

บทที่ 1

ภาษาฟอร์แทรนเบื้องต้น (ตอนที่ 1)

1.1 นิยามตัวอย่าง

1.2 ส่วนประกอบของภาษาฟอร์แทรน

1.2.1 ชุดของตัวอักษร (Character set)

1.2.2 ค่าคงที่ (Constant)

1.2.3 ตัวแปร (Variable)

1.2.4 การคำนวณแบบ integer และแบบ real

1.2.5 นิพจน์เลขคณิต (Arithmetic expression)

1.2.6 คำสั่งในการกำหนดค่า (Assignment statement)

1.2.7 นิพจน์แบบผสมคณิต (Mixed mode expression)

1.2.8 คำสั่งชนิดพิเศษในการกำหนดค่า

1.2.9 FORTRAN coding form

1.2.10 คำสั่ง WRITE

1.2.11 คำสั่ง FORMAT

1.2.12 คำสั่ง STOP

1.2.13 คำสั่ง END

1.2.14 คำสั่ง PAUSE

1.2.15 ข้อผิดพลาด (errors)

บทที่ 1

ภาษาฟอร์แทรนเบื้องต้น (ตอนที่ 1)

ในบทนี้จะเริ่มจากปัญหาตัวอย่างและโปรแกรมภาษาฟอร์แทรนเพื่อแก้ปัญหาตัวอย่างนั้น และจะอธิบายคำสั่งในโปรแกรมอย่างง่าย ๆ พอให้ได้แนวความคิดในการเขียนโปรแกรมอย่างง่ายที่สุด

1.1 ปัญหาตัวอย่าง

นายสมิตที่คนสี่เหลี่ยมผืนผ้า 2 แปลง แปลงที่หนึ่งมีความยาว (W1) 75.6 ม. และความกว้าง (E1) 121.5 ม. แปลงที่สองมีความยาว (W2) 98.5 ม. และความกว้าง (E2) 110.6 ม. ต้องการคำนวณและพิมพ์ของทั้งสองแปลง และหาพื้นที่รวมของทั้งสองแปลง ตัวอย่างโปรแกรมภาษาฟอร์แทรน พร้อมทั้งแผนภูมิสายงาน (flowchart) และคำอธิบายอยู่ในรูปที่ 1

คำสั่ง 5 ชนิดของภาษาฟอร์แทรนที่ใช้ในโปรแกรมตัวอย่างคือ

1. คำสั่งใช้ในการกำหนดค่า (Replacement หรือ Assignment statement) ซึ่งใช้ในการคำนวณเช่นคำสั่งกำหนดค่าเลขคณิต (Arithmetic assignment statement)

$A1=W1*E1$

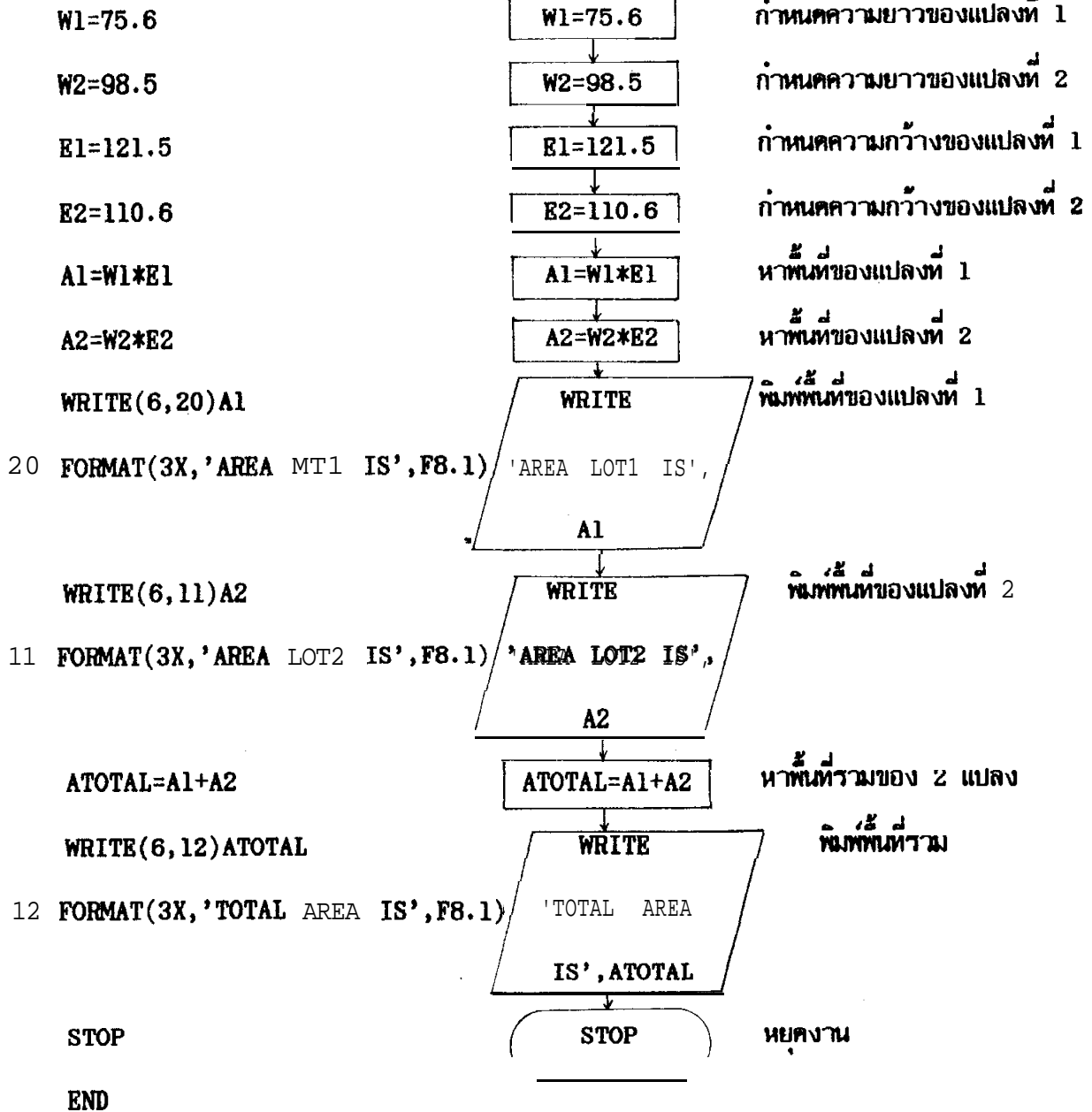
2. คำสั่ง WRITE ซึ่งใช้กำหนดว่าต้องการพิมพ์ค่าของตัวแปรใดบ้าง
3. คำสั่ง FORMAT ซึ่งใช้กำหนดตำแหน่งที่จะแสดงเลขจำนวนและข้อความในการพิมพ์ผล

พิมพ์ผล

4. คำสั่ง STOP ซึ่งใช้สั่งหยุดการปฏิบัติงานของโปรแกรม
5. คำสั่ง END ซึ่งต้องใช้เป็นคำสั่งสุดท้ายของโปรแกรม

โปรแกรมภาษาฟอร์แทรน

คำอธิบาย



รูปที่ 1 การคำนวณหาพื้นที่ของที่ดิน 2 แปลง

โปรแกรมในภาษาฟอร์แทรน

W1=75.6

W2=98.5

E1=121.5

E2=110.6

A1=W1*E1

A2=W2*E2

WRITE(6,20)A1

20 FORMAT(3X,'AREA LOT1 IS',F8.1)

WRITE(6,11)A2

11 FORMAT(3X,'AREA LOT2 IS',F8.1)

ATOTAL=A1+A2

WRITE(6,12)ATOTAL

12 FORMAT(3X,'TOTAL AREA IS',F8.1)

STOP

END

1.2 ส่วนประกอบของภาษาฟอร์แทรน

1.2.1 ชุดอักขระ (Character set)

ตัวอักขระในภาษาฟอร์แทรนตามที่สถาบันมาตรฐานแห่งชาติของสหรัฐอเมริกา (American National Standard Institute:ANSI) ได้กำหนดไว้ มีดังต่อไปนี้คือ

