จะถูกแสดงโดยใช้ตัวเลขน้อยสำคัญ n ตัวนั้น ๆ

เนื่องจากคอมพิวเตอร์ไม่สามารถแสดงหรือเก็บเลขทศนิยมแบบไม่รู้จักได้ เลขทศนิยมเหล่านี้จะถูกปัดค่า (round off) การปัดค่าของเลขจำนวนนี้ให้เหลือตัวเลขน้อยสำคัญ n ตัวก็คือการตัดตัวเลขทั้งหมดที่อยู่ทางขวาของตัวเลขน้อยสำคัญตัวที่ n ออก แต่ถ้าตัวเลขตัวแรกที่ถูกต้องออกมีค่ามากกว่าหรือเท่ากับ 5 เราจะเพิ่ม 1 ให้แก่ตัวเลขตัวที่ n นอกนั้นเราจะตัดทิ้งไปเลย

ตัวอย่าง เลขจำนวนต่อไปนี้ถูกปัดค่าให้เหลือตัวเลขน้อยสำคัญ 7 ตัว ก่อนใช้กับคอมพิวเตอร์ (IBM 360/370)

41.239824	ถูกปัดเป็น 41.23982	=.4123982x10 ²
.0011145678	ถูกปัดเป็น .001114568	=.1114568x10 ⁻²
315.00075	ถูกปัดเป็น 315.0008	=.3150008x10 ³
1000001499999.98	ถูกปัดเป็น 1000001000000.00	=.1000001x10 ¹³

(ในเลขจำนวนสุดท้ายหลังการปัดค่าแล้ว ค่าของมันหายไปถึง 499,999!!)

5.2.2 ข้อมูลอักขระและ A-format code

ข้อมูลอักขระเป็นแถวลำดับของตัวอักษร ตัวเลขและตัวอักขระพิเศษ เช่น '1201 SUKHUMVIT 105' เป็นแถวของตัวอักขระที่ประกอบด้วยตัวอักขระ 18 ตัว คือตัวอักษร 9 ตัว ตัวเลข 7 ตัว และช่องว่าง (blank) 2 ที่ ซึ่งเราเรียกว่าสายวลีอักขระ (character string) หรือค่าคงที่ literal ตัวอย่างของข้อมูลชนิดนี้เช่น ชื่อที่อยู่ เบอร์โทรศัพท์ อาชีพ เป็นต้น เราไม่ทำการคำนวณโดยตัวค่าเนิการทางคณิตศาสตร์กับสายวลีอักขระ แม้ว่ามันอาจประกอบด้วยตัวเลขล้วน ๆ ก็ตาม จำนวนตัวอักขระที่เก็บได้ใน 1 เวิร์ดนั้นขึ้นอยู่กับเครื่องคอมพิวเตอร์ เช่น

คอมพิวเตอร์	ความยาวของเวิร์ด (จำนวนบิต)	จำนวนตัวอักขระ/เวิร์ด	Double precision
IBM 370/--/--	32	4	8
Burroughs B6700	64	6	12

ดังนั้นถ้าเรามีสายวลีอักขระที่ยาวเราต้องใช้หลายเวิร์ดในการเก็บ ตัวอย่างเช่นสายวลี
 อักขระ 1201 SUK HUMV IT1 05
 ในลักษณะดังนี้

1 2 0 1 S U K H U M V I T 1 0 5

การเก็บตัวอักขระแต่ละตัวของสายวลีอักขระภายในหน่วยความจำขึ้นอยู่กับระบบรหัสที่ใช้
 ซึ่งอาจเป็นรหัสเอ็บบิคดิค (EBCDIC) หรือรหัสแอสกี (ASCII) ทั้งนี้แล้วแต่ระบบคอมพิวเตอร์ที่ใช้
 เท่าที่กล่าวมาจะเห็นได้ว่าคอมพิวเตอร์มีวิธีการเก็บต่าง ๆ กันสำหรับข้อมูลต่างชนิดกัน ทั้งนี้
 การเก็บขึ้นอยู่กับวิธีการกำหนดตัวแปรที่ใช้เก็บข้อมูลนั้น ๆ เช่น

I=1234 เก็บเลขจำนวนเต็ม 1234 เพราะ I เป็นตัวแปรชนิด integer

X=1234 เก็บเลขจำนวนจริง 1234. เพราะ X เป็นตัวแปรชนิด real

DATA Y/'1234'/ เก็บตัวเลขแต่ละตัวตามรหัสของมันในที่ ๆ ชื่อ Y เครื่องหมาย
 ' ' (quotes) นี้ใช้แสดงข้อมูลอักขระ

ชนิดของข้อมูลอาจจะระบุได้โดยการใช้ format code ถ้าข้อมูลถูกอ่านจากบัตรข้อมูล

นั่นคือ I-format code บอกคอมพิวเตอร์ว่าต้องการเก็บข้อมูลในรูปของ integer

F-format code บอกคอมพิวเตอร์ว่าต้องการเก็บข้อมูลในรูปของ real

A-format code บอกคอมพิวเตอร์ว่าต้องการเก็บข้อมูลในรูปของรหัสของตัวอักขระ

5.2.2.1 การใช้คำสั่ง DATA ในการกำหนดตัวแปรให้เก็บสายวลีอักขระ

เราอาจทำได้ 2 วิธีคือ

1. นับจำนวนตัวอักขระ (n) ในสายวลีอักขระแล้วเขียน nH นำหน้าสายวลีอักขระนั้น

เช่น

DATA SUN,STAR,J/3HTHE,1H*,4HHHHH/

สำหรับคอมพิวเตอร์ 1 เวิร์ด = 4 ไบต์ จะเก็บสิ่งต่อไปนี้เป็นคือ

SUN เก็บ THE_A (A=ช่องว่าง)

STAR เก็บ *_{AAA}

J เก็บ HHHH

2. ใช้เครื่องหมายอะโพสโทรฟี่ เช่น

```
DATA SUE, SAM, JSUM/'A CA', '+', ' T.'/'
```

สำหรับคอมพิวเตอร์ที่ 1 เวิร์ด = 4 ไบต์ จะเก็บสิ่งต่อไปนี้คือ

SUE เก็บ A CA

S A M เก็บ +AAA

JSUM เก็บ AA T.

5.2.2.2 A-format code

A-format code ใช้เพื่ออธิบายฟิลด์ของข้อมูลเข้า/ออกซึ่งเป็นสายวลีอักขระ



โดยที่ w คือความยาวของฟิลด์

n เป็นจำนวนซ้ำ

ข้อมูลอักขระนั้นเก็บในตัวแปรชนิด integer หรือ real ก็ได้ (ในภาษาฟอร์แทรน 77

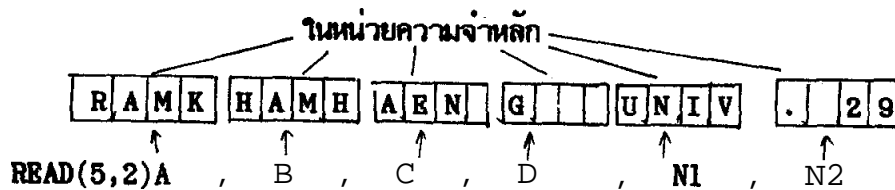
ข้อมูลอักขระต้องเก็บในตัวแปรอักขระเท่านั้น)

ในการนำข้อมูลเข้า 1) ถ้า w น้อยกว่าจำนวนตัวอักขระที่เก็บได้ใน 1 เวิร์ด (l)

อักขระ w ตัวจะถูกเก็บซิกซ์บายของเวิร์ดและหางขวาของเวิร์ดในหน่วยความจำจะเป็นช่องว่าง

(blank)

ตัวอย่าง สมมติว่าตัวแปรแต่ละตัวสามารถเก็บได้ 4 ตัวอักขระ (l=4)



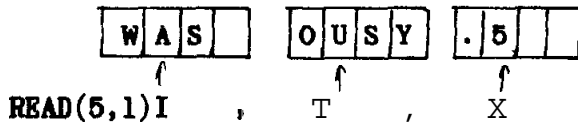
```
2 FORMAT(     A4, A4    , A3 , A2, A4    , A4)
```



2) ถ้า w มากกว่าจำนวนตัวอักขระที่เก็บได้ใน 1 เวิร์ด ตัวอักขระหางขวาของฟิลด์ l ตัวเท่านั้นที่จะถูกเก็บในตัวแปร

ตัวอย่าง สมมติว่าตัวแปร 1 ตัวเก็บได้ 4 ตัวอักษร ($l=4$)

