

344-111 ชุดวิชาการโปรแกรมและขั้นตอนวิธี

ตัวดำเนินการและนิพจน์



ตัวดำเนินการและนิพจน์

- ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic operators)
- ตัวดำเนินการยูนารี (Increment and Decrement Operators)
- ตัวดำเนินการเชิงเปรียบเทียบ (Relational operators)
- ตัวดำเนินการเชิงตรรกะ (Logical operators)
- ตัวดำเนินการกำหนดค่า (Assignment operators)
- ตัวดำเนินการเงื่อนไข (Conditional operators)
- นิพจน์ (expression)
- ขั้นตอนการทำงานของนิพจน์
- Examples



ตัวดำเนินการทางคณิตศาสตร์

ตัวดำเนินการ	ความหมาย	ตัวอย่าง
+	บวก	$A+B$
-	ลบ	$A-B$
*	คูณ	$A*B$
/	หาร	A/B
%	การหารเอาแต่เศษไว้ (modulus)	$A\%B$

ตัวอย่าง

นิพจน์คณิตศาสตร์

นิพจน์คณิตศาสตร์ในภาษาซี

$$\frac{a + b}{cd}$$

$$(a+b)/(c*d)$$

$$10x - 3xy + 10y^2$$

$$10*x - 3*x*y + 10*y*y$$



ตัวดำเนินการทางคณิตศาสตร์

- กำหนดตัวแปร A และ B เป็นจำนวนเต็ม

```
int A = 10, B = 3;
```

นิพจน์	ค่า
A+B	13
A-B	7
A*B	30
A/B	3
A%B	1



ตัวดำเนินการยูนารี

ตัวดำเนินการ	ความหมาย	ตัวอย่าง
--	การลดค่าลง ครั้งละ 1	$A--$ (เหมือนกับ $A=A-1$)
++	การเพิ่มค่าขึ้นครั้งละ 1	$A++$ (เหมือนกับ $A=A+1$)

ตัวอย่าง 1

```
int x;  
x = 10;  
printf("x = %d",x);  
x--;  
printf("x = %d",x);  
x++;  
printf("x = %d",x);
```

ผลลัพธ์ ??????



ตัวดำเนินการยูนารี

ตัวอย่าง 2

```
int i = 1;  
printf("i = %d\n",i);  
printf("i = %d\n",++i);  
printf("i = %d\n",i);
```

++ i คือ บวกก่อนแล้วจึงพิมพ์ค่า

ตัวอย่าง 3

```
int i = 1;  
printf("i = %d\n",i);  
printf("i = %d\n", i++);  
printf("i = %d\n",i);
```

i ++ คือ พิมพ์ค่าก่อนแล้วจึงบวก



ตัวดำเนินการเชิงเปรียบเทียบ

ตัวดำเนินการ	ความหมาย	ตัวอย่าง
>	มากกว่า	$A > B$ (A มากกว่า B)
<	น้อยกว่า	$A > B$ (A น้อยกว่า B)
\geq	มากกว่า หรือ เท่ากับ	$A \geq B$ (A มากกว่าหรือเท่ากับ B)
\leq	น้อยกว่า หรือ เท่ากับ	$A \leq B$ (A น้อยกว่าหรือเท่ากับ B)
$=$	เท่ากับ	$A == B$ (A เท่ากับ B)
$!=$	ไม่เท่ากับ	$A != B$ (A ไม่เท่ากับ B)

หมายเหตุ เรายังใช้ตัวดำเนินการทั้ง 6 ตัวนี้เพื่อสร้างเป็นนิพจน์ทางตรรกะซึ่งใช้แทนเงื่อนไขเป็นจริงหรือเท็จ ผลลัพธ์ที่ได้จากนิพจน์เหล่านี้ จะมีประเภทข้อมูลเป็นจำนวนเต็ม โดยจะแทนจริง (true) ด้วย 1 และเท็จ (false) ด้วย 0



ตัวดำเนินการเชิงเปรียบเทียบ

■ int i=1, j=2, k=3;

นิพจน์	การแปลความหมาย	ค่า
$i < j$	จริง	1
$(i+j) \geq k$	จริง	1
$(i+k) > (i+5)$	เท็จ	0
$k \neq 3$	เท็จ	0
$j == 2$	จริง	1



ตัวดำเนินการเชิงตรรกศาสตร์

ตัวดำเนินการ

&&

||

!