ตัวอักขระ	ความหมาย
1. ตัวเลข 0-Q	ตัวเลข (digit)
2. ตัวอักษร A-Z	ตัวอักษรตัวพิมพ์ใหญ่ (Uppercase letters)

3. ตัวอักขระพิเศษ

ไม่ปรากฏเครื่องหมาย	ช่องว่าง (blank หรือ space)
\$	ดอลลาร์ (dollar sign)
'	อะโพสโทรฟี่ (apostrophe)
(วงเล็บเปิด (right parenthesis)
)	วงเล็บปิด (left parenthesis)
*	ดอกจัน (asterisk)
+	บวก (plus sign)
-	ลบ (minus sign)
/	ขีดทับ (slash)
,	จุลภาค (comma)
.	มหัพภาค (period)
:	มหัพภาคคู่ (colon)
=	เท่ากับ (equal sign)

1.2.2 ค่าคงที่ตัวเลข (Numeric constant)

ค่าคงที่ตัวเลข คือค่าของเลขจำนวนที่ถูกกำหนดไว้อย่างชัดเจนหรืออาจกล่าวว่าเป็นเลขจำนวนหนึ่งนั่นเอง ค่าคงที่ตัวเลขอาจเป็นชนิดจำนวนเต็ม (integer mode) หรือชนิดจำนวนจริง (real mode) ซึ่งจะเรียกว่าเลขจำนวนเต็ม (integer constant) และเลขจำนวนจริง (real constant) ตามลำดับ เราเขียนเลขจำนวนเต็มโดยไม่มีจุดทศนิยม เลขจำนวนจริงนั้นเราเขียนโดยมีจุดทศนิยม แต่ละชนิดอาจมีเครื่องหมายบวก (+) หรือลบ (-) นำก็ได้

ตัวอย่าง

300	-2	63247	0	+4	เป็นเลขจำนวนเต็ม
6.32	-3.21	.0005	63.04	-2.0	0. เป็นเลขจำนวนจริง

ในโปรแกรม ช่องว่างระหว่างตัวเลขในค่าคงที่ตัวเลขไม่มีผลต่อค่าของมัน

ตัวอย่าง ค่าคงที่ตัวเลขต่อไปนี้เหมือนกัน

6 32 , 632 , 6 3 2

นอกจากตัวอักษรพิเศษ 3 ตัวคือ +, - และ . แล้ว เราไม่ใช้ตัวอักษรพิเศษในการเขียนค่าคงที่ตัวเลข

ตัวอย่าง ค่าคงที่ตัวเลขต่อไปนี้ใช้ไม่ได้ (invalid) และเหตุผลที่ใช้ไม่ได้

632,000	มีเครื่องหมาย ,
23.34.	ยอมให้มีจุดทศนิยม 1 จุดเท่านั้น
\$ 30.50	ตัวอักษรพิเศษ '\$' ใช้ไม่ได้
111-334-444	ตัวอักษรพิเศษ '-' ใช้ไม่ได้

ขนาด (magnitude) ของเลขจำนวนเต็มและเลขจำนวนจริงที่ใช้ได้และวิธีการเก็บค่าคงที่ตัวเลขภายในหน่วยความจำของคอมพิวเตอร์นั้นจะได้อีกกล่าวถึงในตอนหลัง วิธีการเก็บเลขจำนวนเต็มและเลขจำนวนจริงภายในหน่วยความจำของคอมพิวเตอร์นั้นแตกต่างกัน

1.2.3 ตัวแปร (Variable)

ตัวแปรคือค่าที่หลายค่า ชื่อของตัวแปรประกอบด้วย 1-6 ตัวอักษร (จำนวนตัวอักษรที่ประกอบกันเป็นชื่อตัวแปรนั้นขึ้นอยู่กับเครื่องที่ใช้ด้วย เช่น บางเครื่องอาจยอมให้ใช้ถึง 8 ตัวอักษร) โดยที่ตัวอักษรตัวแรกต้องเป็นตัวอักษร อักษรพิเศษใช้ไม่ได้ ตัวอย่างในคำสั่ง

PAY=HRS*RATE

PAY, HRS และ RATE คือตัวแปร

ตัวอย่าง ตัวแปรที่ใช้ได้ (Valid) .

X	Q1	COUNT
ABC	SUM	X12345

ตัวแปรที่ใช้ไม่ได้ (Invalid) และเหตุผลที่ทำให้มันใช้ไม่ได้

INVOICES

ยาวเกินไป (สำหรับบางเครื่องที่ยอมให้ใช้ได้

อย่างมากไม่เกิน 6 ตัวอักษร)

A-B

อักขระพิเศษ '-' ใช้ไม่ได้

4PAY

อักขระตัวแรกไม่เป็นตัวอักษร

LIGHT.

อักขระพิเศษ '.' ใช้ไม่ได้

ช่องว่างระหว่างตัวอักษรในชื่อ ไม่ทำให้ชื่อเปลี่ยนไปในโปรแกรม

ตัวอย่าง ชื่อต่อไปนี้ เป็นชื่อตัวแปรเดียวกัน

ABC

A BC

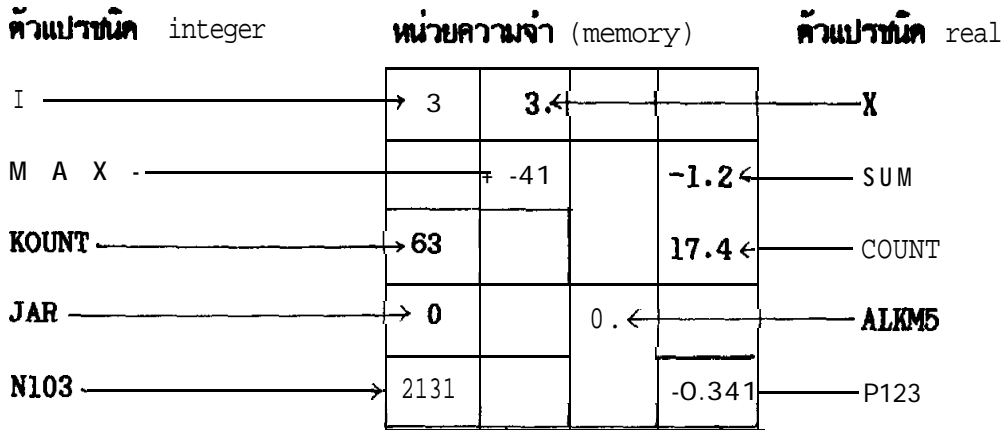
A B C

ตัวแปรหนึ่งอาจเก็บ เลขจำนวนเต็มหรือ เลขจำนวนจริง ได้ทั้งนี้ขึ้นอยู่กับชนิดของตัวแปร การกำหนดชนิดของตัวแปรให้มีชนิด integer หรือชนิด real นั้นเราอาจกำหนดโดยวิธีการตั้งชื่อ นั่นคือถ้าอักษรตัวแรกของชื่อเป็นตัวใดตัวหนึ่งจาก

I, J, K, L, M หรือ N

ตัวแปรนั้น ๆ จะมีชนิด integer แต่ถ้าขึ้นต้นด้วยตัวอักษรอื่นนอกจากข้างต้นตัวแปรจะมีชนิด real การกำหนดชนิดของตัวแปรวิธีนี้เรียกว่าการกำหนดค้อย่าง implicit ซึ่งอาจใช้คำสั่ง IMPLICIT ได้ด้วย

ตัวแปรชนิด integer จะหมายถึงที่ภายในหน่วยความจำของคอมพิวเตอร์ที่ใช้เก็บเลขจำนวนเต็ม ในขณะที่ตัวแปรชนิด real จะหมายถึงที่ภายในหน่วยความจำของคอมพิวเตอร์ที่ใช้เก็บเลขจำนวนจริง ดูตัวอย่างจากรูป



ระบบคอมพิวเตอร์จะกำหนดค่าภายในหน่วยความจำให้แก่ตัวแปรต่าง ๆ เอง

1.2.4 การคำนวณแบบ integer และแบบ real

ความแตกต่างเบื้องต้นของเลขจำนวนเต็มและเลขจำนวนจริงคือเลขจำนวนจริงมีส่วนที่เป็นเลขทศนิยมแต่เลขจำนวนเต็มไม่มี เมื่อทำการคำนวณกับข้อมูล 2 ตัวที่อาจเป็นเลขจำนวนเต็มและ/หรือตัวแปรชนิด integer ผลลัพธ์จะเป็นเลขจำนวนเต็ม แต่ถ้าข้อมูลทั้ง 2 ตัวเป็นชนิด real ผลลัพธ์จะเป็นเลขจำนวนจริง

ตัวอย่าง

นิพจน์ (expression)	ค่า (value)
3/4	0 (3 และ 4 เป็นเลขจำนวนเต็ม)
8/3	2
3./4.	.75 (3. และ 4. เป็นเลขจำนวนจริง)
8./3.	2.666667
(1/2) + (1/2)	0
(1./2.) + (1./2.)	1.

