ตัวดำเนินการและนิพจน์ทางคณิตศาสตร์

การนำเครื่องคอมพิวเตอร์มาช่วยในการทำงานของมนุษย์ โดยการเขียนโปรแกรมสั่งให้เครื่องคอมพิวเตอร์ทำงานทุกภาษารวมทั้งภาษาซี การคำนวณทางคณิตศาสตร์ เป็นสิ่งจำเป็นต่อการทำงานของคอมพิวเตอร์ ที่ต้องทำการคำนวณอย่างถูกต้องและรวดเร็ว ตัวดำเนินการทางคณิตศาสตร์และนิพจน์ทางคณิตศาสตร์ รวมทั้งเครื่องหมายการกระทำการต่างๆ เป็นสิ่งสำคัญในการทำให้การเขียนโปรแกรมคอมพิวเตอร์สามารถแก้ปัญหาโจทย์ได้สำเร็จ ซึ่งในภาษาซีได้มีตัวดำเนินการดังนี้ ตัวดำเนินการกำหนดค่า (assignment operator) ตัวดำเนินการคณิตศาสตร์ (arithmetic operators) ตัวดำเนินการเปลี่ยนชนิดข้อมูล (type cast operator) ตัวดำเนินการเชิงสัมพันธ์ (relational operator) และตัวดำเนินการเชิงตรรกะ (logical operator)

ในการทำงาน ข้อมูลที่กระทำกับตัวดำเนินการ ถูกเรียกว่าตัวถูกดำเนินการ (operand) ในการทำงานของภาษาซี การกระทำของตัวดำเนินการมี 2 ลักษณะด้วยกันคือ มีตัวถูกดำเนินการ 2 ตัว กับมีตัวถูกดำเนินการ 1 ตัว

ตัวดำเนินการกำหนดค่า

ภาษาซีตัวดำเนินการ (Steven และ Lutfar, 2006) (Brian W. K., Online) ในการกำหนดค่าสามารถทำได้ด้วยเครื่องหมายเท่ากับ (=) โดยการกำหนดค่าทางขวามือ (expression) ของเครื่องหมายเท่ากับ ไปเก็บยังทางซ้ายมือ (Identifier) ของเครื่องหมายเท่ากับ ดังรูปแบบคือ

```
Identifier = expression
```

โดยทั่วไป Identifier เป็นตัวแปรที่รับค่าที่ถูกกำหนดจาก expression ที่เป็นค่าคงที่หรือเป็นนิพจน์ที่มีความซับซ้อน เช่น

a = 10 //การกำหนดให้ค่าตัวแปร a มีค่าเป็นจำนวนเต็มเท่ากับ 10

result = 0.99 // การกำหนดให้ค่าตัวแปร result มีค่าทศนิยมเท่ากับ 0.99

r = s //การกำหนดให้ค่าในตัวแปร s ไปเก็บในตัวแปร r

total = receive * 0.7 //การกำหนดให้ค่าตัวแปร total มีค่าเท่ากับผลการคำนวณกันคูณระหว่างตัวแปร receive กับค่าคงที่ทศนิยม 0.7

square = (base * height) * 0.5 //การกำหนดให้ค่าตัวแปร square มีค่าเท่ากับผลการคำนวณกันของตัวแปร base คูณกับตัวแปร height แล้วผลคูณทำการคูณกับค่าค่าคงที่ 0.5

ข้อสังเกต ตัวดำเนินการกำหนดค่าใช้เครื่องหมาย = ส่วนการเปรียบเทียบค่าระหว่างสองค่าใช้เครื่องหมาย = = เป็นการตรวจสอบค่าด้านซ้ายมือของเครื่องหมาย = = กับค่าด้านขวามือของเครื่องหมายว่ามีค่าเท่ากันหรือไม่ ซึ่งให้ผลการแปลความหมายของภาษาซีระหว่างเครื่องหมาย = กับ = = แตกต่างกันอย่างสิ้นเชิง

การเขียนตัวดำเนินการกำหนดค่าแบบลัด

การกำหนดค่าให้กับตัวแปรที่อยู่ด้านซ้ายของตัวดำเนินการกำหนดค่า (=) ด้วยนิพจน์ด้านขวา ที่มีตัวแปรทางด้านซ้ายอยู่ด้วย โดยกำหนดเป็นรูปแบบดังนี้คือ

ตัวแปรด้านซ้าย = นิพจน์ทางด้านขวาที่มีตัวแปรด้านซ้ายอยู่ด้วย

ตัวดำเนินการแบบลัด มีดังนี้ +=, -=, *=, /=, %=, >>=, <<=, &=, |= และ ^=

เช่น	var = var+2	กำหนดนิพจน์แบบทางลัดได้เป็น	var/=2
	i = i +1	กำหนดนิพจน์แบบทางลัดได้เป็น	i+=1
	j = j>>1	กำหนดนิพจน์แบบทางลัดได้เป็น	j>>=1