ความหมาย

และ (and)

หรือ (or)

นิเสธ (not)

P	Q	P && Q	P Q
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

P	!P
1	0
0	1



ตัวดำเนินการเชิงตรรกية

ตัวอย่าง int i = 7;
 float f = 5.5;
 char c = 'w';

นิพจน์	การแปลความหมาย	ค่า
$(i >= 6) \&\& (c == 'w')$	จริง	1
$(i >= 6) \parallel (c == 119)$	จริง	1
$(f < 11) \&\& (i > 100)$	เท็จ	0
$(c != 'p') \parallel ((i + f) <= 10)$	จริง	1
$!(f < 5)$	จริง	1

ค่าเหล่านี้เรายังสามารถนำไปใช้คำนวณต่อได้





ตัวดำเนินการกำหนดค่า

คือ เครื่องหมาย $=, +, -, *, /, \% =$

รูปแบบ 1

identifier = expression

โดยที่ identifier โดยทั่วไปจะเป็นตัวแปร

expression เป็นค่าคงที่ ตัวแปรหรือนิพจน์ที่มีความซับซ้อนได้

ตัวอย่าง

```
a = 4;  
x = y;  
alpha = 0.001;  
sum = a + b;  
area = length * width;
```

ข้อควรจำ !
= และ ==
มีความแตกต่างกันโดยสิ้นเชิง



ตัวดำเนินการกำหนดค่า

- ในกรณีที่ตัวดำเนินการทั้งสองข้างในนิพจน์มีการกำหนดค่าประเภทข้อมูลที่ต่างกัน ค่าของนิพจน์ที่อยู่ทางขวาเมื่อ จะถูกแปลงให้เป็นประเภทเดียวกันกับตัวแปรที่อยู่ทางซ้ายเมื่ออัตโนมัติ ซึ่งจะมีผลกับค่าของข้อมูล เช่น เมื่อมีการกำหนดค่าให้กับตัวแปรที่เป็นจำนวนเต็ม จะมีการตัดส่วนที่เป็นเศษนิยมของเลขเศษนิยมทิ้ง
- ตัวอย่าง

int i;

<u>นิพจน์</u>	<u>ค่าของ i</u>
$i = 3.3;$	3
$i = 3.9;$	3
$i = -3.9;$	-3
$i = 9/2;$	4



ตัวดำเนินการกำหนดค่า

ตัวอย่าง

```
int i;
```

```
int j = 5;
```

นิพจน์

i = j;

i = j / 2;

i = 2 * j / 2;

i = 2 * (j / 2);

ค่าของ i

5

2

5 (ทำงานจากซ้ายไปขวา)

4 (หารตัวเศษทิ้ง แล้วจึงคูณ)



ตัวดำเนินการกำหนดค่า

ตัวอย่าง

```
float i;  
float j=2;  
int k = 5;
```

!!! การประกาศชนิดของตัวแปรอย่างไม่ระมัดระวัง
มักจะเป็นต้นเหตุของข้อผิดพลาดสำหรับนักเขียน
โปรแกรมระดับเริ่มต้น

นิพจน์

i = k;

i = 15 / j;

i = k / j;

i = k / 2;

i = k / 2.0;

ค่าของ i

5.000000

7.500000

2.500000

2.000000

2.500000



ตัวดำเนินการกำหนดค่า

ตัวอย่าง

สมมติ i เป็นจำนวนเต็ม และใช้กับชุดอักษรและสกุล

int i;

<u>นิพจน์</u>	<u>ค่าของ i</u>
i = 'X';	120
i = 'A';	65
i = ('X' - 'A')/8;	6
i = 'X' - 'A'/5;	107

ลองดู !!!

```
char i, j, k;  
i = 'X';  
j = 'A';  
k = 'X' + 'A';  
ค่าของ k = ?
```



ตัวดำเนินการกำหนดค่า

รูปแบบ 2

`identifier1 = identifier2 = ... = expression`

- ชี้การกำหนดค่าลักษณะนี้จะทำงานจากขวาไปซ้าย
- ดังนั้นการกำหนด `identifier1 = identifier2 = expression`
จึงมีค่าเท่ากับ `identifier1 = (identifier2 = expression)`