1.2.5 นิพจน์เลขคณิต (Arithmetic expressions)

นิพจน์เลขคณิตอาจเป็นค่าคงที่ตัวหนึ่ง ตัวแปรตัวหนึ่ง หรือประกอบขึ้นจากค่าคงที่และ/หรือตัวแปรที่ถูกเชื่อมกันด้วยตัวดำเนินการทางคณิตศาสตร์ (arithmetic operators) เราอาจใช้เครื่องหมายวงเล็บเพื่อแสดงลำดับที่ของการคำนวณ

ตัวดำเนินการทางคณิตศาสตร์	ความหมาย
+	การบวก (addition)
-	การลบ (subtraction)
*	การคูณ (multiplication)
/	การหาร (division)
**	การยกกำลัง (exponentiation)

ตัวอย่าง นิพจน์ที่ใช้ได้ในภาษาฟอร์แทรน

นิพจน์ภาษาฟอร์แทรน	นิพจน์พีชคณิต
A	a
14	14
(A/B)*C	$\frac{ac}{b}$
A*B-30.	ab-30
-C	-c
(A*B)**2	$(ab)^2$
(-C+B)*D	$(-c+b)d$
A**B	a^b
-3.7	-3.7
A W . 5	\sqrt{a}
((A-B)**3)**.25	$\sqrt[4]{(a-b)^3}$
PMT*(1+IC)**N/IC-1	$PMT \frac{(1+ic)^n - 1}{ic}$
-2*V*(X*X+Y*Y)/(U*U+V*V)**2	$\frac{-2v(x^2+y^2)}{(u^2+v^2)^2}$
F*((LAMPDA-1)/RHO-L*(L+1)/(RHO*RHO))	$F \left[\frac{\lambda-1}{\rho} - \frac{l(l+1)}{\rho^2} \right]$

ตัวอย่าง นิพจน์ที่ใช้ไม่ได้ และเหตุที่ทำให้ใช้ไม่ได้

$3(A+JB)$

ตัวดำเนินการหลังเลข 3 ไม่มี

$A-(B+C*(K)$

จำนวนวงเล็บเปิดและปิดไม่เท่ากัน

$X*-3$

ตัวดำเนินการ 2 ตัวเขียนติดกันไม่ได้

ควรแก้เป็น $X*(-3)$ หรือ $-3*X$

เมื่อในนิพจน์มีเครื่องหมายวงเล็บอยู่ การคำนวณในเครื่องหมายวงเล็บจะถูกทำก่อน และถ้ามีวงเล็บซ้อนกันอยู่หลายชั้น จะทำการคำนวณนิพจน์ที่อยู่ในวงเล็บชั้นในสุดก่อน

ตัวอย่าง

นิพจน์	การคำนวณ
$3*(4+5)$	$3*9 = 27$
$3*(4+(8/2))$	$3*(4+4) = 3*8 = 24$

ถ้าในนิพจน์ไม่มีเครื่องหมายวงเล็บ ลำดับของการคำนวณจะเป็นไปตามกฎของกามิลลำดับเหนือกว่า (rule of precedence) ดังต่อไปนี้

การดำเนินการ (operation)	กามิลลำดับเหนือกว่า
**	สูง
* uia /	กลาง
+ หรือ -	ต่ำ

การดำเนินการที่มีลำดับสูงกว่าจะถูกทำก่อนการดำเนินการที่มีลำดับต่ำกว่า การดำเนินการที่มีลำดับเท่ากันคือ บวก/ลบ และคูณ/หาร ให้ทำจากซ้ายไปขวา ส่วนการยกกำลังนั้นจะทำจากขวาไปซ้าย

ตัวอย่าง

นิพจน์	การดำเนินการ
1. $A-B+C$	$(A-B)+C$

นิพจน์	การดำเนินการ
2. $A+B*C$ $3.+2.*3.$	$A+(B*C)$ $3.+(2.*3.)=3.+6.=9.$
3. $A/B*C$ $9./4.*2.$	$(A/B)*C$ $(9./4.)*2.=2.25*2.=4.50$
4. $A/B/C$ $8./4./2.$	$(A/B)/C$ $(8./4.)/2.=2./2.=1.$
5. $(A+B)/C*D$ $(3.+6.)/3.*6.$	' $((A+B)/C)*D = \frac{A+B}{C} *D$ ไม่ใช่ $\frac{A+B}{C*D}$ $((3.+6.)/4.)*6.=\frac{9.}{3.}*6.=3.*6.=18.$
6. $A+B*C**2$ $3.+3.*2.**2$	$A+(B*(C**2))$ $3.+(3.*2.^2)=3.+(3.*4.)=3.+12.$ $=15.$
7. $A**B**C$ $3.**2.**3.$	$A**(B**C)$ (ไม่ใช่ $(A**B)**C$) $3.**(2.**3.)=3.**8.=6561.$ $(ไม่ใช่ (3.**2.)**3.=9.**3.=729.)$

1.2.6 คำสั่งในการกำหนดค่า (Replacement หรือ Assignment statement)

รูปทั่วไป

ตัวแปร = นิพจน์

คำสั่งนี้จะทำการคำนวณค่าของนิพจน์ก่อน แล้วค่าของมันจะถูกเก็บไว้ในตัวแปร
ทางด้านซ้ายของคำสั่ง เครื่องหมายเท่ากับ (=) เป็นเครื่องหมายแสดงการแทนที่ คำสั่ง $X=X+1$
ใช้ได้เพราะหมายความว่า บวกหนึ่งเข้ากับค่าของ X ที่มีอยู่ แล้วเก็บผลลัพธ์ใหม่ใน X (X มีค่า
เพิ่มจากเดิมอีกหนึ่ง)

ตัวอย่าง คำสั่งกำหนดค่าเลขคณิต (Arithmetic assignment statement) ที่ใช้ได้

$x=3.123$

กำหนด x ให้มีค่า 3.123 (เก็บ 3.123 ในที่ ๆ ชื่อ x)
(ค่าเก่าของ x จะหายไป)

$C1=(A+B)/C$

คำนวณ $(A+B)/C$ แล้วเก็บผลลัพธ์ใน $C1$

$Z=3.**2$

z มีค่าเท่ากับกำลังสองของ 3

$SK=(Z+4.)**.5$

SK มีค่า $\square \sqrt{Z+4}$

ตัวอย่าง คำสั่งกำหนดค่าที่ใช้ไม่ได้และเหตุผล

$3.16=X$

ตัวแปรเท่านั้นที่จะปรากฏทางซ้ายของเครื่องหมายเท่ากับ

$X+Y=1$

นิพจน์จะปรากฏทางซ้ายของเครื่องหมายเท่ากับไม่ได้

$HRS*RATE+BONUS$

ไม่ได้กำหนดตัวแปรที่จะเก็บค่าของนิพจน์นี้

1.2.7 **นิพจน์แบบผสมคณิต** (Mixed-mode expression)