ตัวดำเนินการทางคณิตศาสตร์

ภาษาซีมีเครื่องหมายทางคณิตศาสตร์ 5 ชนิดดังที่แสดงในตาราง 3.1

ตารางที่ 3.1 ตัวกระทำทางคณิตศาสตร์

ตัวดำเนินการทางคณิตศาสตร์	ความหมาย
+	บวก
-	ลบ
*	คูณ
/	หาร
%	หารคิดเฉพาะเศษ

ในการคำนวณทางคณิตศาสตร์ของตัวดำเนินการทางคณิตศาสตร์ของเครื่อง บวก ลบ คูณ และหาร เป็นการกระทำทางคณิตศาสตร์ทั่วไป ยกเว้นเครื่องหมาย % เป็นตัวดำเนินการ

ทางคณิตศาสตร์ที่เรียกว่า มอดูลัส (modulus operator) ภาษาซีตีความหมายของการมอดูลัส เป็นการหารคิดเศษ หมายความว่า การให้ผลการหารของเลขสองจำนวนเป็นค่าของเศษของการหาร เช่น 10 % 3 ผลลัพธ์ได้เท่ากับ 1 เพราะ 10 หารด้วย 3 ผลการหารได้ 3 โดยได้ผลมอดูลัสที่นำผลเศษมาตอบเป็น 1 ดังตัวอย่างต่อไป

ตัวอย่างที่ 3.1 กำหนดให้ตัวแปร x เป็นตัวแปรจำนวนเต็มมีค่าเท่ากับ 9 และตัวแปร y เป็นตัวแปรจำนวนเต็มมีค่าเท่ากับ 2 มีการดำเนินการทางคณิตศาสตร์ และมีผลการการดำเนินการดังนี้

นิพจน์	ผลการดำเนินการ
x + y	11
x – y	7
x * y	18
x / y	4.5
x % y	1

กำหนดให้ตัวแปร x1 เป็นตัวแปรทศนิยมมีค่าเท่ากับ 15.5 และตัวแปร x2 เป็นตัวแปรทศนิยมมีค่าเท่ากับ 2.0 มีการดำเนินการทางคณิตศาสตร์ และมีผลการการดำเนินการดังนี้

นิพจน์	ผลการดำเนินการ
x1 + x2	17.5
x1 - x2	13.5
x1 * x2	31
x1 / x2	7.75

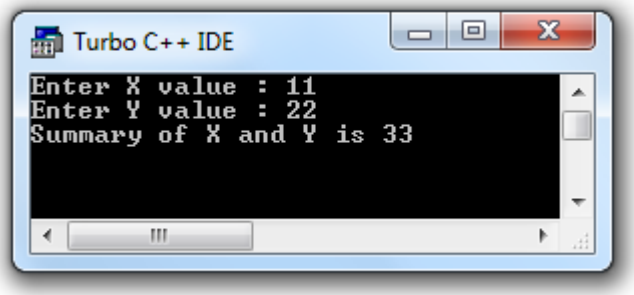
กำหนดให้ตัวแปร y1 และ y2 เป็นตัวแปรอักขระเก็บค่า S ตามลำดับและ K ตามลำดับ (ใช้ตารางที่ 2.1 แสดงชุดตัวอักขระแอสกี [เลขฐาน 10] : S = 83, K = 75, 9 = 57) โดยทางผลการดำเนินของตัวแปรทั้งสอง แสดงดังนี้

นิพจน์	ผลการดำเนินการ
y1	83
y1 + y2	83 + 75 = 158
y1 + y2 +2	83 + 75 + 2 = 160
y1 + y2 + '9'	83 + 75 + 57 = 217

ตัวอย่างโปรแกรมภาษาซี

```
#include <stdio.h>
void main( )
{
        int x, y, z;
        printf("Enter X value : ");
        scanf("%d", &x);
        printf("Enter Y value : ");
        scanf("%d", &y);
        z = x + y;
        printf("Summary of X and Y is %d", z);
        getch();
}
```

ผลการทำงานของโปรแกรม ดังนี้



ตัวอย่างที่ 3.2 กำหนดให้ตัวแปร r เป็นตัวแปรจำนวนเต็มมีค่าเท่ากับ 7 ตัวแปร s เป็น ตัวแปรทศนิยมมีค่าเท่ากับ 5.5 และตัวแปร t เป็นตัวแปรอักขระมีค่าเท่ากับ w (ตัวอักขระแอสกี [เลขฐาน 10]: w = 119) การดำเนินการทางคณิตศาสตร์ และมีผลการดำเนินการดังนี้

นิพจน์	ผลการดำเนินการ	ชนิดข้อมูล
r + s	7 + 5.5 = 12.5	double
r – t	7 + 119 = 126	int
r + t – ‘0’	7 + 119 – 48 = 78	int
(r + t) – (2 * s / 5)	(7 +119) – (2 * 5.5 / 5) = 123.8	double

ตัวดำเนินการเปลี่ยนชนิดข้อมูล (Type Cast Operator)