ตัวอย่าง

`i=j=5;`



ตัวดำเนินการกำหนดค่า

รูปแบบ 3

`expression1 += expression2`

- การกำหนดค่า เช่นนี้มีค่าเท่ากับ `expression1 = expression1 + expression3`
- ตัวอย่าง

`a= 4;`

`a += 5;`

หลังจากทำงานเสร็จจะได้ a มีค่าเท่ากับ 9;

หมายเหตุ โดยทั่วไป `expression1` จะเป็นตัวแปร



ตัวดำเนินการกำหนดค่า

- ในทำงานเดียวกัน
- รูปแบบ 4

`expression1 -= expression2`

- จะมีค่าเท่ากับ `expression1 = expression2 - expression3`

ตัวดำเนินการที่เหลืออีก 3 ตัว คือ *=, /= และ %=
ก็มีการทำงานในลักษณะเดียวกัน



ตัวดำเนินการกำหนดค่า

ตัวอย่าง

```
int i=5, j=7;  
float f=5.5, g=-3.25;
```

นิพจน์	นิพจน์ที่เทียบเท่ากัน	ค่า
$i += 5$	$i = i + 5$	10
$f -= g$	$f = f - g$	8.75
$j *= (i-3)$	$j = j * (i-3)$	14
$f /= 3$	$f = f/3$	1.833333
$i %= (j-2)$	$i = i\%(j-2)$	0



ตัวดำเนินการเงื่อนไข

รูปแบบ

expression1 ? expression2 : expression3

ตัวอย่าง 1

(i<0) ? 0:100;

หมายความว่า ถ้า i น้อยกว่า 0 นิพจน์นี้จะมีค่าเป็น 0

ถ้า i มีค่ามากกว่า หรือ เท่ากับ 0 นิพจน์นี้มีค่าเป็น 100

ตัวอย่าง 2

max = (a>=b) ? a:b;

หมายความว่า ถ้า a มากกว่าหรือเท่ากับ b และ max มีค่าเท่ากับ a

ถ้า a น้อยกว่า b และ max มีค่าเท่ากับ b



กฎในการเขียนนิพจน์

- ห้ามเขียนตัวแปร 2 ตัวติดกันโดยไม่มีเครื่องหมาย เช่น
 - ab ในภาษาซีต้องเขียนเป็น a*b
 - การเขียน ab ถือว่าเป็นตัวแปรเดียว คือชื่อตัวแปร ab "ไม่ใช่ค่า a คูณ b"
- ถ้าเขียนนิพจน์โดยมีค่าของตัวแปรหรือค่าคงที่ต่างชนิดกันในนิพจน์เดียวกันภาษาซี จะเปลี่ยนชนิดของข้อมูลที่มีขนาดเล็กให้เป็นชนิดของข้อมูลที่ใหญ่ขึ้น เช่น
 - ถ้าใช้ char กับ int จะเปลี่ยนเป็น int
 - ถ้าใช้ int กับ float จะเปลี่ยนเป็น float
 - ถ้าใช้ float กับ double จะเปลี่ยนเป็น double

ตัวอย่าง

```
int a = 15;  
float b = 2;  
sum = a+b;  
mul = a*b;
```

คำถาม: เรายังประกาศตัวแปร sum และ mul เป็นตัวแปรชนิดใด



กฎในการเขียนนิพจน์

การหาร

$$\frac{\text{Integer}}{\text{Integer}} = \text{Integer}$$

ตัวอย่าง

```
int a = 5;  
int b = 2;  
float c = 2;
```

ถ้า a/b จะได้ผลลัพธ์เป็น 2

ถ้า a/c จะได้ผลลัพธ์เป็น 2.500000

$$\frac{\text{Integer}}{\text{Float}} = \text{Float}$$

$$\frac{\text{Float}}{\text{Integer}} = \text{Float}$$



ขั้นตอนทำงานของนิพจน์

- นิพจน์ในภาษาซี จะทำงานตามลำดับการทำงานของตัวดำเนินการต่อไปนี้
- โดยเรียงจากการทำงานอันดับที่ 1 ลงไปถึงอันดับสุดท้าย คือ