ในหน่วยความจำหลัก



1 FORMAT(A7 , A5 ,A3)

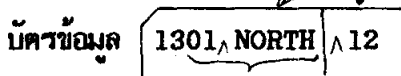


ตัวอย่าง DOUBLE PRECISION X, I, Y

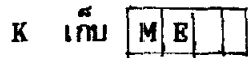
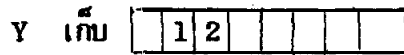
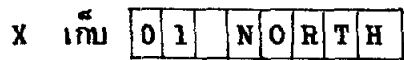
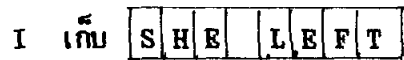
DATA I, K/'SHE LEFT', 'ME' /

READ(5,2) X, Y

2 FORMAT(A10, A3)



ผลการอ่านทำให้มีการเก็บสิ่งต่อไปนี้ในหน่วยความจำหลัก



ในการนำข้อมูลออก 1) ถ้า w มากกว่า l จะพิมพ์อักขระทั้ง l ตัวขีดขวาของฟิลด์โดยที่

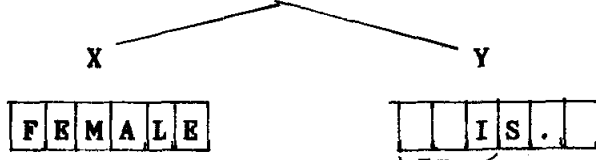
ทางซ้ายจะเป็นช่องว่าง

2) ถ้า w น้อยกว่า l ตัวอักษร w ตัวทางซ้ายของเวิร์คในหน่วย-

ความจำจะถูกนำออกมาแสดง

ตัวอย่าง สมมติว่าตัวแปร 1 ตัวเก็บได้ 6 ตัวอักษร ($l=6$, Burroughs B6700)

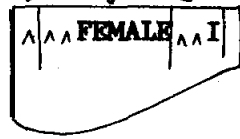
ในหน่วยความจำหลัก



WRITE(6,1)X,Y

1 FORMAT(1X,A8,A3)

ผลกาารพิมพ์



5.2.2.3 ข้อมูลอักขระและคำสั่ง IF

เราอาจเปรียบเทียบข้อมูลอักขระในคำสั่ง IF ได้ เมื่อทำการเปรียบเทียบลำดับของตัวอักขระจะเป็นดังนี้

ช่องว่าง < A < B < C < ... < Y < Z และ 0 < 1 < ... < 9

ตัวอย่าง ถ้า J เก็บ

B	E	T	A
---	---	---	---

K เก็บ

B	E	T	
---	---	---	--

L เก็บ

1	A	B	C
---	---	---	---

แล้ว J > K เพราะ ช่องว่าง < A

K < L เพราะ B < 1

ในการเปรียบเทียบตัวอักขระ คอมพิวเตอร์จะเปรียบเทียบครั้งละตัว การเปรียบเทียบสายวลีอักขระต้องระมัดระวังเพราะอาจเกิดข้อผิดพลาดดังเช่นตัวอย่างต่อไปนี้

ตัวอย่าง ในการตรวจว่าตัวแปร A เก็บคำว่า CAR ไว้หรือไม่ คำสั่งต่อไปนี้อาจใช้ไม่ได้กับระบบคอมพิวเตอร์บางระบบในคอนคอม โพลีไปแกม

```
IF(A.EQ.'CAR')GO TO 20
```

คำสั่ง IF(A.EQ.CAR)GO TO 20 จะผิดถ้า CAR ไม่มีค่ามาก่อน อย่างไรก็ตามคำสั่งนี้จะถูกต้องถ้าใช้

```
DATA CAR/'CAR'/
```

```
:
```

```
IF(A.EQ.CAR)GO TO 20
```

5.2.3 คำสั่ง CHARACTER

ในภาษาฟอร์แทรน 77 มีคำสั่ง CHARACTER เพิ่มจากภาษาฟอร์แทรนรุ่นเก่า ซึ่งเราใช้ในการกำหนดตัวแปรให้เป็นตัวแปรอักขระเพื่อเก็บข้อมูลอักขระ ความปกตินาของตัวแปรอักขระคือ 1 ไบต์ (1 ไบต์เก็บได้ 1 ตัวอักขระ) นอกจากนี้เราจะระบุความยาวเป็นอย่างอื่น คำสั่ง CHARACTER เป็นคำสั่งไม่ปฏิบัติการ และต้องใส่คำสั่งนี้ก่อนคำสั่งปฏิบัติการใด ๆ

ตัวอย่าง CHARACTER*10 ALPHA,BETA,IOTA

คำสั่งนี้ทำให้ตัวแปรทั้ง 3 เป็นตัวแปรอักขระ ซึ่งแต่ละตัวมีความยาว 10 ไบต์

CHARACTER INIT,FIRST

คำสั่งนี้เป็นการกำหนดให้ตัวแปรทั้ง 2 เป็นตัวแปรอักขระ ซึ่งแต่ละตัวมีความยาว 1 ไบต์

CHARACTER*10 FNAME,LNAME*5,INIT*1,STREET,CITY

คำสั่งนี้เป็นการกำหนดให้ตัวแปรทั้ง 5 ตัวเป็นตัวแปรอักขระ โดยที่

FNAME, STREET, CITY แต่ละตัวมีความยาว 10 ไบต์

LNAME มีความยาว 5 ไบต์

ส่วน INIT มีความยาว 1 ไบต์

5.2.3.1 การกำหนดข้อมูลเริ่มต้นแก่ตัวแปรในคำสั่ง CHARACTER

ตัวอย่าง CHARACTER A*4/'HATS',L/'M',ST*3/'DO',NO/'TR'/

หมายความว่า A มีความยาว 4 ไบต์ ดังนั้น A เก็บ

H	A	T	S
---	---	---	---

L ไม่ระบุความยาวดังนั้นจะยาว 1 ไบต์ L เก็บ

M

ST มีความยาว 3 ไบต์ ดังนั้น ST เก็บ

D	O	
---	---	--

NO มีความยาว 1 ไบต์ ดังนั้น NO เก็บ

T

โปรดสังเกตว่าในการใช้คำสั่ง CHARACTER เพื่อกำหนดข้อมูลเริ่มต้นแก่ตัวแปร

ตัวอักขระ 1 ตัวทางซ้ายเท่านั้นที่จะถูกเก็บในตัวแปร คู่สิ่งที่เก็บใน NO

ความยาวสูงสุดที่อาจกำหนดให้แก่ตัวแปรอักขระนั้นขึ้นอยู่กับระบบคอมพิวเตอร์ เช่น

VAX-11/750 นั้นความยาวสูงสุดของตัวแปรอักขระคือ $l=2000$

ตัวอย่าง CHARACTER*4 A/'HATS'/, B*5/'DON''T'/, X/'MAN'/'

หมายความว่า A มีความยาว 4 ไบต์และเก็บ

H	A	T	S
---	---	---	---

B มีความยาว 5 ไบต์และเก็บ

D	O	N	'	T
---	---	---	---	---

X มีความยาว 4 ไบต์และเก็บ

M	A	N	
---	---	---	--

คำสั่งนี้ทำให้ตัวแปรทุกตัวยกเว้น B มีความยาว (l) = 4 ไบต์

ในคำว่า DON'T หรือในสายวลีอักขระใด ๆ ที่มีตัวอะโพสโทรฟี่ (') อยู่ เมื่อเขียนเป็นค่าคงที่อักขระ (literal constant) ให้ใส่เครื่องหมายอะโพสโทรฟี่เพิ่มอีก 1 ตัว เพื่อบอกคอมพิวเตอร์ว่าเครื่องหมายอะโพสโทรฟี่เป็นส่วนหนึ่งของสายวลีอักขระ ไม่ใช่ตัวบอกขอบเขตของสายวลีอักขระ

5.2.3.2 คำสั่งกำหนดค่าอักขระ (Character assignment statement)

ตัวแปรที่ถูกกำหนดให้เป็นตัวแปรอักขระแล้ว เราอาจกำหนดค่าที่เป็นค่าคงที่อักขระให้มัน โดยการใส่คำสั่งกำหนดค่า เช่น

```
CHARACTER BOX*4, M, FIRST*2
```

```
BOX='GO'
```

```
M='TOP'
```

```
FIRST='NO'
```

ในหน่วยความจำหลัก BOX เก็บ

G	O		
---	---	--	--

M เก็บ

T

FIRST เก็บ

N	O
---	---

5.2.3.3 ตัวแปรอักขระกับ A-format code

```
CHARACTER ME*4, SEX, NAME*8, LOAN*4
```

```
READ(5, 10) NAME, SEX, ME, LOAN
```

```
10 FORMAT( A8, A1, 2X, A3, A6)
```