นิพจน์แบบผสมคณิตคือนิพจน์ที่ประกอบด้วยค่าคงที่ และ/หรือตัวแปรที่มีชนิดต่างกัน
ทุกตัว ๆ ไปในการคำนวณค่านิพจน์ชนิดนี้คือ ถ้าเลขจำนวนเต็มและเลขจำนวนจริงอยู่ในการ
คำนวณ เลขจำนวนเต็มจะถูกเปลี่ยนไปเป็นเลขจำนวนจริงก่อนจะถูกคำนวณภายในเครื่องคอมพิวเตอร์
และผลลัพธ์จะมีค่าเป็นเลขจำนวนจริง

ตัวอย่าง

นิพจน์	ค่าของนิพจน์	คำอธิบาย
$3/2.$	1.6	3 ถูกเปลี่ยนเป็น 3.
$3./2$	1.5	2 ถูกเปลี่ยนเป็น 2.
$4.+3/2$	5.	$3/2=1.5$, 1 ถูกเปลี่ยนเป็น 1. ก่อนบวกกับ 4.
$4.*3/2$	6.	$4.*3.=12.$, $12./2.=6.$

นิพจน์	ค่าของนิพจน์	คำอธิบาย
$4.*(3/2)$	4.	$3/2=1, 1$ ถูกเปลี่ยนเป็น 1. ก่อนคูณกับ 4.
$4+3/2.$	5.5	$3./2.=1.5, 4.+1.5=5.5$
$I+7.2$	6.2	ถ้า $I=-1$

1.2.8 คำสั่งชนิดพิเศษในการกำหนดค่า

คำสั่งชนิดพิเศษในการกำหนดค่าคือคำสั่งในการกำหนดค่าเลขคณิตที่ชนิดของตัวแปรทางซ้ายของเครื่องหมายเท่ากับต่างไปจากชนิดของค่าของนิพจน์ที่จะถูกคำนวณหาค่าทางขวา ในการที่ชนิดของค่าของนิพจน์จะถูกเปลี่ยนเป็นชนิดเดียวกันกับตัวแปรทางด้านซ้ายก่อนที่ค่าของมันจะถูกเก็บในตัวแปรนั้น นั่นคือชนิดของตัวแปรทางซ้ายของเครื่องหมายเท่ากับจะเป็นตัวกำหนดชนิดของค่าของนิพจน์ทางขวา

ตัวอย่าง

คำสั่ง	คำอธิบาย
$IX=3.2$	ค่าที่เก็บใน IX คือเลขจำนวนเต็ม 3 เมื่อเลขจำนวนจริงถูกแปลงเป็นเลขจำนวนเต็ม เลขทศนิยมจะถูกตัดทิ้ง (truncate) ไปเลย
$X=3+2$	ค่าของ x คือ 5. ไม่ใช่ 6
$J=3./2$	ค่าของนิพจน์คือ 1.6 แต่ค่าที่เก็บใน J คือ 1 เพราะ J เป็นตัวแปรชนิด integer
$KX=4/3+6.8$	$4/3=1, 1.+6.8=7.8$ แต่ค่าที่เก็บใน KX คือ 7 เพราะ KX เป็นตัวแปรชนิด integer (ให้สังเกตว่า .8 ถูกตัดทิ้งไปเลย โดยไม่มีการปัดจตุทศนิยม แม้ว่าทศนิยมจะเกิน .5)

1.2.9 FORTRAN coding form

โปรแกรมภาษาฟอร์แทรนประกอบด้วยชุดของคำสั่ง คำสั่งเหล่านี้จะถูกบันทึกลงในฟิลด์ (field) ที่ระบุตำแหน่งแน่นอนบนบรรทัดของจอภาพ (ถ้าเขียนโปรแกรมผ่านทางจอภาพ)

ฟิลด์ต่อไปนี้ใช้ในการบันทึกคำสั่งภาษาฟอร์แทรน

สัณฐาน (column)	สิ่งที่บันทึก
1-6	เลขประจำคำสั่ง (statement number) (ไม่ใช้ก็ได้ แต่สำหรับคำสั่ง FORMAT ต้องมีเลขประจำคำสั่งเสมอ)
6	ใส่ตัวอักษรใด ๆ เพื่อแสดงการต่อบรรทัด (ไม่ต่อก็ได้ ถ้าเขียนคำสั่งใน 1 บรรทัดพอ) กรณีที่คำสั่งยาวเกิน 1 บรรทัด
7-72	คำสั่ง (statement)
73-80	ชื่อและลำดับของบรรทัด (มีหรือไม่ก็ได้)

เลขประจำคำสั่งของคำสั่งหนึ่งเป็นเลขจำนวนเต็ม ไม่มีเครื่องหมาย คำสั่งที่ต้องการอ้างอิงโดยคำสั่งอื่นเท่านั้นเราจึงจะกำหนดเลขประจำคำสั่งให้มัน เลขประจำคำสั่งอาจเป็นเลขจำนวน 1-5 หลัก เขียนที่ใดก็ได้ในสัณฐาน 1-5 และไม่จำเป็นต้องเรียงลำดับกัน บางเครื่องบังคับว่าต้องบันทึกเลขประจำคำสั่งซิกขวาของฟิลด์นั้น (สัณฐาน 1-5) เพราะมันจะถือว่าช่องว่างคือเลขศูนย์

คำสั่งภาษาฟอร์แทรน เขียนที่ใด ๆ ก็ได้ในสัณฐาน 7-72 เพื่อความสะดวกในการอ่าน โปรแกรม โปรแกรมเมอร์ใหม่อาจจะเขียนทุกคำสั่งให้เริ่มต้นที่สัณฐาน 7 ถ้าคำสั่งหนึ่งยาวมากไม่สามารถเขียนลงในสัณฐาน 7-72 ของบรรทัดหนึ่งได้ เราอาจต่อบรรทัดโดยการใส่ตัวอักษรใด ๆ ก็ได้ (ยกเว้นช่องว่างและศูนย์) ในสัณฐานที่ 6 ของบรรทัดใหม่ แล้วเขียนคำสั่งที่เกินมาในสัณฐาน 7-72 ด้ทันที

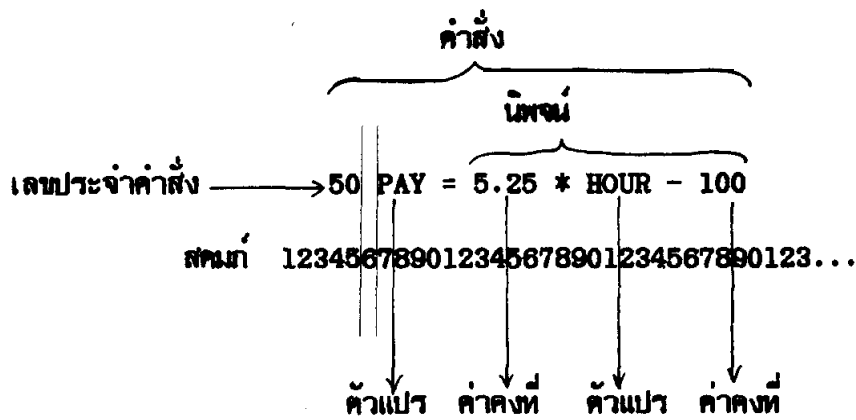
แผ่นที่ 73-80 ใช้เพื่อแสดงชื่อผู้เขียนหรือชื่อโปรแกรม และ/หรือลำดับที่ของโปรแกรม
ในบรรทัด หรืออาจเว้นว่างไว้ก็ได้ สิ่งที่เขียนลงในฟิลด์นี้ คำคอมพิวเตอร์จะไม่แปลเป็นภาษา-
เครื่อง (Machine language)

Coding sheet ช่วยให้การเขียนโปรแกรมที่เตรียมไว้ลงในบรรทัดของจอภาพได้
สะดวกขึ้น

A FORTRAN coding form

IBM		FORTRAN CODING FORM		X38-7327-6 10/74 800 Printed in U.S.A.	
PROGRAM	DATE	PUNCHING INSTRUCTIONS	GRAPHIC PUNCH	PAGE	OF
PROGRAMMER				CARD (MICROFILM)	
STATEMENT NUMBER	FORTRAN STATEMENT				INDENTATION INCHES
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80					
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80					