(1)	0	ชี้ไปขวา
(2)	!, ++, --	ขวาไปซ้าย
(3)	*	ชี้ไปขวา
(4)	/, %	ชี้ไปขวา
(5)	+, -	ชี้ไปขวา
(6)	<, <=, >, >=	ชี้ไปขวา
(7)	==, !=	ชี้ไปขวา
(8)	&	ชี้ไปขวา
(9)		ชี้ไปขวา
(10)	? :	ขวาไปซ้าย
	=, +=, -=, *=, /=, %=	ขวาไปซ้าย



ขั้นตอนการทำงานของนิพจน์

ตัวอย่าง 1

$$\underbrace{a * b}_{\textcircled{1}} \quad / \quad c$$
$$\qquad \qquad \qquad \underbrace{\qquad}_{\textcircled{2}}$$

ตัวอย่าง 2

$$a * b + c / d - e$$

$$\underbrace{\qquad}_{\textcircled{1}} \quad \underbrace{\qquad}_{\textcircled{2}}$$

$$\qquad \qquad \qquad \underbrace{\qquad}_{\textcircled{3}}$$

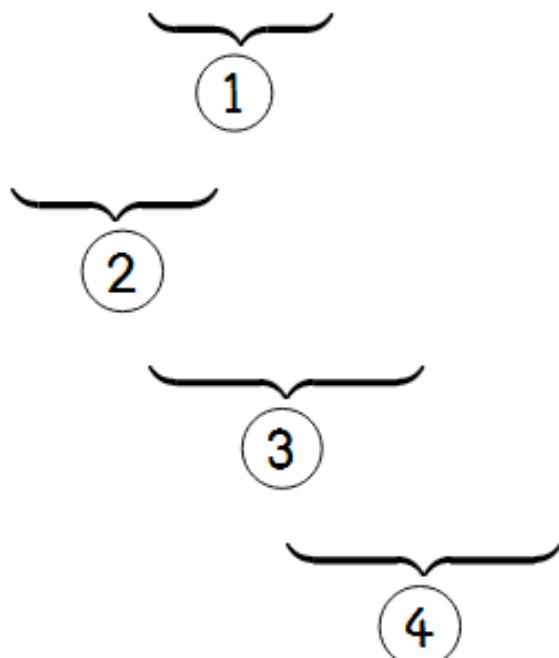
$$\qquad \qquad \qquad \underbrace{\qquad}_{\textcircled{4}}$$