ในการดำเนินการทางคณิตศาสตร์ของข้อมูลชนิดตัวเลขหรือชนิดตัวอักขระ ในบางกรณีอาจต้องการทำการเปลี่ยนชนิดข้อมูล เพื่อความถูกต้องของนิพจน์ในการเขียนโปรแกรมในภาษาซีสามารถทำได้โดยการชื่อชนิดของข้อมูลที่ต้องการเปลี่ยนไว้ในวงเล็บ ก่อนหน้านิพจน์หรือตัวแปร (Alexander, A. ,Online) ดังรูปแบบด้านล่าง

(data type) expression

โดย data type คือ ชนิดข้อมูลที่ต้องการเปลี่ยน, expression คือ นิพจน์หรือตัวแปรที่เปลี่ยน

ตัวอย่างที่ 3.3 กำหนดให้ตัวแปร s เป็นตัวแปรทศนิยมมีค่าเท่ากับ 5.5 ต้องการให้การดำเนินการทางคณิตศาสตร์ที่ถูกต้อง

((int s) % 2)

เป็นการเปลี่ยนชนิดข้อมูล s ให้เป็นชนิด int ได้ด้วยคำสั่ง (int s) มีผลทำให้ข้อมูลในตัวแปร s มีค่าเป็นจำนวนเต็มคือ 5 และนำ 2 มาทำการมอดุลัส การดำเนินการทางคณิตศาสตร์ ได้ผลลัพธ์เป็น 1

ตัวอย่างที่ 3.4 กำหนดให้ตัวแปร r เป็นตัวแปรจำนวนเต็มมีค่าเท่ากับ 7 ตัวแปร s เป็นตัวแปรทศนิยมมีค่าเท่ากับ 8.5 นิพจน์คณิตศาสตร์ดังนี้

(r + s) % 2

เป็นนิพจน์ที่ผิดด้วย (r + s) เมื่อมีผลการดำเนินทางคณิตศาสตร์ จะให้ผลของชนิดข้อมูลเป็นทศนิยมไม่ใช่จำนวนเต็ม ดังนั้นนำมา 2 มามอดุลัส ไม่สามารถดำเนินการได้ จึงต้องทำการเปลี่ยนนิพจน์ให้เป็นชนิดจำนวนเต็มก่อนได้ดังนี้

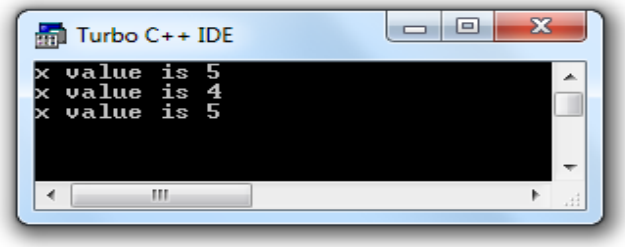
((int) (r + s)) % 2

นิพจน์ (r + s) ต้องทำการเปลี่ยนชนิดข้อมูล ให้เป็นชนิด int ได้ด้วยคำสั่ง (int) (r + s) มีผลทำให้ข้อมูลในผลการดำเนินคณิตศาสตร์ (int) (r + s) มีค่าเป็นจำนวนเต็มคือ 15 และนำ 2 มาทำการมอดูลัส การดำเนินการทางคณิตศาสตร์ ได้ผลลัพธ์เป็น 1

ตัวอย่างโปรแกรมภาษาซี

```
#include<stdio.h>
#include<conio.h>
void main( )
{
        int x;
        clrscr();
        x = 2.5 * 2;
        printf("x value is %d", x);
        x = (int)2.5 * 2;
        printf("\nx value is %d", x);
        x = (int)(2.5 * 2);
        printf("\nx value is %d", x);
        getch();
}
```

ผลการทำงานของโปรแกรม



ลำดับการประมวล

ในนิพจน์ทางคณิตศาสตร์ลำดับการประมวลผลมีความสำคัญต่อความถูกต้องของการคำนวณ โดยนิพจน์ที่มีวงเล็บในนิพจน์ จะทำการประมวลผลในวงเล็บเป็นลำดับแรก มีวงเล็บหลายวงเล็บจะทำวงเล็บในเป็นลำดับแรก ในลำดับถัดมาทำการประมวลผลจากเรียงจากซ้ายไปขวาของนิพจน์ และในเครื่องหมายทางคณิตศาสตร์ จะมีลำดับการประมวลผลดังตารางที 3.2

ตารางที่ 3.2 ลำดับการประมวลผล

ลำดับการประมวลผล	เครื่องหมายทางคณิตศาสตร์
1	++,--
2	- (เครื่องหมายลบหน้าตัวเลข)
3	* / %
4	+ -

ตัวอย่างที่ 3.5

(2+3)*5 = 25 // 2 บวก 3 ได้ 5 แล้วนำไปคูณกับ 5 ได้ผลลัพธ์ 25

2+3*5 = 17 // 3 คูณ 5 ได้ 15 แล้วบวกกับ 2 ได้ผลลัพธ์ 17

(5+2)*15%4 = 1 // 5 บวก 2 ได้ 7 แล้วคูณด้วย 15 ได้ 105 หาร 4 ได้ผลลัพธ์เหลือเศษ = 1