บัตรข้อมูล

MALINEE	^	F		12		PSC		^	^	BUS.
---------	---	---	--	----	--	-----	--	---	---	------

ในหน่วยความจำหลัก NAME เก็บ

M	A	L	I	N	E	E
---	---	---	---	---	---	---

 SEX เก็บ

F

 ME เก็บ

P	S	C
---	---	---

 LOAN เก็บ

B	U	S	.
---	---	---	---

 *

5.2.3.4 การเปรียบเทียบตัวแปรอักขระ

ตัวดำเนินการเปรียบเทียบ(relational operator) 6 ตัวคือ

ตัวดำเนินการ	ความหมาย
.EQ.	=
.NE.	≠
.GT.	>
.GE.	≥
.LT.	<
.LE.	≤

เราอาจเปรียบเทียบตัวแปรอักขระ 2 ตัวโดยใช้ตัวดำเนินการเปรียบเทียบข้างต้น
 นิพจน์การเปรียบเทียบดังกล่าวนี้อาจใช้ในคำสั่ง IF ได้ การเปรียบเทียบจะเปรียบเทียบทีละ
 ตัวอักขระ ถ้าสายวลีอักขระทั้ง 2 มีความยาวไม่เท่ากัน คอมพิวเตอร์จะเติม (pad) ช่องว่าง
 ให้ตัวสั้นกว่าเพื่อทำให้มีความยาวเท่ากันก่อนทำการเปรียบเทียบ การเปรียบเทียบจะหยุดเมื่อ
 พบตัวอักขระที่ไม่เหมือนกัน เราไม่เปรียบเทียบตัวแปรแบบ real และ integer กับตัวแปรอัก-
 ขระ เพราะจะทำให้เกิดข้อผิดพลาดขึ้น

ตัวอย่าง คำสั่งที่ใช้ได้

CHARACTER FNAME*8, JOB*6

READ(5, 2) FNAME, JOB

2 FORMAT(A8, 2X, A6)

IF(FNAME.EQ. JOB)GO TO 5

IF(FNAME.GT.'H' .AND. FNAME.LT.'K')WRITE(6, 20) FNAME

:

ตัวอย่าง คำสั่งที่ใช้ไม่ได้

DATA A/'FIND'/

A เป็นตัวแปรชนิด real จะเก็บค่าคงที่อักขระไม่ได้
นอกจากถ้ากำหนด A ให้เป็นตัวแปรอักขระก่อนโดยใช้
คำสั่ง CHARACTER*4 คำสั่ง DATA นี้จึงจะใช้ได้

IF(B.EQ.'FIND')GO TO 7

ใช้ไม่ได้เพราะเปรียบเทียบตัวแปรแบบ real กับค่าคงที่
อักขระ

5.2.3.5 การเชื่อมต่อข้อมูลอักขระ

1. การเชื่อมต่อ (Concatenation) สายวลีอักขระนั้นทำได้โดยการไว้

ตัวดำเนินการ //

ตัวอย่าง 'COM'//'PUTER' จะได้สายวลีอักขระใหม่คือ 'COMPUTER'

ตัวอย่าง CHARACTER*6 MODEL

DATA MODEL/'PRIME'/

ใน MODEL เก็บ

P	R	I	M	E	
---	---	---	---	---	--

ดังนั้น MODEL//'COM'//'PUTER' จะเท่ากับ 'PRIME COMPUTER'

2. การดึงสายวลย่อย (substring) ออกมาจากค่าคงที่อักขระ

ตัวอย่าง จาก 'COMPUTER' เราต้องการ 'PUT' เราทำได้ดังนี้

CHARACTER ITEM*8, SUB*3

ITEM='COMPUTER'

SUB=ITEM(4:6) ระบุตัวอักษรตั้งแต่ตัวที่ 4-6

ดังนั้น SUB จะเก็บ

P	U	T
---	---	---

ตัวอย่าง CHARACTER*15 COURSE

COURSE='MATHEMATICS'

ดังนั้น COURSE เก็บ 'MATHEMATICS_{AAAA}'

ถ้าต้องการ 'MATH' เราใช้ COURSE(:4)

ส่วนค่าของ COURSE(8:) คือ 'TICS_{AAAA}'

ถ้า N=3 แล้ว COURSE(N:N+3) คือ 'THEM'

ตัวอย่าง CHARACTER*5 STRA, STRB*14, TRUN, PAD

STRA='ALPHA'

STRB=STRA//'^ PARTICLE'

ใน STRA จะเก็บ 'ALPHA'

ใน STRB จะเก็บ 'ALPHA PARTICLE'

ถ้าความยาวของตัวแปรอักขระยาวไม่พอที่จะเก็บสายวลอักขระที่ต้องการเก็บ ตัวอักขระบางส่วนจะถูกตัดทิ้ง และถ้าสายวลอักขระสั้นกว่าความยาวของตัวแปรอักขระ ในการเก็บจะมีการเติม (pad) ด้วยช่องว่าง (blank)

ตัวอย่าง PAD='PAY' จะทำให้ PAD เก็บค่า 'PAY_{AA}' ถ้าเรากำหนด PAD ให้ยาว

5 ไบต์

ถ้า TRUN มีความยาว 5 ไบต์

คำสั่ง TRUN='TEMPERATURE' จะทำให้ TRUN เก็บค่า 'TEMPE'

5.2.4 ค่าคงที่จำนวนจริงในรูปเลขชี้กำลังและ E-format code

ค่าคงที่จำนวนจริงในภาษาฟอร์แทรนเราอาจเขียนได้ 2 รูปคือ

1. รูปทั่วไป คือเลขจำนวนเต็มตัวเลขเขียนเรียงกันและมีจุดทศนิยมนั่นเอง

เช่น -99.99, 28.65

2. รูปเลขชี้กำลังคืออยู่ในรูป

เลขจำนวนจริง E เลขชี้กำลัง

โดยที่ เลขจำนวนจริงนั้นอยู่ในรูปทั่วไป และเลขชี้กำลังต้องเป็นเลขจำนวนเต็มบวกหรือลบก็ได้

จำนวนตัวเลขสำคัญของเลขจำนวนจริงและเลขชี้กำลังนั้นขึ้นอยู่กับระบบคอมพิวเตอร์

ตัวอย่าง	เลขจำนวนจริงในรูปเลขชี้กำลัง	ค่า
	6.2E+4	$6.2 \times 10^4 = 62000.$
	-4.32E14	$-4.32 \times 10^{14} = -4320000000000000.$
	.034E-2	$.034 \times 10^{-2} = .00034$
	-1.2E-7	$-1.2 \times 10^{-7} = -.00000012$

เลขจำนวนจริงในรูปเลขชี้กำลังนั้นมักจะใช้เมื่อเลขจำนวนจริงในรูปทั่วไปนั้นจะมีขนาดใหญ่หรือเล็กมาก เราใช้เลขจำนวนจริงในรูปเลขชี้กำลังในนิพจน์เลขคณิตได้

ตัวอย่าง $Y = 16.2E-4 * Z + W$

หรือ $Y = .00162 * Z + W$

$X = -.01E3 * Y$

หรือ $X = -10. * Y$

$Z = 4.2E+20 ** 2 * Z * .5E-21$

(การนี้เราไม่เขียนเลขจำนวนจริงในรูปทั่วไป)

E-format code

รูปทั่วไป

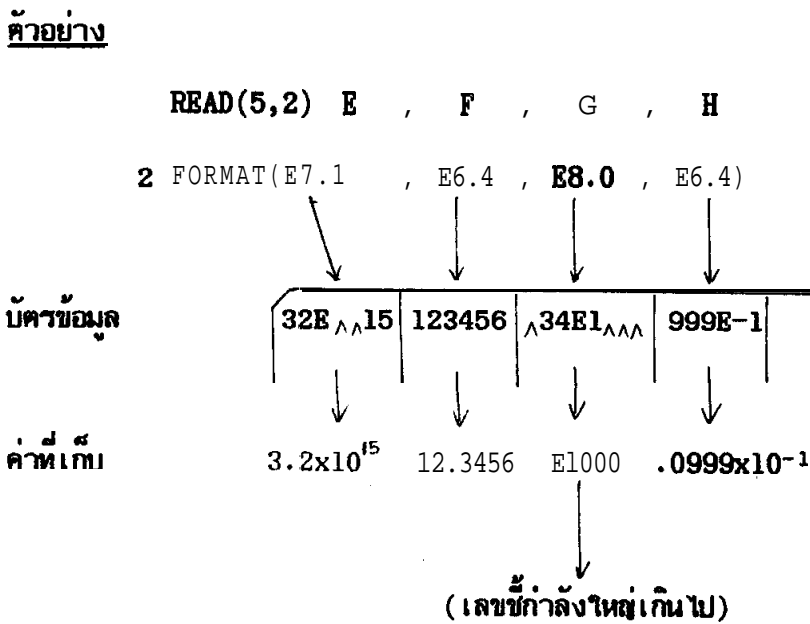
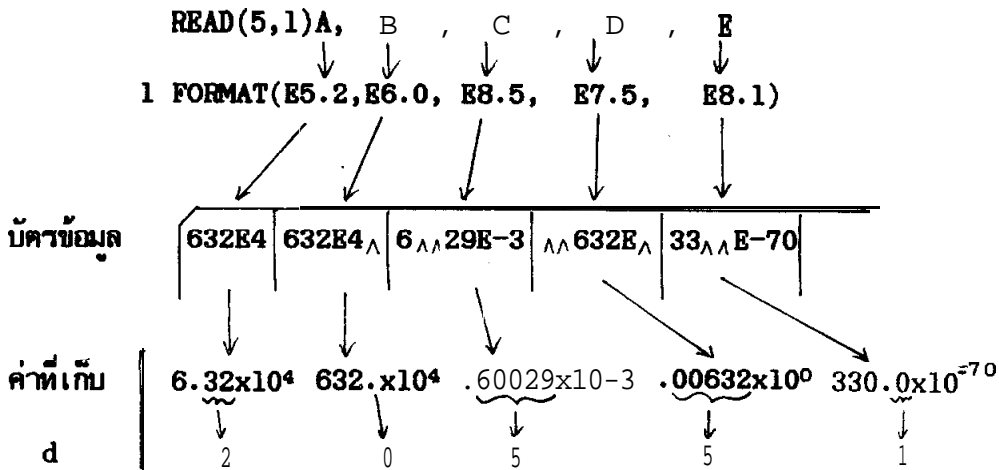
Ew.d