ขั้นตอนการทำงานของนิพจน์

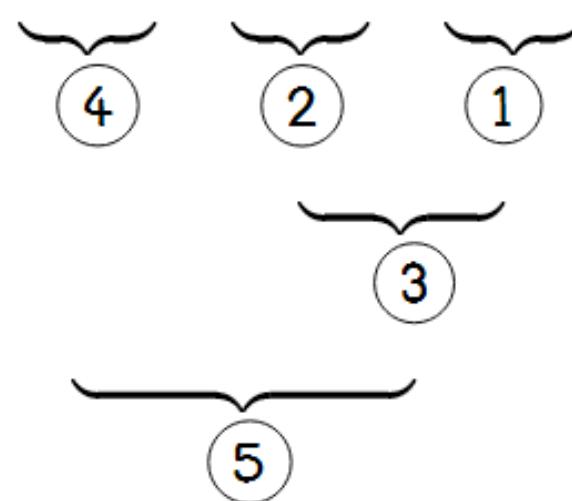
ตัวอย่าง 3

$$a * (b + c) / d - e$$



ตัวอย่าง 4

$$a - b + c * d / e --$$

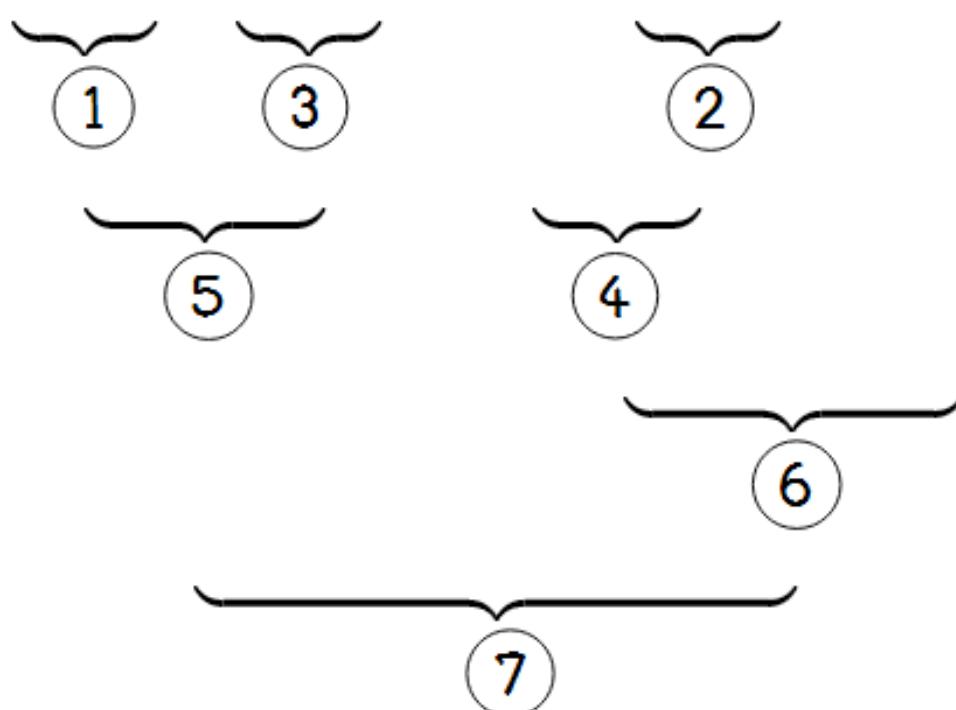




ขั้นตอนการทำงานของนิพจน์

ตัวอย่าง 5

$b * c > 5 - d \&& e - f * g == 10$

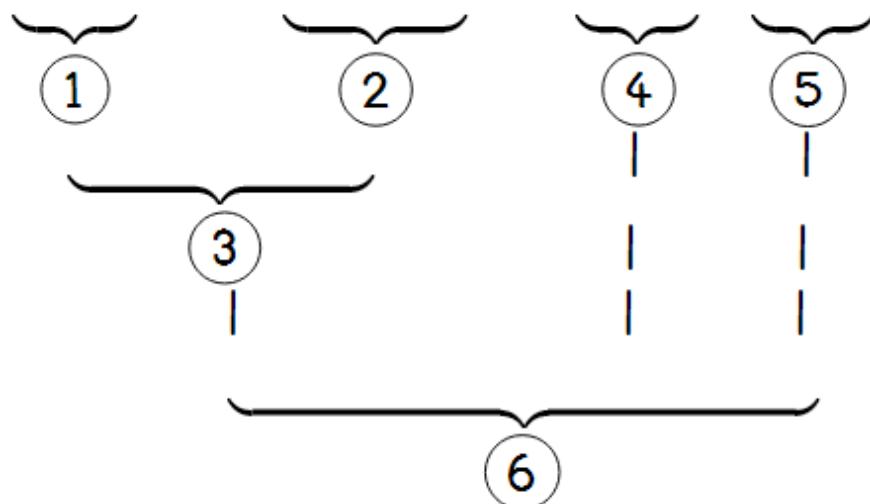




ขั้นตอนการทำงานของนิพจน์

ตัวอย่าง 6

`c += (a > 0 && a <= 10) ? ++a : a / b`



ถ้าสมมุติให้

$a = 1, b = 2, c = 3$

หลังจบการทำงานของนิพจน์นี้แล้วจะได้

$c = ???$

ถ้าสมมุติให้

$a = 50, b = 10, c = 20$

หลังจบการทำงานของนิพจน์นี้แล้วจะได้

$c = ???$



Exercise 1

- What is the output

```
#include<stdio.h>
int main()
{
    int i = 5;
    int a = ++i + ++i;
    printf("%d",a);
    return 0;
}
```



Exercise 2

- What is a output

```
#include<stdio.h>
int main(){
    int i = 16;
    i =! i > 15;
    printf("i = %d",i);
    return 0;
}
```