การมอดูลัส (Module : %) หรือ การหารคิดเศษ จะสามารถใช้ได้กับตัวแปรจำนวนเต็มเท่านั้น แต่ถ้าต้องการใช้กับตัวแปรที่ไม่เป็นจำนวนเต็มจะต้องแปลงให้เป็นจำนวนเต็ม

การกระทำทางคณิตศาสตร์ระหว่างตัวแปรที่ต่างชนิดกันจะทำให้เกิดการเปลี่ยนชนิดของตัวแปรซึ่งโดยทั่วไปถ้าชนิดตัวแปรมีขนาดเล็กกว่าจะถูกเปลี่ยนเป็นชนิดของตัวแปรที่มีขนาดใหญ่กว่าดังตารางที่ 3.3

ตารางที่ 3.3 ขนาดของชนิดข้อมูล

ขนาดของชนิดข้อมูล	ชนิดข้อมูล
ใหญ่	long double
	double
	float
	unsigned long
	long
เล็ก	unsigned

ตัวดำเนินการเชิงสัมพันธ์ และตัวดำเนินการเชิงตรรกะ

ตัวดำเนินการเชิงสัมพันธ์ของภาษาซีมี 4 ตัวดำเนินการ ตัวดำเนินการทั้งสี่ตัวมีลำดับความสำคัญเท่าๆ กัน แต่ลำดับความสำคัญต่ำกว่า ตัวดำเนินการทางคณิตศาสตร์ การทำงานของกลุ่มตัวดำเนินการเชิงสัมพันธ์จะมีลำดับการทำงานจากซ้ายไปขวาตามลำดับของตัวดำเนินการสเชิงสัมพันธ์ ตัวดำเนินการเชิงสัมพันธ์ ดังแสดงตารางที่ 3.4

ตารางที่ 3.4 ตัวดำเนินการเชิงสัมพันธ์

ตัวดำเนินการ	ความหมาย
>	มากกว่า
>=	มากกว่าหรือเท่ากับ
<	น้อยกว่า
<=	น้อยกว่าหรือเท่ากับ

ตัวดำเนินการที่มีการทำงานคล้ายการกับการทำงานของ ตัวดำเนินการสัมพันธ์ มี 2 ตัวดำเนินการ เรียกว่า ตัวดำเนินการเปรียบเทียบ ดังตารางที่ 3.5

ตารางที่ 3.5 ตัวดำเนินการเปรียบเทียบ

ตัวดำเนินการ	ความหมาย
==	เท่ากับหรือเท่ากัน
!=	ไม่เท่ากับหรือไม่เท่ากัน

ดังนั้นตัวดำเนินการในกลุ่มเชิงสัมพันธ์ มี 6 ตัวดำเนินการ ใช้ในการสร้างนิพจน์ทางตรรกะ ใช้แทนผลการดำเนินนิพจน์ทดสอบเงื่อนไขว่า เป็นจริง (true) แสดงค่าเป็น 1 (หนึ่ง) และถ้าผลการทดสอบเงื่อนไขเป็นเท็จ (false) แสดงค่าเป็น 0 (ศูนย์)

ตัวอย่างที่ 3.6

ในการเขียนโปรแกรมก่อนดำเนินการด้วย ตัวดำเนินการเชิงสัมพันธ์ ต้องกำหนดค่าให้กับตัวแปรก่อนประมวลผลดังนี้ ให้ q = 10, r = 5, s = 0

นิพจน์	การแปลความหมาย	ค่า
s < q – r	จริง	1
q / r < s	เท็จ	0
s + r == q-5	จริง	1
s – r == s + r	เท็จ	0
q >= r * 2	จริง	1
r <= s	เท็จ	0
s – r != s + r	จริง	1

ตัวอย่างที่ 3.7

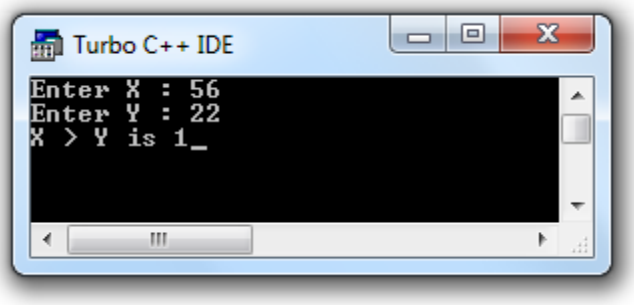
เป็นตัวอย่างที่ทำการเปรียบเทียบ ข้อมูลในตัวแปรที่ต่างชนิดกัน ระหว่างตัวแปรที่เป็นจำนวนเต็ม i มีค่าเท่ากับ 7 ตัวแปรทศนิยม f มีค่าเท่ากับ 5.5 ตัวแปรอักขระ (แอสกี) c เก็บตัวอักขระ ‘z’ โดยในกลุ่มของตัวแปรที่ตัวเลขต้องทำการแปลงชนิดข้อมูลให้มีชนิดเดียวกัน ดังหัวข้อ ตัวดำเนินการเปลี่ยนชนิดข้อมูล ในการเขียนโปรแกรม