ในการกำหนดข้อมูลเข้า

w = จำนวนสทน์หรือความยาวของฟิลด์

d ใช้ในภาษาซีที่ส่วนเลขนจำนวนจริงเต็ม ไม่มีจุดทศนิยม

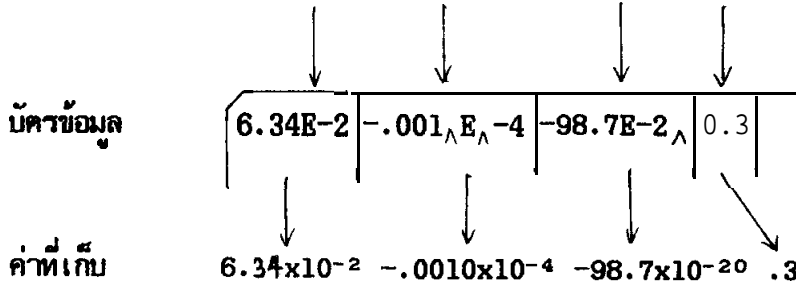
ใช้ d เพื่อกำหนดตำแหน่งของจุดทศนิยมในเลขจำนวนเต็ม ๆ เช่น $-1234E+30$ แล้วจึงจะปรับตำแหน่งทศนิยมตามเลขชี้กำลังอีกทีหนึ่ง แต่ถ้าในส่วนเลขนจำนวนจริงมีจุดทศนิยมอยู่แล้ว d จะไม่มีผลกระทบใด ๆ ต่อค่าที่จะอ่านเข้าไปเก็บ แต่จะต้องใส่ d ไว้ใน format code ด้วย



ตัวอย่าง

```
READ(5,3) X, Y, Z, W
```

```
3 FORMAT(E7.0, B10.4, E9.1, E3.3)
```

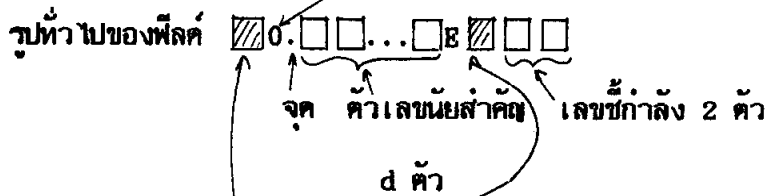


บัตรข้อมูล

ค่าที่เก็บ

เพื่อหลีกเลี่ยงความผิดพลาดที่อาจเกิดขึ้น โปรแกรมเมอร์ควาบันทึกเลขจำนวนชนิดวิชาของฟิลด์ ทั้งจำนวนทั้งส่วนที่เป็นเลขชี้กำลังด้วย

ในการนำข้อมูลออก เลขศูนย์ตัวนำ



เครื่องหมาย (ถ้าเลขจำนวนเป็นบวกจะไม่แสดงเครื่องหมายบวก (+))

แต่ถ้าเลขจำนวนเป็นลบจะแสดงเครื่องหมายลบ (-)

คั้งจำนวนค่าสุดของ p คือ 7 ($p \geq 7$) นั่นคือถ้าเลขจำนวนเป็นจำนวนลบไม่มีตัวเลขนัยสำคัญ และ เลขชี้กำลังเป็นลบ

```
p=7 ----> -0.E-□□
```

(ในบางระบบ ถ้า p ยาวไม่พอ การพิมพ์อาจไม่แสดงเลขศูนย์ที่เป็นตัวนำ)

ตัวอย่าง

ค่าที่เก็บในหน่วยความจำหลัก	format code	ค่าที่พิมพ์ให้	หมายเหตุ
632000.	E10.3	$^{\wedge}0.632E_{\wedge}06$	มีเลข 0 นำ
632000.	E 8.3	.632E $_{\wedge}06$	ไม่มีเลข 0 นำ
832999.	E 9.3	0.633E $_{\wedge}06$	มีการปัดเศษ

ค่าที่เก็บในหน่วยความจำหลัก	format code	ค่าที่พิมพ์ให้	หมายเหตุ
-.83247	312.5	-0.83247E _Λ 00	มีเลข 0 นำและ
-.83247	E13.4	M-0.8325&00	มีการปัดเศษ
-.000004269	E10.3	-0.427E-05	มีการปัดเศษ
-.000004269	E10.1	ΛΛ-0.4E-05	มีเลข 0 นำ
-98.5678	E8.1	-0.1E _Λ 03	มีการปัดจุด
-98.5678	38.0	Λ-0.E _Λ 03	ไม่มีตัวเลขนัยสำคัญ
123.456721	B15.9	.123456721E _Λ 03	ถ้าเครื่องที่เก็บตัวเลข นัยสำคัญได้ 7 ตัว ตัวเลข 21 จะไม่มี
3.256	B8.4	*****	ตำแหน่งการพิมพ์ไม่พอ

5.2.5 เลขจำนวนจริงชนิด double precision และ D-format code

สำหรับการประยุกต์ใช้งานบางอย่างจำนวนตัวเลขนัยสำคัญที่ใช้ได้กับเลขจำนวนจริงนั้น อาจจะไม่เพียงพอ คอมพิวเตอร์ส่วนมากได้ขยายจำนวนตัวเลขนัยสำคัญออกไปอีก ในตารางต่อไปนี้จะแสดงจำนวนตัวเลขนัยสำคัญที่ใช้ได้กับคอมพิวเตอร์บางระบบ

คอมพิวเตอร์	จำนวนบิตใน 2 ไบต์	พิสัยของเลข- ชี้กำลัง	จำนวนตัวเลขนัยสำคัญ ในฐานะสิบ
IBM 1130	48	10 ³⁸ -10 ⁻³⁹	10
IBM 370/30XX/43XX	64	10 ⁷⁵ -10 ⁻⁷⁸	16
Burroughs B6700	96	10 ²⁹⁶⁰³ -10 ⁻²⁹⁵⁸¹	24
CDC Cyber 72	120	10 ³⁰⁸ -10 ⁻³⁰⁸	33

วิธีการกำหนดค่าแปรให้มีเนื้อที่ขนาด 2 เวิร์ด (double word) โดยการใช้คำสั่ง

DOUBLE PRECISION ซึ่งเป็นคำสั่งไม่ปฏิบัติการ มีรูปทั่วไปดังนี้

DOUBLE PRECISION รายชื่อค่าแปร

คำสั่งนี้จะทำให้ค่าแปรทั้งหมดในรายชื่อค่าแปรใช้เก็บเลขจำนวนจริงแบบ double precision

ตัวอย่าง DOUBLE PRECISION X, I

$$X = 123456769012345.$$

$$I = .123456789012345D40$$

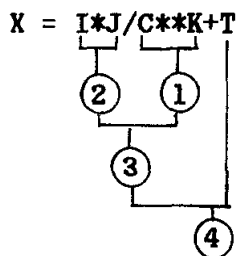
คำสั่งที่ double precision นั้นเขียนในรูปทั่วไป ซึ่งวิธีการเขียนทำนองเดียวกับรูปทั่วไปของเลขจำนวนจริง เพียงแต่มีจำนวนตัวเลขน้อยสำคัญมากกว่า นอกจากนี้ยังอาจเขียนในรูปเลขชี้กำลังได้โดยการใช้ตัวอักษร D การใช้ตัว D นี้มีกฎเกณฑ์ทำนองเดียวกับการใช้ E