โดยทั่วไป คำสั่งภาษาฟอร์แทรนประกอบด้วยส่วนต่าง ๆ ที่แน่นอน ซึ่งอาจเป็น
คีย์เวิร์ด (keyword) คำคงที่ ตัวแปร อักขระพิเศษ เขียนประกอบกันขึ้นภายใต้กฎเกณฑ์ตาม
หลักไวยากรณ์ของภาษา (syntax) คำสั่งตามปกติที่ใช้บ่อยคือ คำสั่งในการกำหนดค่า คำสั่งควบคุม
คำสั่งนำข้อมูลเข้า/ออก ตัวอย่างต่อไปนี้แสดงส่วนต่าง ๆ ของคำสั่งในการกำหนดค่าเลขคณิต
และแสดงตำแหน่งเจาะบนบรรทัด



ช่องว่าง (blank) อาจมีอยู่ที่ใด ๆ ก็ได้ระหว่างสแกน 7-72 ในคำสั่ง นั่นคือ
ในตัวแปร ค่าคงที่ คีย์เวิร์ด อาจมีช่องว่างรวมอยู่ด้วยก็ได้ ดังเช่นตัวอย่างข้างต้นอาจเขียน
ลงบนบรรทัดได้ดังนี้

60 P AY = 6.2 5 *HRS - 100

สแกน 1234567890123456789012345678901234567...

1.2.10 คำสั่ง WRITE

รูปทั่วไป

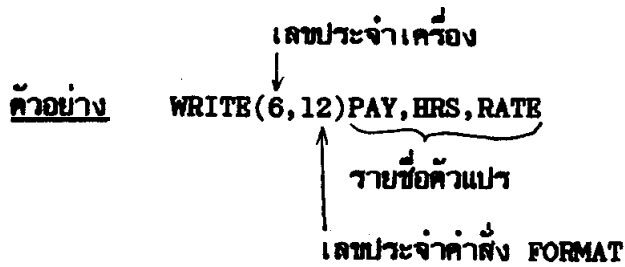
WRITE(เลขประจำเครื่อง, เลขประจำคำสั่ง FORMAT) [รายชื่อตัวแปร]

โดยที่ - เลขประจำเครื่อง คือเลขจำนวนเต็มชี้แทนเครื่องนำข้อมูลออกที่เราจะใช้ซึ่งขึ้นอยู่กับระบบคอมพิวเตอร์ที่ใช้ เช่น 6 อาจหมายถึงเครื่องพิมพ์หรือ 6 อาจหมายถึงจอภาพ เป็นต้น

- เลขประจำคำสั่ง FORMAT ซึ่งคำสั่ง FORMAT ดังกล่าวจะถูกใช้ในการแสดงผลลัพธ์ นั่นคือเป็นการกำหนดรูปแบบของข้อมูลออกนั่นเอง

- รายชื่อตัวแปร (variable list) เป็นตัวแปรที่เราต้องการแสดงค่าของมัน ถ้ามีตัวแปรเกิน 1 ตัวให้คั่นด้วยเครื่องหมายจุลภาค (,)

เมื่อคอมพิวเตอร์ปฏิบัติตามคำสั่ง WRITE ค่าของตัวแปรในรายชื่อตัวแปรจะถูกแสดงออกที่เครื่องนำข้อมูลออกที่ระบุไว้และตามรูปแบบที่กำหนดในคำสั่ง FORMAT ที่มีเลขประจำคำสั่งที่อ้างในคำสั่ง WRITE



คำสั่งนี้หมายความว่า ให้พิมพ์ค่าที่เก็บในส่วน of หน่วยความจำชื่อ PAY, HRS และ RATE ลงบนเครื่องนำข้อมูลออกหมายเลข 6 ตามรูปแบบที่กำหนดในคำสั่ง FORMAT หมายเลข 12 ไปรอสั่งเกตุว่าค่าของตัวแปรทั้ง 3 แม้ว่าจะถูกนำออกมาแสดงแล้ว ค่าของตัวแปรทั้ง 3 ในหน่วยความจำจะไม่หายไป นอกจากว่าจะมีคำสั่งกำหนดค่าให้มันใหม่

ในกรณีที่ใช้เครื่องพิมพ์เป็นเครื่องนำข้อมูลออก ขนาดตัว ๆ ไปของ 1 บรรทัดของการพิมพ์คือ 132 ตำแหน่งหรือตัวอักษร

ตัวอย่าง คำสั่ง WRITE ที่ถูกต้อง

65 WRITE(6, 11)A, IJT	ค่าที่เก็บไว้ที่ส่วน of หน่วยความจำชื่อ A และ IJT จะถูกนำออกมาแสดงที่เครื่องนำข้อมูลออกหมายเลข 6 โดยใช้รูปแบบที่กำหนดในคำสั่ง FORMAT หมายเลข 11
WRITE(7, 15)Q	ค่าที่เก็บไว้ที่ส่วน of หน่วยความจำชื่อ Q จะถูกนำออกมาแสดงที่เครื่องนำข้อมูลออกหมายเลข 7 โดยใช้รูปแบบที่กำหนดในคำสั่ง FORMAT หมายเลข 15

ตัวอย่าง คำสั่ง WRITE ที่ไม่ถูกต้อง

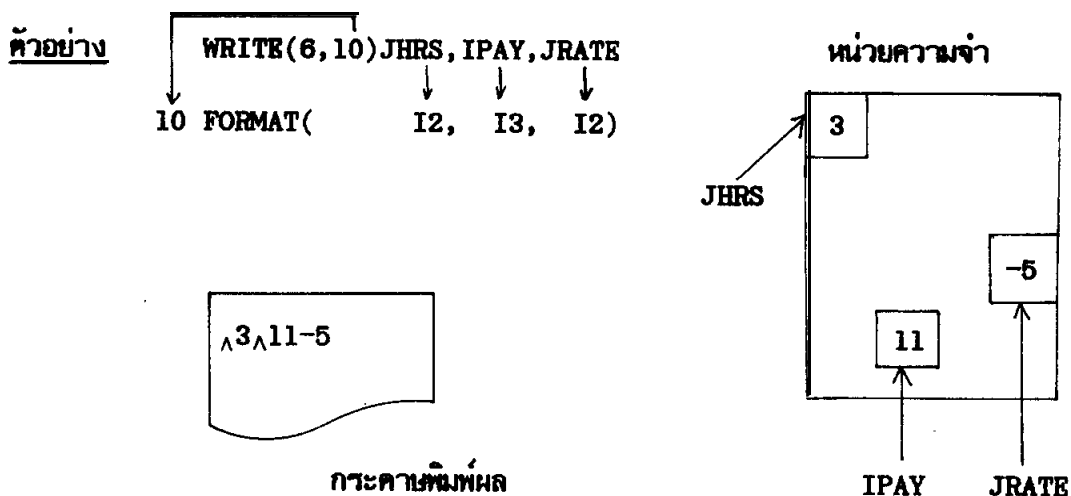
WRITE(6, 14)4	4 ไม่ใช่ตัวแปร
WRITE(6, 16)X+Y	X+Y เป็นนิพจน์ไม่ใช่ตัวแปร
WRITE(6, 11)'THE SUM IS', X	'THE SUM IS' เป็นสายวลีอักขระ ไม่ใช่ตัวแปร
WRITE(6, 15.)X, Y	15. เป็นเลขจำนวนจริง ใช้เป็นเลขประจำคำสั่งไม่ได้

1.2.11 คำสั่ง FORMAT

จะเห็นว่าคำสั่ง WRITE เพียงคำสั่งเดียวจะไม่พอเพียงสำหรับการสั่งให้เครื่องคอมพิวเตอร์พิมพ์สิ่งที่ต้องการได้ตามรูปแบบที่ต้องการ เพราะยังตอบคำถามเหล่านี้ยังไม่ได้คือ

1. ในแต่ละบรรทัดของการพิมพ์จะแสดงผลที่ตรงไหน
2. ในการแสดงผลนั้นต้องการแสดงทศนิยมกี่ตำแหน่ง
3. ในแต่ละบรรทัดจะใช้กี่ตำแหน่งสำหรับการพิมพ์