นิพจน์	การแปลความหมาย	ค่า
‘A’ == 65	จริง	1
(f + i) <= 5	เท็จ	0
f > 5	จริง	1
c != ‘s’	จริง	1
c == 122	จริง	1

ตัวอย่างโปรแกรมภาษาซี

```
#include<stdio.h>
#include<conio.h>
void main( )
{
        int x, y;
        clrscr();
        printf("Enter X : ");
        scanf("%d", &x);
        printf("Enter Y : ");
        scanf("%d", &y);
        printf("X > Y is %d", x>y);
        getch();
}
```

ผลการทำงานของโปรแกรม



ตัวดำเนินการเชิงตรรกะ (logical operator)

ตัวดำเนินการเชิงตรรกะ เป็นเครื่องหมายตัวกระทำที่ใช้ในการเปรียบเทียบและตัดสินใจโดยเงื่อนไขตั้งแต่ 2 เงื่อนไขขึ้นไปมาเปรียบเทียบกัน ซึ่งผลที่ได้จากการเปรียบเทียบจะได้ผลเป็น 2 กรณี คือ จริง (true) จะให้ค่าเป็น 1 และเท็จ (false) จะให้ค่าเป็น 0 เช่นเดียวกับเครื่องหมายตัวกระทำเปรียบเทียบ เครื่องหมายตัวกระทำทางลอจิก ตัวดำเนินการเชิงตรรกะมีดังนี้

ตารางที่ 3.6 ตัวดำเนินการเชิงตรรกะ

ตัวดำเนินการ	ความหมาย	
&&	AND	และ
||	OR	หรือ
!	NOT	นิเสธ

การดำเนินการเชิงตรรกะ เป็นการนำนิพจน์ที่มีผลเป็นตรรกะ มาเชื่อมกับนิพจน์มีผลตรรกะอีกประโยค เพื่อต้องการทราบผลการดำเนินการนิพจน์ทั้งสองนิพจน์ โดยภาพรวมมีผลทางตรรกะเป็นอย่างไร

ตัวอย่างที่ 3.8

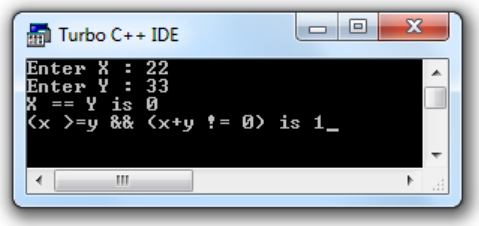
กำหนดให้ i เป็นตัวแปรชนิดตัวเลขจำนวนเต็มมีเท่ากับ 7 ให้ f เป็นตัวแปรชนิดทศนิยมมีค่าเท่ากับ 5.5 และ ให้ c เป็นตัวแปรอักขระเก็บค่า ‘z’ มาพิจารณานิพจน์ต่างๆ ดังนี้

นิพจน์	การแปลความหมาย	ค่า
(f < 11) && (I > 50)	เท็จ	0
(i >= f) && (c == 122)	จริง	1
(f > 15) || (c == ‘z’)	จริง	1
(c < 50) || (i > c)	เท็จ	0
!((f < 11) && (I > 50) **)**	จริง	1
!((i >= f) && (c == 122) **)**	เท็จ	0

ตัวอย่างโปรแกรมภาษาซี

```
#include<stdio.h>
#include<conio.h>
void main( )
{
        int x, y, result1,result2;
        clrscr();
        printf("Enter X : ");
        scanf("%d", &x);
        printf("Enter Y : ");
        scanf("%d", &y);
        result1 = (x==y);
        result2 = (x <=y && (x+y != 0));
        printf("X == Y is %d", result1);
        printf("\n(x <=y && (x+y != 0) is %d", result2);
        getch();
}
```

ผลการทำงานของโปรแกรม



ตัวดำเนินการเงื่อนไข

ตัวดำเนินการเงื่อนไขเป็นตัวดำเนินการที่ใช้ในการตรวจสอบเงื่อนไขของนิพจน์ที่มีรูปแบบ คือ

นิพจน์ที่ 1 ? นิพจน์ที่ 2 : นิพจน์ที่ 3

ในการประมวลผลของภาษาซี ทำการตรวจสอบ นิพจน์ที่ 1 ว่าเป็นจริงหรือไม่
ถ้าหากเป็นจริงตัวโปรแกรมภาษาซี สั่งให้ทำการประมวลผล นิพจน์ที่ 2
แต่หากเป็นเท็จตัวโปรแกรมภาษาซี สั่งให้ทำการประมวลผล นิพจน์ที่ 3

ดังตัวอย่าง

min = (a < b) ? a : b;