ตัวอย่าง $6.2D+4$ มีค่าเท่ากับ $6.2 \times 10^4 = 62000.$

$$-.0326798432156D -4 \text{ มีค่าเท่ากับ } -.0326798432156 \times 10^{-4} \\ = -.00000326798432156$$

เราอาจใช้คำสั่งที่ double precision ผสมกับคำสั่งที่และค่าแปรชนิด real และ integer ในโปรแกรมคอมพิวเตอร์ได้ เมื่อตัวถูกกระทำทางคณิตศาสตร์มีทั้งชนิด double precision และชนิดอื่น ก่อนการคำนวณ ตัวถูกกระทำที่ไม่ใช่ชนิด double precision จะถูกแปลงเป็นชนิด double precision ก่อนแล้วจึงจะทำการคำนวณ ผลลัพธ์ของการคำนวณจะเป็นคำสั่งที่ double precision

ตัวอย่าง DOUBLE PRECISION T, X



C**K คำนวณในแบบ real

I*J คำนวณในแบบ integer

(I*J)/(C**K) คำนวณในแบบ real

(I*J)/(C**K)+T คำนวณในแบบ double precision แล้วผลลัพธ์จากการคำนวณซึ่งเป็นชนิด double precision จะถูกเก็บในตัวแปรชนิด double precision ชื่อ X

ถ้าแทนที่จะใช้ตัวแปรชนิด double precision เป็นตัวแปรทางคำนวณซ้ายของเครื่องหมายเท่ากับในคำสั่งกำหนดค่า เราใช้ตัวแปรชนิด real จะเห็นว่าผลลัพธ์ที่เป็นค่าคงที่ double precision เมื่อเก็บในตัวแปรชนิด real ตัวเลขท้ายสำคัญบางส่วนจะหายไป จากตัวอย่างต่อไปนี้ สมมติว่าตัวแปรชนิด real นั้นเก็บตัวเลขท้ายสำคัญได้ 7 ตัว

DOUBLE PRECISION X,Y	<u>ผลลัพธ์</u>
REAL A,B	
X=1.23456789D3	X เก็บค่า 1234.56789
A=X	A เก็บค่า 1234.567
Y=999999999	Y เก็บค่า 999999999.
B=Y	B เก็บค่า 999999900.
I=1.29456789D1	I เก็บค่า 12 (ไม่มีทศนิยม)
J=Y	J เก็บค่า 999999999 (ตัวแปรชนิด integer เก็บเลขฐานสิบได้ ใหญ่ถึง 10 หลัก)

Dw.d นั้นใช้ได้ทั้งในการนำข้อมูลเข้าและออก รายละเอียดการใช้งานนั้นเช่นเดียวกับ Ew.d ข้อมูลชนิด double precision นั้นใช้หน่วยความจำหลัก (จำนวนบิต) มากกว่าข้อมูลชนิด real และการคำนวณกับข้อมูลชนิดนี้ใช้เวลามากกว่าการคำนวณอย่างเดียวกันกับข้อมูลชนิด real

5.2.6 ข้อมูลตรรกะและ L-format code

เท่าที่ผ่านมาเราได้ใช้นิพจน์ตรรกะมาแล้วบ้างในคำสั่ง IF นิพจน์ตรรกะมีค่าเป็นจริง (true) หรือเท็จ (false) เราอาจให้นิยามนิพจน์ตรรกะในภาษาฟอร์แทรนว่าอาจเป็น

1. นิพจน์เลขคณิต 2 นิพจน์ซึ่งเชื่อมด้วยตัวดำเนินการเปรียบเทียบ (relational operator เช่น .GT., .GE., .LT. เป็นต้น)

2. นิพจน์ที่ประกอบด้วยตัวดำเนินการตรรกะ (logical operator คือ .AND., .OR., .NOT.)

ตัวอย่าง นิพจน์ตรรกะชนิดที่ 1

ถูกต้อง	ไม่ถูกต้อง
X.LE.3	X+23*Z ค่าของนิพจน์ไม่เป็นจริงหรือเท็จ
XAM.NE.Y	.LE.100 ขาดนิพจน์เลขคณิตหน้า .LE.
(X+Y)**2.EQ.4	(X.LT.Y)+Z นิพจน์ตรรกะจะใช้เป็นคำถูกกระทำ
2.*Y.GT.X-Y	
-1..LE.SQRT(Z)	ของตัวดำเนินการเลขคณิตไม่ได้

5.2.6.1 คำสั่ง LOGICAL

นิพจน์ตรรกะนั้นใช้ในคำสั่งกำหนดค่าก็ได้ โดยเราต้องกำหนดตัวแปรที่จะเก็บค่าของนิพจน์ตรรกะให้เป็นตัวแปรตรรกะ (logical variable) โดยใช้คำสั่ง LOGICAL ซึ่งมีรูปแบบทั่วไปคือ

LOGICAL รายชื่อตัวแปร

คำสั่งนี้จะทำให้ตัวแปรทุกตัวในรายชื่อตัวแปรเป็นตัวแปรตรรกะ ซึ่งจะเก็บค่าคงที่ตรรกะคือ .TRUE. และ .FALSE. เท่านั้น

ตัวอย่าง LOGICAL X,I,A

X, I และ A จะเป็นตัวแปรตรรกะ ซึ่งจะถูกกำหนดค่าเป็น .TRUE. หรือ .FALSE. โดยใช้คำสั่งกำหนดค่าตรรกะได้ (ในกรณีนี้จะกำหนดค่าที่เป็นเลขจำนวนในขั้นไม่ได้)

5.2.6.2 คำสั่งกำหนดค่าตรรกะ (Logical assignment statement)

รูปทั่วไปของคำสั่งกำหนดค่าตรรกะคือ

ตัวแปรตรรกะ = นิพจน์ตรรกะ

ตัวอย่าง นิพจน์ตรรกะที่ใช้ได้

LOGICAL A,B,C	กำหนดค่าให้ A, B และ C เป็นตัวแปรตรรกะ
x=4	} คำสั่งกำหนดค่าเลขคณิต
Y=-2	
A=.TRUE.	A มีค่าจริง
C=A	C มีค่าจริง
B=X.LT.Y	B มีค่าเท็จ เพราะ $4 \not< -2$
C=Y.NE.SQRT(X)	C มีค่าจริง เพราะ $-2 \neq 2$
B=(X+Y)**2.EQ.4	B มีค่าจริง เพราะ $4=4$
C=(2.*Y).GT.(X-Y)	C มีค่าเท็จ เพราะ $-4 \not> 6$

ตัวอย่าง นิพจน์ตรรกะที่ใช้ไม่ได้

LOGICAL A,B,C	
A=2.1*X	ไม่มีตัวดำเนินการตรรกะ
B=2.*C	C เป็นตัวแปรตรรกะคูณกับ 2. ไม่ได้
C=A+B	เราไม่บวกตัวแปรตรรกะ 2 ตัว
A=B.EQ..TRUE.	เราไม่เปรียบเทียบค่าตรรกะ 2 ค่า
C=A.LT.X	A ไม่เป็นนิพจน์เลขคณิต
Z=.FALSE.	Z ไม่ใช่ตัวแปรตรรกะ
C=(X.LT.Y)+(Z.GT.3)	เราไม่บวกค่าตรรกะ 2 ค่า

นิพจน์ตรรกะชนิดที่ 2 คือนิพจน์ที่ใช้ตัวดำเนินการตรรกะ .AND., .OR. และ .NOT.

ตัวถูกกระทำในนิพจน์ตรรกะอาจเป็น ตัวแปรตรรกะ ค่าคงที่ตรรกะหรือนิพจน์ตรรกะชนิดที่ 1

ถ้า L_1 และ L_2 เป็นตัวถูกกระทำตรรกะ ผลการใช้ตัวดำเนินการตรรกะทั้ง 3 เป็นดังนี้

L_1 .AND. L_2 จะมีค่า .TRUE. ถ้าทั้ง L_1 และ L_2 มีค่าเป็น .TRUE. เท่านั้น

L_1 .OR. L_2 จะมีค่า .TRUE. ถ้า L_1 หรือ L_2 หรือทั้ง 2 ตัวมีค่าเป็น .TRUE.

.NOT. L_2 จะมีค่า .TRUE. ถ้า L_2 เป็น .FALSE.

นอกเหนือไปจากที่กล่าวข้างต้น นิพจน์จะมีค่าเป็น .FALSE.

เราอาจแสดงผลการกระทำดังกล่าวข้างต้น ในรูปตารางได้ดังนี้

					FORTRAN 77	
L ₁	L ₂	L ₁ .AND. L ₂	L ₁ .OR. L ₂	.NOT. L ₁	L ₁ { .EQV. } L ₂ { .XOR. }	L ₁ .NEQV. L ₂
T	T	T	T	F	T	F
T	F	F	T	F	F	T
F	T	F	T	T	F	T
F	F	F	F	T	T	F

T = .TRUE. และ F = .FALSE.