เพื่อตอบปัญหาข้างต้นเราจะใช้คำสั่ง FORMAT คู่กับคำสั่ง WRITE เพื่อที่จะบอกคอมพิวเตอร์ให้ชัดเจนว่าจะแสดงผลในรูปแบบอย่างไร แต่ละคำสั่ง WRITE จะอ้างถึงเลขประจำคำสั่ง FORMAT ที่จะใช้และแต่ละคำสั่ง FORMAT จะต้องมีเลขประจำคำสั่งเพื่อใช้ในการอ้างถึง คำสั่ง FORMAT จะใส่ไว้ที่ใดก็ได้ (ก่อนคำสั่ง END) โปรแกรมเมอร์บางคนนิยมใส่คำสั่ง FORMAT รวมไว้ตอนเริ่มต้นหรือตอนท้ายของโปรแกรมในขณะที่คนอื่นอีกหลาย ๆ คนใส่ตามหลังคำสั่ง WRITE ที่อ้างถึงทันที



[เครื่องหมาย '^' ใช้แทนช่องว่าง 1 ช่อง ถ้าเว้น 2 ช่องว่างจะเขียน ^^]

นั่นคือสั่งให้พิมพ์ค่าของตัวแปร 3 ตัวตามรูปแบบที่กำหนดในคำสั่ง FORMAT หมายเลข 10 ลงบนกระดาษพิมพ์ผล (สมมติ 6=เครื่องพิมพ์) นั่นคือพิมพ์ค่าของ JHRS แบบเลขจำนวนเต็ม (กำหนดด้วย I-format code) ใน 2 ตำแหน่งแรก 3 ตำแหน่งถัดไปพิมพ์ค่าของ IPAY แบบเลขจำนวนเต็ม และพิมพ์ค่าของ JRATE ที่ 2 ตำแหน่งถัดไปนั่นคือที่ตำแหน่ง 6-7

การสั่งพิมพ์ผลลงบนกระดาษพิมพ์ผล เราจะ ไม่สั่งพิมพ์สิ่งใดลงบนตำแหน่งแรกของทุกบรรทัด เนื่องจากเราจะใช้ตำแหน่งนี้เพื่อใส่ตัวอักษรควบคุมการพิมพ์ (Carriage control character) ในขณะที่ใช้เครื่องพิมพ์

ตัวอักษรควบคุมการพิมพ์	ความหมาย
1	ขึ้นหน้าใหม่
0	เว้น 1 บรรทัด
+	พิมพ์บรรทัดเดิม (เริ่มที่ต้นบรรทัด)
ช่องว่าง	ขึ้นบรรทัดใหม่
ตัวอักษรอื่น ๆ	อาจมีความหมายพิเศษขึ้นอยู่กับ การกำหนด

รูปแบบของคำสั่ง FORMAT

เลขประจำคำสั่ง $FORMAT(f_1, f_2, \dots, f_n)$

โดยที่ เลขประจำคำสั่ง คือเลขที่ใช้เพื่ออ้างถึงในคำสั่ง WRITE

f_1, f_2, \dots, f_n เป็น format code ซึ่งอาจเป็น

1) Data format code ที่ใช้กับข้อมูลที่จะถูกแสดง เช่น

I-format code ใช้สำหรับเลขจำนวนเต็ม

F-format code ใช้สำหรับเลขจำนวนจริง

หรือ 2) Editing format code ที่ใช้เพื่อควบคุมการวางตำแหน่งของผลลัพธ์ เช่น

X-format code ใช้สำหรับการเว้นวรรค

' (single quote) ใช้สำหรับกำหนดสายวลีอักขระ

T-format code ใช้สำหรับระบุตำแหน่งเริ่มต้นที่ใช้พิมพ์

format code ทั้งหมดต้องแยกกันด้วยเครื่องหมายจุลภาค (,)

ในบทนี้จะพิจารณาเฉพาะ format code : I, F, X, literal, และ T ก่อน

X-format code

รูปทั่วไป nX

โดยที่ n เป็นเลขจำนวนเต็มที่ระบุจำนวนช่องว่างที่ต้องการ

ตัวอย่าง K=14 ผลกาารพิมพ์

WRITE(6,11)K

11 FORMAT(3X,I2)

^^^14^^...

I-format code

รูปทั่วไป Iw

โดยที่ I ระบุว่าเลขจำนวนเต็มจะถูกพิมพ์

w คือจำนวนตำแหน่งที่เตรียมไว้สำหรับเลขจำนวนเต็มที่จะถูกพิมพ์ (ความกว้างของฟิลด์)

ถ้าเป็นเลขจำนวนเต็มลบ จำนวน w ต้องเพิ่ม 1 ที่สำหรับเครื่องหมายด้วย

ตัวอย่าง ICT = 43200

L1 = -4

JSUM = 33

WRITE(6,14)ICT, L1, JSUM

14 FORMAT(2X,I5,2X,I2,I4)

ไปคสังเกตุการจับคู่หนึ่งต่อหนึ่ง
ระหว่างตัวแปรและ format code

ผลกาารพิมพ์ ^^|43200|^|^|-4|^|^|33|^ ...

ในกรณีของ JSUM ความกว้างของฟิลด์มากกว่าจำนวนหลักของค่าของ JSUM ดังนั้นค่าของ JSUM จะถูกพิมพ์ทิศทางของฟิลด์ (right justified) ส่วนค่านซ้ายของฟิลด์จะเป็นช่องว่าง

ตัวอย่าง

K=35

J=5

WRITE(6,12)K

12 FORMAT(2X,I3)

L=K*J

WRITE(6,11)L,J

11 FORMAT(6X,I3,1X,I2)

:

:

ผลกาารพิมพ์

บรรทัดที่ 1 K
 ↓
 ΛΛΛ35|บรรทัดที่ 2 L J
 ↑ ↑
 ΛΛΛΛΛ175ΛΛ5|

โปรดสังเกตุว่าคำสั่ง WRITE แต่ละคำสั่งจะทำให้ได้ผลกาารพิมพ์ 1 บรรทัด คำสั่ง WRITE คำสั่งแรก จะทำให้เกิดกาารพิมพ์เลขจำนวนหนึ่งบนบรรทัดที่ 1 คำสั่ง WRITE คำสั่งที่สอง จะทำให้เกิดกาารพิมพ์เลข 2 จำนวนบนบรรทัดใหม่

ในการกำหนด w จะต้องกำหนดคให้พอเพียงสำหรับตัวเลขทุกตัวของเลขจำนวนนั้น ๆ (รวมทั้งสำหรับเครื่องหมายลบด้วย) ถ้ากำหนดคที่ไม่พอเพียงเครื่องคอมพิวเตอร์จะมีวิธีระบุข้อผิดพลาดซึ่งรูปแบบการการระบุข้อผิดพลาดให้โปรแกรมเมอร์ทราบนั้นจะค่างกันไปแล้วแต่ระบบ ในตัวอย่างถัดไปเป็นการใช้เครื่องหมายคอกกันเพื่อระบุว่าการกว้างของฟิลด์ไม่พอแสดงผล

ตัวอย่าง K=1234

J=-14

WRITE(6,15)K, J, J

15 FORMAT(2X,I3,2X,I2,2X,I4)

ผลกาารพิมพ์

 ↓ ↓ ↓
 ΛΛ***|ΛΛ**|ΛΛ-14|

F-format code

รูปทั่วไป

Fw.d

โดยที่ F เป็นตัวระบุว่าจะจำนวนจริงจะถูกพิมพ์

w ความกว้างของฟิลด์

d ระบุจำนวนตัวเลขทางขวาของจุดทศนิยม

จุดที่ใช้ 1 ที่ในฟิลด์ของการแสดงผล (output field) ด้วย

ตัวอย่าง A = 63.426

B = -4.2

WRITE(6,15) A, A, A, B, B

15 FORMAT(1X,F6.3,F9.5,1X,F6.1,F6.1,F4.0)

ผลการพิมพ์

ตัวอย่าง A = 12.6634

B = 13.7

WRITE(6,16) A, A, A, B

16 FORMAT(2X,F9.3,F6.1,F5.0,F7.3)

ผลการพิมพ์

ตัวอย่าง A = -132.466

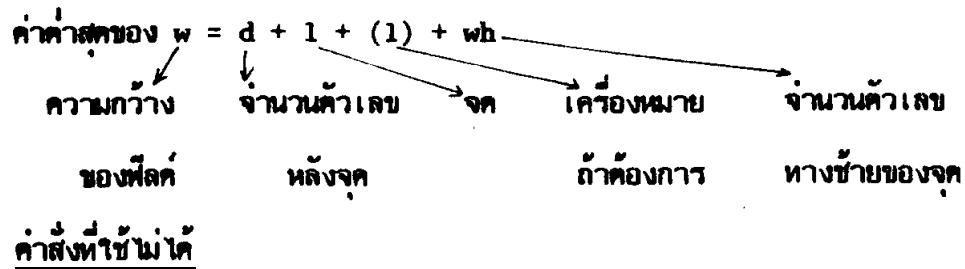
B = 2509.01

c = 12.