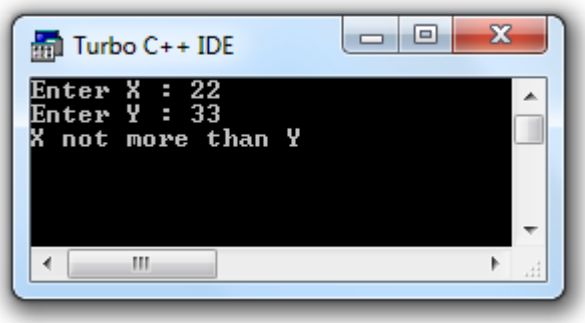
การดำเนินของจะทำการเปรียบเทียบค่า a กับ ค่า b ว่าค่า a มีค่าน้อยกว่า b จริงหรือไม่
ถ้าหากเป็นจริง ค่า min มีค่าเท่ากับ a
แต่หากเป็นเท็จ ค่า min มีค่าเท่ากับ b

ตัวอย่างโปรแกรมภาษาซี

โปรแกรมสั่งให้มีการรับข้อมูลจำนวนเต็ม 2 จำนวนจากผู้ใช้งานกำหนดค่า x กับ y ถ้าหากค่า x มากกว่าให้แสดงข้อความว่า X more than Y แต่ถ้าหากไม่ใช่ให้แสดงข้อความว่า X not more than Y

```
#include<stdio.h>
#include<conio.h>
void main( )
{
        int x, y;
        clrscr();
        printf("Enter X : ");
        scanf("%d", &x);
        printf("Enter Y : ");
        scanf("%d", &y);
        (x > y) ? printf("X more than Y") : printf("X not more than Y");
        getch();
}
```

ผลการทำงานโปรแกรม



ตัวดำเนินการระดับบิต (bit operator)

โปรแกรมภาษาซีมีข้อที่แตกต่างจากภาษาอื่น คือ ความสามารถในการใช้คำสั่งได้ใกล้เคียงกับภาษาแอสเซมบลี ซึ่งสามารถใช้ควบคุมติดต่อกับอุปกรณ์ทางฮาร์ดแวร์ได้ และยังสามารถใช้กับการคำนวณทางคณิตศาสตร์ของเลขฐานสองได้ โดยสามารถทำการทดสอบแต่ละบิตของตัวแปรเลขจำนวนเต็มและอักขระได้ ตัวดำเนินการระดับบิต ดังตารางที่ 3.7

ตารางที่ 3.7 แสดงตัวดำเนินการระดับ

ตัวดำเนินการ	ความหมาย
&	AND
|	OR
^	Exclusive OR
<<	เลื่อนซ้าย (Shift Left)
>>	เลื่อนขวา (Shift Right)
~	(One's complement)

ตัวดำเนินการแอนด์ (&)

การประมวลผลของตัวดำเนินแอนด์ เป็นการดำเนินการคล้ายทางคณิตศาสตร์เครื่องหมายคูณ คือเมื่ออินพุตขาใดขาหนึ่งเป็นศูนย์ ส่งผลทำให้เอาต์พุตเป็นศูนย์ และถ้าอินพุตเป็นหนึ่งทั้งหมด ส่งผลทำให้เอาต์พุตเป็นหนึ่ง ซึ่งจะมีค่าตารางความจริง (true table) ดังตารางที่ 3.8

ตารางที่ 3.8 ตารางความจริงของตัวดำเนินการแอนด์ (&)

อินพุต		เอาต์พุต
A	**B**	**A & B**
0	0	0
0	1	0
1	0	0
1	1	1

ตัวตัวดำเนินการออร์ (|)

การประมวลผลของตัวดำเนินออร์ เป็นการดำเนินการคล้ายทางคณิตศาสตร์เครื่องหมายบวก คือเมื่ออินพุตขาใดขาหนึ่งเป็นหนึ่ง ส่งผลทำให้เอาต์พุตเป็นหนึ่ง และถ้าอินพุตเป็นศูนย์ทั้งหมด ส่งผลทำให้เอาต์พุตเป็นศูนย์ ซึ่งจะมีค่าตารางความจริง (true table) ดังตารางที่ 3.9

ตารางที่ 3.9 ตารางความจริงของตัวดำเนินการออร์ (|)

อินพุต		เอาต์พุต
A	**B**	**A | B**
0	0	0
0	1	1
1	0	1
1	1	1

ตัวดำเนินการเอ็กซ์คลูซีฟออร์ (^)

การประมวลผลของตัวดำเนินเอ็กคลูซีฟออร์ เป็นการดำเนินการใช้ในการเปรียบเทียบข้อมูล คือเมื่ออินพุตเหมือนกันทั้งหมด ส่งผลทำให้เอาต์พุตเป็นศูนย์ และถ้าอินพุตเป็นไม่เหมือนกัน ส่งผลทำให้เอาต์พุตเป็นหนึ่ง ซึ่งจะมีค่าตารางความจริง (true table) ดังตารางที่ 3.10

ตารางที่ 3.10 ตารางความจริงของตัวดำเนินการเอ็กซ์คลูซีฟออร์ (^)