ตัวอย่าง ค่าของนิพจน์ตรรกะเมื่อ x=4, Y=2 และ Z=2

L ₁	ตัวดำเนินการ	L ₂	ค่าของ L ₁	ค่าของ L ₂	ค่าของ นิพจน์ตรรกะ
X.GT.SQRT(Y)	.AND.	Y.EQ.Z	T	T	T
X.LT.Z	.OR.	Y.NE.Z	F	F	F
Y.LE.X	.OR.	Y+Z.LT.X	T	F	T
X.GT.5.1	.AND.	X.EQ.2*Z	F	T	F
	.NOT.	X.LT.Y+Z		F	T

ตัวอย่าง คำสั่งกำหนดค่าตรรกะ (Logical assignment statement)

LOGICAL A,B,C,D,E,F

x=3.

Y=-2.3

ค่าของตัวแปรตรรกะ

$$A = X \cdot LT \cdot Y$$

$$B = Y \cdot LE \cdot 20$$

$$C = A \cdot \text{AND} \cdot B$$

$$A = \cdot \text{NOT} \cdot (X \cdot LT \cdot Y)$$

$$D = A \cdot \text{OR} \cdot X \cdot LT \cdot 6$$

$$E = \cdot \text{NOT} \cdot C$$

$$F = A \cdot \text{AND} \cdot \cdot \text{NOT} \cdot B$$

$$E = (\cdot \text{TRUE} \cdot \cdot \text{OR} \cdot \text{FALSE} \cdot) \cdot \text{AND} \cdot \cdot \text{TRUE} \cdot$$

A มีค่าเท็จ เพราะ $3 \not< -2.3$

B มีค่าจริง เพราะ $-2.3 < 20$

C มีค่าเท็จ เพราะ A เป็นเท็จ

A มีค่าจริง เพราะ $X \cdot LT \cdot Y$ เป็นเท็จ

D มีค่าจริง เพราะ $X \cdot LT \cdot 6$ เป็นจริง

E มีค่าจริง เพราะ C เป็นเท็จ

F มีค่าเท็จ เพราะทั้ง A และ $\cdot \text{NOT} \cdot B$ เป็นเท็จ

E มีค่าจริง

ตัวอย่าง คำสั่งที่ไม่ถูกต้อง

LOGICAL A, B, C

$$A = \cdot \text{NOT} \cdot (X)$$

$$B = (X+1) \cdot \text{OR} \cdot (Y+6)$$

$$C = X \cdot LT \cdot Y + Z \cdot \text{GT} \cdot 3$$

$$A = B \cdot \text{OR} \cdot (C+1)$$

X ไม่ใช่ตัวแปรตรรกะ

X+1 และ Y+6 ไม่ใช่นิพจน์ตรรกะ

ไม่มีตัวดำเนินการตรรกะ บางใช้ไม่ได้

C+1 ไม่ใช่นิพจน์ตรรกะ

ถ้านิพจน์ตรรกะมีตัวแปรตรรกะมากกว่า 1 ตัว เราอาจใช้วงเล็บเพื่อกำหนดว่าเราต้องการคำนวณนิพจน์ใดก่อน พิจารณานิพจน์ $L_1 \cdot \text{AND} \cdot L_2 \cdot \text{OR} \cdot L_3$ ซึ่งถ้า L_1 เป็นเท็จ, L_2 และ L_3 เป็นจริง ถ้าเราใช้เครื่องหมายวงเล็บต่างกัน ค่าของนิพจน์ตรรกะจะต่างกันด้วย

$$L_1 \cdot \text{AND} \cdot (L_2 \cdot \text{OR} \cdot L_3) \quad \text{มีค่าเท็จ}$$

$$(L_1 \cdot \text{AND} \cdot L_2) \cdot \text{OR} \cdot L_3 \quad \text{มีค่าจริง}$$

$$L_1 \cdot \text{AND} \cdot L_2 \cdot \text{OR} \cdot L_3 \quad \text{ก็มีค่าจริงเช่นเดียวกัน}$$

ในกรณีที่เราไม่ใช่เครื่องหมายวงเล็บในนิพจน์ตรรกะ เราควรทราบลำดับชั้นในการกระทำตามตัวดำเนินการต่าง ๆ ซึ่งเป็นไปตามลำดับก่อนหลังตามตารางต่อไปนี้

การกระทำ	หมายเหตุ	ลำดับ
การกระทำทางคณิตศาสตร์ การกระทำการเปรียบเทียบ (.LT., .GE., เป็นต้น) .NOT. .AND. .OR. .EQV. (หรือ .XOR.), .NEQV	เป็นไปตามกฎการคำนวณนิพจน์เลขคณิต ทำจากซ้ายไปขวา กระทำกับนิพจน์ทางขวาที่ติดกับ .NOT. (ใน FORTRAN 77 เท่านั้น)	สูงสุด ↓ ต่ำสุด สุดท้าย (ถ้ามี)

ถ้าเราต้องการเปลี่ยนลำดับการกระทำตามตารางข้างบน เราใช้เครื่องหมายวงเล็บ

เข้าช่วยได้เสมอ

ตัวอย่าง ถ้า $X=30$, $Y=40$ และ $A=.TRUE.$

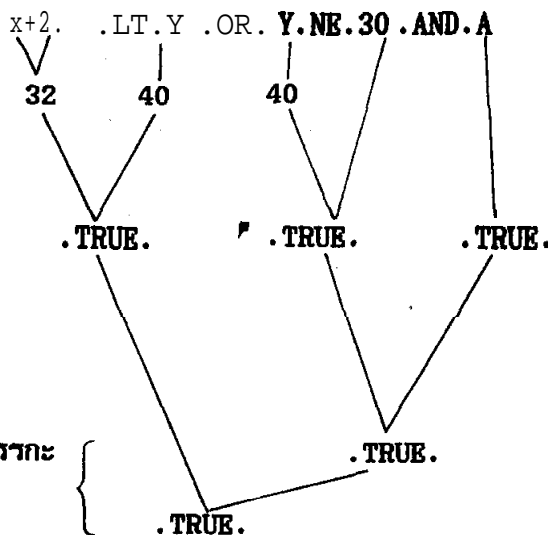
นิพจน์ตรรกะ

หาค่านิพจน์เลขคณิต

หาค่าตามตัวดำเนินการ

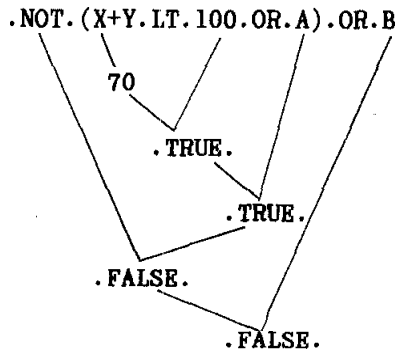
เปรียบเทียบ

หาค่าตามตัวดำเนินการตรรกะ



การคำนวณในนิพจน์ข้างต้นคือ $((X+2).LT.Y).OR.((Y.NE.30).AND.A)$

ตัวอย่าง ถ้า X=30, Y=40, และ A=true และ B=false



5.2.6.3 นิพจน์ตรรกะในคำสั่ง IF ตรรกะ

ในคำสั่ง IF ตรรกะนั้นใช้ นิพจน์ตรรกะในรูปดังนี้

IF(นิพจน์ตรรกะ)คำสั่งปฏิบัติการ

ซึ่งถ้าค่าของนิพจน์ตรรกะเป็นจริง คำสั่งปฏิบัติการในคำสั่ง IF จะถูกทำ (execute) แต่ถ้าค่าของนิพจน์ตรรกะเป็นเท็จ คำสั่งถัดจากคำสั่ง IF จะถูกทำ

ตัวอย่าง ถ้า X=30, Y=40 และ B=.TRUE.

LOGICAL A,B,C

IF(B)GO TO 30

คำสั่งที่ 30 จะถูกทำต่อไปเพราะ B เป็นจริง

:

A=X.LT.Y

A เป็นจริง

IF(A.AND.B)C=.TRUE.

(A.AND.B) เป็นจริง ดังนั้นคำสั่ง C=.TRUE.