WRITE(6,12) A, A, B, C

12 FORMAT(1X,F7.3,1X,F5.1,1X,F6.2,1X,F5.4)

ผลการพิมพ์



WRITE(6,11)A,B,C

11 FORMAT(I2, I4, I3) ควรใช้ FORMAT(F3.0, F5.0, F4.0)

WRITE(6,12)I,J,K

12 FORMAT(F2.0, F2.0, F3.0) ควรใช้ FORMAT(I2, I2, I3)

T-format code

รูปแบบทั่วไป

Tn

โดยที่ n ระบุตำแหน่งเริ่มต้นของฟิลด์ถัดไป

ตัวอย่าง K = 35

J = 2

WRITE(6,10) J,K

10 FORMAT(T5 , I2 , T10 , I3)

ผลกาพิมพ์

กาพิมพ์คำอธิบายหรือหัวคาราง (Literal format)

เราใช้เครื่องหมายอะโพสโทรฟไว้ก่อนและหลังสายวลีอักขระ (character string) ที่เราต้องการจะพิมพ์พร้อมกับผลลัพธ์อื่น ๆ เรามักใช้พิมพ์หัวคาราง ใช้ระบุชื่อของผลลัพธ์ที่เป็นเลขจำนวน หรือใช้เขียนข่าวสาร ข้อความ หรือคำอธิบาย

ตัวอย่าง WRITE(6,10)KV

10 FORMAT(T5, 'AVERAGE IS', I3)

ผลกาพิมพ์

(ถ้า KV มีค่า 25)

ตัวอย่าง IERROR = 999

WRITE(6,6) IERROR

6 FORMAT(T2, '*** ***_', I3)

ผลกาารพิมพ์ ^|*** ***_|999|

ตัวอย่าง K = 36

J = 3

WRITE(6,11) J, K

11 FORMAT(T2, 'NO-BOYS=', I2, 2X, 'NO-GIRLS=', I3)

ผลกาารพิมพ์ ^|NO-BOYS=^3|^|^|NO-GIRLS=|^35|^
 ↑ ↑
 J K

ตัวอย่าง M = 5

ID = 26

IY = 81

WRITE(6,5) M, ID, IY

5 FORMAT(T60, 'XYZ COMPANY PAYROLL REPORT DATE ',

11X, I2, '/', I2, '/', I2)

ตำแหน่งกาารพิมพ์ 1 6 9

ผลกาารพิมพ์ |XYZ DATE|^|^|5/26/81

ตัวอย่าง WRITE(6,11)

11 FORMAT(3X, 'XYZ^CORPORATION')

ผลกาารพิมพ์ ^^^XYZ^CORPORATION

1.2.12 คำสั่ง STOP

รูปทั่วไป

STOP

เราใช้คำสั่งนี้เมื่อเราต้องการหยุดการปฏิบัติงานในโปรแกรม ในโปรแกรมภาษาฟอร์แทรนทุกโปรแกรมต้องมีคำสั่ง STOP อย่างน้อย 1 คำสั่ง คำสั่งนี้อาจปรากฏมากกว่า 1 ครั้งถ้าต้องการ

1.2.13 คำสั่ง END

รูปทั่วไป

END

ในโปรแกรมภาษาฟอร์แทรนต้องมีคำสั่ง END เพียงคำสั่งเดียวและอยู่เป็นคำสั่งสุดท้ายของโปรแกรมเสมอ คำสั่ง END เป็นคำสั่งไม่ปฏิบัติการ เราจึงไม่กำหนดเลขประจำคำสั่งให้

1.2.14 คำสั่ง PAUSE

รูปทั่วไป

PAUSE

เป็นคำสั่งให้คอมพิวเตอร์หยุดปฏิบัติงานชั่วคราวในขณะที่วิ่งโปรแกรมอยู่ ผู้ควบคุมเครื่อง (operator) จะเป็นผู้สั่งให้คอมพิวเตอร์ปฏิบัติงานต่อไปใหม่ได้โดยผ่านทางคอนโซล คำสั่ง PAUSE แตกต่างจากคำสั่ง STOP ตรงที่ว่า เมื่อคอมพิวเตอร์หยุดงานตามคำสั่ง STOP แล้วจะไม่ปฏิบัติงานตามคำสั่งในโปรแกรมนั้น ๆ อีก นอกจากเริ่มบรรทัด และวิ่งโปรแกรมใหม่

1.2.15 ข้อผิดพลาด (Errors)

ข้อผิดพลาดอาจเกิดจากการกดแป้นพิมพ์ผิดพลาด ผิดไวยากรณ์ ผิดตรรกะ (โปรแกรมไม่สามารถแก้ปัญหาที่มีอยู่ได้) ผิดขณะปฏิบัติงาน ผิดที่คำสั่งควบคุมระบบ หรือผิดขั้นตอนการทำงาน ประมวลผล ข้อผิดพลาดนั้นนอกจากใช้คำว่า error แล้ว ยังนิยมเรียกว่า bug โปรแกรมเมอร์จะมีความสามารถมากแค่ไหนมักจะคู่ความสามารถในการแก้ไขข้อผิดพลาดในโปรแกรม (debug)

ข้อผิดพลาดที่มักเกิดขึ้นบ่อย ๆ ขณะทำการประมวลผลโดยใช้ภาษาฟอร์แทรน อาจแบ่งเป็น 2 พวกคือ

1. ข้อผิดพลาดขณะแปลโปรแกรม (Compile-time error หรือ Syntax error) คอมไพเลอร์จะเป็นผู้พบข้อผิดพลาดชนิดนี้ขณะทำการแปลโปรแกรม รายงานข้อผิดพลาด (diagnostic error list) จะถูกแสดงออกมาพร้อมกับโปรแกรมของเรา (program listing)

ตัวอย่าง

$X = 3(2+A)$ ขาดเครื่องหมายคูณ
 $Z = X+(Y-Z$ ขาดเครื่องหมายวงเล็บปิด

2. ข้อผิดพลาดขณะวิ่งโปรแกรม (Run-time error หรือ Logical error) ถึงแม้ว่าตอนที่คอมไพเลอร์แปลโปรแกรมแล้วไม่พบข้อผิดพลาด แต่หน่วยประมวลผลกลาง (Central processing unit : CPU) อาจจะไม่สามารถปฏิบัติตามคำสั่งได้ คอมไพเลอร์เพียงแต่แปลคำสั่งในภาษาฟอร์แทรนไปเป็นคำสั่งในภาษาเครื่อง แต่ไม่ได้ปฏิบัติตามคำสั่งในภาษาเครื่อง มันจึงไม่สามารถบอกได้ว่าคำสั่งนั้น ๆ เป็นไปได้ในทางปฏิบัติหรือไม่

ตัวอย่าง

$I = 0$
 $10 T = J/I$ ขี้นัยในทำการหารด้วยเลขศูนย์

$WRITE(5,15)X$ ถ้า 5 คือเครื่องอ่านบัตร (เท่านั้น) คอมพิวเตอร์ไม่สามารถบันทึกลงบนเครื่องอ่านบัตรได้