อินพุต		เอาต์พุต
A	**B**	**A ^ B**
0	0	0
0	1	1
1	0	1
1	1	0

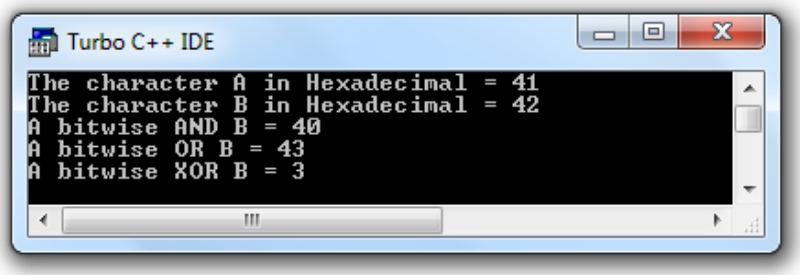
ตัวอย่างโปรแกรมภาษาซี

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int c,d;
        clrscr();
```

ต่อจากด้านบน

```
        c = 'A';  // 41(Hexadicimal) => 0100 0001(Binary)
        d = 'B';  // 42(Hexadicimal> => 0100 0010(binary)
        printf("The character A in Hexadecimal = %x\n",c);
        printf("The character B in Hexadecimal = %x\n",d);
        printf("A bitwise AND B = %x\n",c&d); //0100 0000
        printf("A bitwise OR B = %x\n",c|d); //0100 0011
        printf("A bitwise XOR B = %x\n",c^d); //0000 0011
        getch();
}
```

ผลการทำงานของโปรแกรม



ตัวดำเนินเลื่อน : ซ้าย (<<) และขวา (>>)

ตัวดำเนินการเลื่อน จะใช้ในการเลื่อนบิตของไบนารีจากซ้ายไปขวา (>>) และจากขวาไปซ้าย (>>)

Express_1	<<	Express_2
Express_1	>>	Express_2

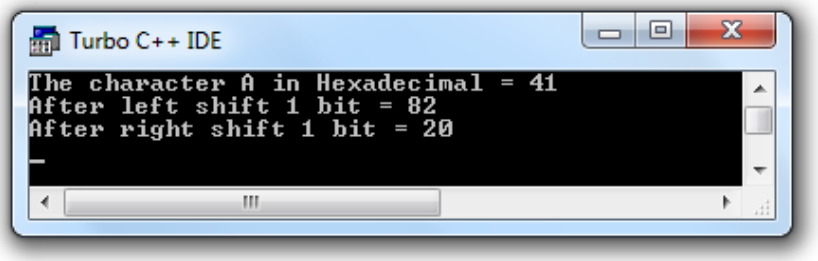
Express_1 แทนตัวเลขที่ต้องการจะเลื่อนที่ละบิต

Express_2 แทนจำนวนบิตที่ต้องการจะเลื่อน

ตัวอย่างโปรแกรมภาษาซี