:

จะถูกทำ

IF(X.GT.100.OR..NOT.B)GO TO 7

(X.GT.100) เป็นเท็จ .NOT.B เป็นเท็จ

:

เช่นกัน ดังนั้นคำสั่งถัดจากคำสั่ง IF จะถูกทำ

5.2.6.4 L-format code

รูปทั่วไป

Lw

โดยที่ w คือความกว้างของฟิลด์

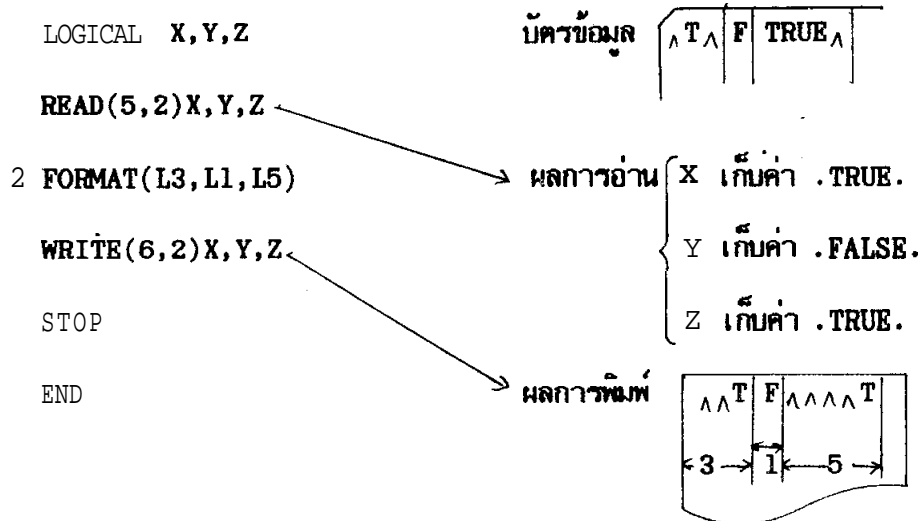
เราใช้ตัวแปรตรรกะในคำสั่งนำข้อมูลเข้า/ออกได้

ในการกำหนดค่าให้ตัวแปรวาระ โดยการใช้คำสั่ง READ หรือแสดงค่าของมันโดยใช้คำสั่ง WRITE เราต้องใช้ L-format code

ในการนำข้อมูลเข้า เมื่ออ่านค่าของตัวแปรวาระ การอ่านจะเริ่มอ่านจากซ้ายไปขวาของฟิลด์ ถ้าตัวอักษรตัวแรกที่ไม่ใช่ช่องว่างตัวแรกคือ T ค่าของตัวแปรวาระนั้นจะเป็น .TRUE. แต่ถ้าตัวอักษรตัวแรกคือ F หรือทั้งฟิลด์เป็นช่องว่างค่าของตัวแปรวาระนั้นจะเป็น .FALSE. อักษรอื่น ๆ จะไม่กระทบต่อค่าของตัวแปรวาระ

ในการนำข้อมูลออก T หรือ F เท่านั้นจะถูกพิมพ์ ทั้งนี้เมื่อค่าเป็น .TRUE. หรือ .FALSE. ตามลำดับ ตัวอักษร T หรือ F จะถูกพิมพ์ไว้คี่ขวาของฟิลด์โดยที่ (w-1) ที่ทางซ้ายจะเป็นช่องว่าง

ตัวอย่าง



เราใช้คำสั่ง DATA กำหนดค่าให้ตัวแปรวาระ ได้ดังตัวอย่างข้างล่าง

ตัวอย่าง LOGICAL X,Y

DATA X/.TRUE./,Y/.FALSE./

5.2.7 ข้อมูลเชิงซ้อน (Complex data)

เลขจำนวนเชิงซ้อน (complex number) คือเลขจำนวนที่อยู่ในรูป

$$a+bi$$

โดยที่ $i = \sqrt{-1}$ และ a, b เป็นเลขจำนวนจริง โดยที่ a เป็นส่วนจริง (real part) และ bi เป็นส่วนจินตภาพ (imaginary part) ตัวอย่างของเลขจำนวนเชิงซ้อนคือ

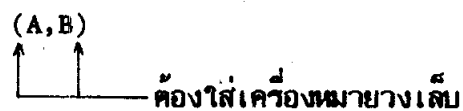
$$4+3i$$

$$3.2+(-4)i$$

$$-.6+70i$$

$$3.-6i$$

เลขจำนวนเชิงซ้อนในภาษาฟอร์แทรนอยู่ในรูป



โดยที่ A, B เป็นเลขจำนวนจริง A เป็นส่วนจริงและ B เป็นส่วนจินตภาพ

ตัวอย่าง	เลขจำนวนเชิงซ้อนในภาษาฟอร์แทรน	ค่าทางคณิตศาสตร์
	(4.83.)	$4+3i$
	(3.2,4.)	$3.2+4i$
	(24.3E-2,79.)	$.243+79i$
	(1.1E+10,.2E-3)	$11000000000+.0002i$

ตัวอย่าง	เลขจำนวนเชิงซ้อนที่ไม่ถูกต้อง และ เหตุผล
(0.,I)	I ไม่ใช่ เลขจำนวนจริง
1.,1.	ไม่มีเครื่องหมายวงเล็บ
(3115,3.4)	ส่วนจริงเป็นเลขจำนวนเต็ม
(.004E+4,.5D0)	ส่วนทั้ง 2 ต่างชนิดกัน

5.2.7.1 คำสั่ง COMPLEX

ตัวแปรที่จะเก็บเลขจำนวนเชิงซ้อนได้จะต้องถูกกำหนดให้เป็นตัวแปรเชิงซ้อน (Complex variable) ก่อนโดยการใช้คำสั่ง COMPLEX ซึ่งมีรูปแบบทั่วไปดังนี้

COMPLEX รายชื่อตัวแปร

ตัวอย่าง COMPLEX A,B,X(10)

คำสั่งนี้จะประกาศตัวแปร A, B และ X(1), ..., X(10) เป็นตัวแปรเชิงซ้อน

ในที่นี้ X(1), ..., X(10) เป็นสมาชิกของแถวลำดับ (array) X ซึ่งเป็นแถวลำดับแบบ 1 มิติ (one dimensional array) ซึ่งจะได้กล่าวถึงเรื่องแถวลำดับในบทต่อไป

ตัวแปรเชิงซ้อนตัวหนึ่งจะประกอบด้วยตัวที่ 2 ที่ภายในหน่วยความจำหลัก โดยที่ ๆ หนึ่งเก็บส่วนจริง และอีกที่หนึ่งเก็บส่วนจินตภาพ เราอาจเขียนนิพจน์เชิงซ้อนซึ่งจะให้ค่าเป็นเลขจำนวน

เชิงซ้อนได้ทำนองเดียวกับการเขียนนิพจน์เลขคณิต

ตัวอย่าง COMPLEX A,B,C,X,Z

ผลจากคำสั่ง

A=(2.,2.)+(0.,1.)

A=(2.,3.)=2+3i

B=(5.,-1.)*2.

B=(10.,-2.)=10-2i

C=A+B-5.

C=(7.,1.)=7+i

Y = A - B

Y=-8 (Y เป็นตัวแปรชนิด real
ดังนั้นส่วนจินตภาพจึงถูกตัดทิ้ง)

C=A*B

C=(26.,26.)=26+26i

C=(3.,5.)-A

C=(1.,2.)

C=(A*B)/2.*(0.,1.)

C=(-13.,13.)

C=A**2-(2.,3.4)**3

C=(-66.36,64.904)

C=(X,Z)

ใช้ไม่ได้เพราะ X และ Z ไม่ใช่
เลขจำนวนจริง

C=A+2

C=(4.,3.)

5.2.7.2 การคำนวณกับเลขจำนวนเชิงซ้อน

การบวก $(a_1+a_2i)+(b_1+b_2i) = (a_1+b_1)+(a_2+b_2)i$

ตัวอย่าง $(2+2i)+(0+i) = (2+0)+(2+1)i$
 $= 2+3i$

การลบ $(a_1+a_2i)-(b_1+b_2i) = (a_1-b_1)+(a_2-b_2)i$

ตัวอย่าง $(3+5i)-(2+2i) = (3-2)+(5-2)i$
 $= 1+3i$

การคูณ $(a_1+a_2i)*(b_1+b_2i) = (a_1b_1-a_2b_2)+(a_1b_2+a_2b_1)i$

การคูณด้วยค่าคงที่ $c(a_1+a_2i) = ca_1+ca_2i$

ตัวอย่าง $(3+5i)*(2+2i) = (6-10)+(6+10)i = -4+16i$
 $2(5-i) = 10-2i$

การหาร $\frac{a_1+a_2i}{b_1+b_2i} = \frac{a_1+a_2i}{b_1+b_2i} * \frac{b_1-b_2i}{b_1-b_2i}$
 $= \frac{a_1b_1+a_2b_2}{b_1^2+b_2^2} + \left(\frac{a_2b_1-a_1b_2}{b_1^2+b_2^2} \right) i$

b_1-b_2i คือสังยุค (Conjugate) ของ b_1+b_2i

การยกกำลัง 1) เลขจำนวนเชิงซ้อนเป็นเลขชี้กำลังไม่ได้ เช่น

$3.14^{(1.,3.)}$ นั้นใช้ไม่ได้

2) เลขจำนวนเชิงซ้อนยกกำลังด้วยเลขจำนวนจริงไม่ได้ เช่น

ถ้า A เป็นตัวแปรเชิงซ้อน $C=A^{2.0}$ ใช้ไม่ได้

แต่ $C=A^{**2}$ นั้นใช้ได้

5.2.7.3 ฟังก์ชันภายในที่ใช้กับข้อมูลเชิงซ้อน

ฟังก์ชันภายในของภาษาฟอร์แทรนที่ใช้กับข้อมูลเชิงซ้อนมักจะพบในระบบคอมพิวเตอร์

ทั่วไปเช่น

ฟังก์ชัน	ความหมาย
REAL	Real(a,b)=a (a,b เป็นเลขจำนวนจริง)