โปรแกรมควบคุมระบบ (OS) อาจเป็นผู้พบข้อผิดพลาดตรรกะ เช่นคำสั่งที่ทำให้เกิดลูป (loop) ไม่รู้จบขึ้น เช่นคำสั่ง

5 GO TO 5

ซึ่งระบบอาจพิมพ์ข้อความบอกว่าเวลาที่ใช้เกินระยะเวลาที่กำหนดไว้ (time exceeded) ความผิดพลาดชนิดนี้มักจะทำให้เครื่องคอมพิวเตอร์หยุดทำงานนั้น ๆ และระบบจะพิมพ์รายการข้อผิดพลาดออกมาให้

แบบฝึกหัดที่ 1

1. จงบอกว่่าสิ่งต่อไปนี้คืออะไรในภาษาฟอร์แทรน นั่นคือเป็นเลขจำนวนเต็ม เลขจำนวนจริง ตัวแปรชนิด integer หรือตัวแปรชนิด real บางข้ออาจใช้ไม่ได้ในภาษาฟอร์แทรน จงบอก

เหตุผล

- | | | | |
|--------------|------------|----------------|-------------------|
| 1.1) F | 1.2) 1123 | 1.3) FORTRAN | 1.4) x1.3 |
| 1.5) -1234 | 1.6) 3ABC | 1.7) XRAY | 1.8) .000000006 |
| 1.9) +72 | 1.10) 4(Y) | 1.11) A1B2C | 1.12) 234-567-999 |
| 1.13) 1A2B3C | 1.14) IRAY | 1.15) COUNT | 1.16) KOUNT |
| 1.17) 3+4 | 1.18) I2+1 | 1.19) \$300.50 | 1.20) 3<4 |

2. จงเขียนนิพจน์ต่อไปนี้เป็นนิพจน์ในภาษาฟอร์แทรน

- | | | |
|---|--|--|
| 2.1) $x(y+z)$ | 2.2) $\frac{a}{b}c$ | 2.3) $\frac{a}{b^3}$ |
| 2.4) ax^2+bx+c | 2.5) $\frac{a}{b^3}$ | 2.6) $y^{1/3}$ |
| 2.7) $\sqrt{r^2}$ | 2.8) $\frac{a}{x+y} - \frac{.5}{x^2}$ | 2.9) $2(xy^{-1})$ |
| 2.10) $-x^2$ | 2.11) $a^{x+y}+3.5$ | 2.12) $y+a^x$ |
| 2.13) $\frac{y-y_0}{y_1-y_0} \frac{x-x_0}{x_1-x_0}$ | 2.14) $z^{-1} + \frac{1}{2 + \frac{3}{1-x}}$ | 2.15) $\frac{-b + \sqrt{b^2-4ac}}{2a}$ |
| 2.16) u^{2n} | 2.17) $\sqrt{\frac{4(x_1^2+x_2^2+x_3^2)-(x_1+x_2+x_3)^2}{3(3-1)}}$ | |

3. จงพิจารณาคำสั่ง WRITE และคำสั่ง FORMAT ของมันว่าคำสั่งใดผิด จงให้เหตุผลด้วย

(ถ้า 6=หมายเลขแทนหน่วยนำข้อมูลออก)

- | | |
|------------------------------|--------------------------|
| 3.1) WRITE(6,10) , A,B,C | 3.2) WRITE(6,5)I,J |
| 10 FORMAT(1X,F1.0,F2.0,F3.0) | 5 FORMAT(1X,I4) |
| 3.3) WRITE(6,11)4,5.23 | 3.4) WRITE(6,16)A,I,J |
| 11 FORMAT(T5,I2,F4.2) | 16 FORMAT(T3,F4.5,I2,I1) |

3.5) **WRITE(6,7)A,K,C**

7 **FORMAT(F5.2,I3,F4.)**

3.7) **WRITE(6,15)I,X,K**

15 **FORMAT(1X,I2,I3,I1)**

3.6) **WRITE(6,12.)X,Y**

12 **FORMAT(2X,F3.1,F3.1)**

3.8) **WRITE(6,11)X-Y**

11 **FORMAT(F3.2,F4.0)**

4. ถ้า $A=3.$, $B=-2.$, $I=6$ และ $J=0$ จงหาค่านิพจน์ต่อไปนี้

4.1) $A**2+B$

4.2) $I+2/3$

4.3) $A**B$

4.4) $A*3.+B*4$

4.5) A/B

4.6) $A/B*3+A$

4.7) $A/B/2$

4.8) $A/B+2.$

4.9) J/I

4.10) I/J

4.11) $A**I$

4.12) $(A+I)/B$

4.13) $A**2**3$

4.14) $B**B$

4.15) $J**B$

5. จงบอกค่าที่จะเก็บในตัวแปร X หรือ IX จากคำสั่งต่อไปนี้ ใช้ค่า $A=3.2$, $B=-2.$, $I=6$, และ $J=0$

5.1) $X=I$

5.2) $IX=A$

5.3) $X=(I+3)/2$

5.4) $IX=-A+B$

5.5) $X=I**B$

5.6) $X=J*I/.1$

5.7) $X=J$

5.8) $X=B**J$

5.9) $IX=J*A$

6. จากส่วนของโปรแกรมจงบอกค่าที่เก็บใน S, J และ JK

I=4

A=1

B=2

S=(3/I)*3

J=(3./9)*3

JK=(A+2./B)/2

7. แสดงการพิมพ์ค่า A เมื่อกำหนด format code ให้

7.1) **A=743.25**

F10.3

7.2) **A=328.74**

F5.2

7.3) **A=-643.281**

F7.2

7.4) **A=.37**

F5.2

7.5) A=-4768.6 F6.0

8. แสดงผลการทำงานของคำสั่งของโปรแกรมต่อไปนี้

8.1) X=3.2

Y=X*.16

WRITE(6,10)X,Y

10 FORMAT(3X,F4.0,T10,F9.2,'ALL')

8.2) I=+1632

J=-4

K=I/J

WRITE(6,11)I,J,K

11 FORMAT(T8,I4,3X,I1,'+',I5)

8.3) XX=4.3257

YY=-.0008

ZZ=XX+YY

WRITE(6,12)XX,YY,ZZ

12 FORMAT(T4,F7.3,F7.3,F7.3)

8.4) ABC=19.2

IJ3=4

WRITE(6,13) IJ3,ABC

13 FORMAT(2X,I4,3X,F6.0)

8.5) I=.6

J=.6

Z=I+J

WRITE(6,11)Z

11 FORMAT(1X,F2.0,' I2,F4.1')

8.6) I=11

WRITE(6,12)I

12 FORMAT(I2)

WRITE(6,13)

13 FORMAT('1','1','ALL')

9. อุณหภูมิค่าสุดที่เคยวัดได้ทั่วโลกเหนือคือ -126.9°F จงเขียนโปรแกรมเพื่อแปลงอุณหภูมินี้เป็นองศาเซลเซียสโดยใช้สูตร $C = \frac{5}{9}(F-32)$

10. จงเขียนโปรแกรมหนึ่งโปรแกรมเพื่อหาค่าของสูตรแต่ละสูตร (กำหนด $\pi = 3.1416$) และพิมพ์คำตอบและคำอธิบายค่าด้วย

1) คอกเบี้ย $i = prt$ เมื่อ $r = .04$, $t = 3$, $p = 100$

2) ปริมาตรของรูปลูกบาศก์ $v = c^3$ เมื่อ $c = 3.1672$

3) พื้นที่ของวงกลม $A = \pi r^2$ เมื่อ $r = 6.2$

4) ปริมาตรของกรวย $v = \frac{1}{3}\pi r^2 h$ เมื่อ $r = 9.1$ และ $h = 4.932$

11. จงเขียนส่วนของโปรแกรมเพื่อแลกเปลี่ยนค่าของ S และ T ในหน่วยความจำหลัก

12. จงเขียนโปรแกรมเพื่อคำนวณหาความยาวของด้านตรงข้ามมุมฉากของสามเหลี่ยมมุมฉาก โดยกำหนดความยาวของด้านอีก 2 ด้านคือ $A1 = .0056$ และ $A2 = 135.77$