ฟังก์ชัน	ความหมาย
AIMAG	Imaginary (a,b) = b
CONJG	Conjugate (a,b) = (a,-b)
CABS	Absolute value (a,b) = $\sqrt{a^2 + b^2}$ =modulus (a,b)
CSQRT	Principal square root of a+bi
CLOG	Logarithm
CSIN	} ฟังก์ชันตรีโกณมิติ
CCOS	
CMPLX	ถ้า Z เป็นตัวแปรเชิงซ้อน, a,b เป็นเลขจำนวนจริง Z=CMPLX(a,b) หมายความว่า Z=a+bi

5.2.7.4 การนำข้อมูลเชิงซ้อนเข้า/ออก

ตัวแปรเชิงซ้อนแต่ละตัวต้องใช้ format code 2 ตัว ตัวแรกสำหรับส่วนจริงและตัว-
หลังสำหรับส่วนจินตภาพ

ตัวอย่าง

```

COMPLEX A,B,C
READ(5,1)A
                                     B
1 FORMAT(F4.0, F6.0, F3.2, F4.3)
      ส่วนจริงของ   ส่วนจินตภาพ   ส่วนจริงของ   ส่วนจินตภาพ
      A             ของ A           B             ของ B
C=(6.2,-4.325)
WRITE(6,3)A,B,C
3 FORMAT(2X,6F8.4)

```

5.2.7.5 รากเชิงซ้อนของสมการกำลังสอง (Complex roots Of a quadratic equation)

รากของ $ax^2+bx+c=0$ คือ

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{และ} \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

ถ้า $b^2-4ac \geq 0$ ค่าของ x_1 และ x_2 เป็นเลขจำนวนจริง

ถ้า $b^2-4ac < 0$ ค่าของ x_1 และ x_2 เป็นเลขจำนวนเชิงซ้อน

ในกรณีที่รากเป็นเลขจำนวนจริง เราก็คือว่าส่วนจินตภาพของรากจะเป็น 0 (สมมติว่า $A1 \neq 0$)

COMPLEX X1,X2,A1,B1,C1,D

2 READ(5,1,END=10)A,B,C

1 FORMAT(3F4.0)

A1=A

B1=B

C1=C

D=CSQRT(B1**2-4.*A1*C1)

(ฟังก์ชัน CSQRT ต้องใช้อำนาจิเวเมนต์

X1=(-B1+D)/(2.*A1)

เป็นชนิดข้อมูลเชิงซ้อน)

X2=(-B1-D)/(2.*A1)

WRITE(6,3)A,B,C,X1,X2

3 FORMAT(2X,'A=',F5.0,'B=',F5.0,'C=',F5.0,

*'X1=',2F7.3,'X2=',2F7.3)

GO TO 2

10 STOP

END

แบบฝึกหัดที่ 5

1. จงเขียนเลขจำนวนจริงในรูปเลขชี้กำลัง เป็นเลขจำนวนจริงในรูปทั่วไป

- | | |
|---------------------------|----------------------------|
| 1.1) 3.2E-4 | 1.2) .0034E10 |
| 1.3) -132.4E6 | 1.4) -132.4E-6 |
| 1.5) 432.4D2 | 1.6) -163.948720-10 |
| 1.7) 1632453.11D-8 | 1.8) .0000324D15 |

2. จงบอกค่าที่จะเก็บในหน่วยความจำหลักในรูป 0.ddddddd $\times 10^e$

	ข้อมูลใน input field	READ format code	ค่าที่เก็บ
2.1)	632E4	E5.2	
2.2)	-.623E14	E8.2	
2.3)	1234E-2	E7.0	
2.4)	-12343-5	68.2	
2.5)	69.52D4	D7.2	
2.6)	000003241	E9.3	
2.7)	-00002561E4	B11.4	
2.8)	333.447E-50	E12.0	

3. จงแสดงผลการพิมพ์ค่าตาม format code ต่อไปนี้

	ข้อมูลในหน่วยความจำหลัก	WRITE format code	ผลการพิมพ์
3.1)	.0032456	E8.2	
3.2)	-98.9437	E15.1	
3.3)	.0032456	E11.4	
3.4)	31245.F31	B15.2	
3.5)	-12340000.	B13.4	
3.6)	-12340000.	D16.7	
3.7)	-.0000006972	E8.1	

3.8)	+212.E+26	B7.1
3.9)	212.E26	E6.1
3.10)	123.4567891	E17.10
3.11)	123.4587891	D16.8

4. จงหาค่าของนิพจน์ตรรกะ ถ้า $A=3.0$, $B=-4$. และ $C=0$.

4.1) A.LT.B

4.2) .NOT.A.GT.O.

4.3) B.LT.C.OR.A.LT.B

4.4) B.LE.C.AND.A.LT.B

4.5) C.GT.B.AND.(A.LE.16.O.OR.B.EQ.4)

4.6) .NOT.(A.GT.B.OR.C.EQ.O)

4.7) .NOT.A.GT.B.OR.C.EQ.O

4.8) A.EQ.B.AND.B.LT.C.OR.(.NOT.A.LT.B)

5. จงหาค่าของนิพจน์ตรรกะต่อไปนี้ ถ้า $A=.TRUE.$, $B=.TRUE.$ และ $C=.FALSE.$

5.1) A.OR.B

5.2) .NOT.C

5.3) (A.OR.B).AND.C

5.4) .TRUE..OR.C

5.5) .TRUE..AND.C

5.6) .NOT.C.OR.B

6. จงเขียนคำสั่ง IF เพียง 1 คำสั่งให้ได้ผลเช่นเดียวกับคำสั่งที่กำหนดให้

6.1) IF(X.GT.O)GO TO 20

GO TO 30

20 IF(X.LT.10)STOP

30 . . .

6.2) **IF(X.GT.10)GO** TO 20
 IF(X.LT.10)GO TO 20
 GO TO 30
 20 STOP
 30 . . .

7. จงหาค่าของนิพจน์ตรรกะต่อไปนี้ โดยใช้ค่าของตัวแปรตรรกะ A, B, และ C ที่เป็นไปได้ทั้งหมด นั่นคือถ้านิพจน์มีตัวแปรตรรกะ A และ B ให้หาค่าของมันเมื่อ (A,B) มีค่า (T,T), (T,F), (F,T), (F,F) และถ้านิพจน์มีตัวแปร A, B และ C ให้หาค่าของนิพจน์เมื่อ (A,B,C) มีค่า (T,T,T), (T,T,F), (T,F,T), (T,F,F), (F,T,T), (F,T,F), (F,F,T), (F,F,F)

- 7.1) A.OR..NOT.B
- 7.2) .NOT.(A.AND.B)
- 7.3) .NOT.A.AND.B
- 7.4) .NOT.A.OR..NOT.B
- 7.5) A.AND..TRUE..OR.(1+2.EQ.4)
- 7.6) A.AND.(B.OR.C)
- 7.7) (A.AND.B).OR.(A.AND.C)

8. จงเขียนนิพจน์ตรรกะสำหรับเงื่อนไขต่อไปนี้

- 8.1) X มีค่ามากกว่า 3
- 8.2) $2 < Y < 5$
- 8.3) R มีค่าลบและ Z มีค่าบวก
- 8.4) ALPHA และ BETA มีค่าบวกทั้งคู่
- 8.5) $-5 \leq X \leq 5$
- 8.6) A มีค่าน้อยกว่า 6 หรือมีค่ามากกว่า 10
- 8.7) $A+B > C$ และ $B+C > A$ และ $C+A > B$
- 8.8) ALPHA และ BETA มีเครื่องหมายเหมือนกัน (คือเป็นบวกทั้งคู่หรือเป็นลบทั้งคู่)

8.9) K เท่ากับ $K/2*2$ (ตรวจสอบว่า K เป็นเลขคู่หรือเลขคี่)