```
#include<stdio.h>
#include<conio.h>
void main()
{       int c,d;
        clrscr();
        c = 'A';    // 41(Hexadicimal) => 0100 0001(Binary)
        printf("The character A in Hexadecimal = %x\n",c);
        d = c << 1; // 82 (Hexadicimal) => 1000 0010(Binary)
        printf("After left shift 1 bit = %x\n",d);
        d = c >> 1; // 20 (Hexadicimal) => 0010 0000(Binary)
        printf("After right shift 1 bit = %x\n",d);
        getch();
}
```

ผลการทำงานของโปรแกรม



ตัวดำเนินการคอมพลีเมนต์ (~)

เป็นตัวดำเนินการในการกลับบิตทั้งหมดให้เป็นตรงกันข้าม จากค่าบิต 0 เป็น 1 และจาก 1 เป็น 0 เช่น ถ้ากำหนด

x = 11110000 10100101

~x = 00001111 01011010

ตัวดำเนินการยูนารี

ภาษาซีมีกลุ่มของตัวดำเนินการซึ่งใช้กับตัวถูกดำเนินการเพียงตัวเดียว จึงเรียกตัวดำเนินการ ว่าตัวดำเนินการยูนารี เป็นการเพิ่มค่า และลดค่าที่ 1 ค่า เข้าไปในตัวถูกดำเนินการ โดยการเขียน ++i เป็นการเพิ่มค่าที่ละ 1 ให้กับตัวแปร i และถ้าเขียน --j เป็นการลดค่าที่ละ 1 ออกจากตัวแปร j การเขียนสามารถเขียนได้ 2 รูปแบบ คือ

นำเครื่องหมาย ++ หรือ – อยู่หน้าตัวดำเนินการ

นำเครื่องหมาย ++ หรือ – อยู่หลังตัวดำเนินการ

ตัวอย่างเช่น กำหนดให้ x = 1

z = x++

หมายความว่า หลังทำงานคำสั่งนี้ x=1 และ z = 2 (หมายเหตุ x จะเท่ากับ 2 เมื่อมีการประมวลผลอีกรอบ)

สามารถเขียนได้อีกรูปแบบ

z = ++x

หลังการประมวลผลคำสั่งนี้ ค่าของตัวแปร z และ x มีค่าเท่ากันคือ 2 ทั้งสองตัวแปร

สรุป

การคำนวณของโปรแกรมคอมพิวเตอร์ จะทำการประมวลผล จากนิพจน์ ทางคณิตศาสตร์ หรือเรียกว่า สมการ ในการดำเนินการสมการเครื่องหมายเท่ากับ (=) เป็นตัวกำหนดผลการดำเนินการคำนวณ กล่าวคือ ค่าทางซ้ายมือของเครื่องหมายเท่ากับ เป็นส่วนของการเก็บผลการคำนวณจากสมการทางขวามือ ของเครื่องหมายเท่ากับ นอกจากนี้ในภาษาซีมีตัวดำเนินการกับตัวถูกดำเนินการเป็นตัวแปรตัวเดียวกันได้ เรียกว่า ยูนารี เช่น คำสั่ง i++ เป็นต้น

แบบฝึกหัดท้ายบทที่ 3

ตอนที่ 1 จงเติมคำหรือข้อความในช่องว่างต่อไปนี้ให้ถูกต้อง

1. ตัวดำเนินการในภาษาซีมีความแตกต่างกับตัวถูกดำเนินการอย่างไร

.....

2. เครื่องหมาย = กับเครื่องหมาย == มีความแตกต่างกันอย่างไร

.....

3. จงเขียนตัวดำเนินการแบบลัดมาอย่างน้อย 4 ตัวดำเนินการ

.....

4. อยากทราบว่า % เป็นตัวดำเนินการใด

.....

5. อยากทราบผลลัพธ์ของนิพจน์ 9%5 มีค่าเท่าใด

.....

6. อยากทราบผลลัพธ์ของนิพจน์ 9/5 มีค่าเท่าใด

.....

7. ตัวดำเนินการเปลี่ยนชนิดข้อมูลมีรูปแบบอย่างไร

.....

8. อยากทราบว่า a – a ได้หรือไม่

.....

9. อยากทราบว่า ถ้า x = 9.55 เมื่อทำการ ((int x)+10) จะมีค่าเท่าใด

.....

10. อยากทราบว่า ถ้า x = 9.55 และ y = 5.45 เมื่อทำการ ((int x)+y) จะมีค่าเท่าใด

.....

ตอนที่ 2 จงทำเครื่องหมายกากบาท (x) ทับหน้าข้อที่ถูกต้องที่สุด

1. อยากทราบว่าเครื่องหมาย + มีลำดับการประมวลผลที่เท่าใด

ก. 1 ข. 2

ค. 3 ง. 4

2. อยากทราบว่าเครื่องหมาย * มีลำดับการประมวลผลที่เท่าใด

ก. 1 ข. 2

ค. 3 ง. 4

3. อยากทราบว่าเครื่องหมาย / มีลำดับการประมวลผลที่เท่าใด

ก. 1 ข. 2

ค. 3 ง. 4

4. อยากทราบว่าเครื่องหมาย ++ มีลำดับการประมวลผลที่เท่าใด

ก. 1 ข. 2

ค. 3 ง. 4

5. อยากทราบว่าเครื่องหมาย % มีลำดับการประมวลผลที่เท่าใด

ก. 1 ข. 2

ค. 3 ง. 4

6. อยากทราบว่าข้อใดเขียนตัวดำเนินการเชิงสัมพันธ์ผิด

ก. >= ข. =<

ค. > ง. <

7. อยากทราบว่าข้อใดเขียนตัวดำเนินการเปรียบเทียบถูกต้อง

ก. <> ข. !=

ค. >< ง. .>

8. อยากทราบว่าข้อใดให้ค่าเท่ากับ 1 หรือ true

ก. 0 > 10 ข. 0 != 10

ค. 10 > 10 ง. 0 > (0/10)

9. อยากทราบว่าข้อใดให้ค่าเท่ากับ 1 หรือ true

ก. 0 > 10 && 10 > 0 ข. 0 != 10 && 10 != 0

ค. 10 > 10 || 10 < 10 ง. 0 > (0/10) || 0 = (0/10)

10. อยากทราบว่าข้อใดให้ค่าเท่ากับ 0 (ศูนย์)

ก. a | a ข. A ^ a

ค. a ^ a ง. A & a

……..

ตอนที่ 3 จงทำการวิเคราะห์คำถามและทำการเขียนอภิปรายคำตอบตามที่ผู้อ่านเข้าใจโดยยึดความถูกต้องของเนื้อหาประกอบการบรรยาย

1. ให้ผู้อ่านหาค่าของนิพจน์ กำหนดให้ a b และ c เป็นตัวแปรจำนวนเต็มมีค่าดังนี้ 8 3 และ -5 ตามลำดับ

   2 * b +3 * (a – c)

   a /b

   a % b

   (a * c) % b

   a * (c % b)

2. จงคำนวณหาค่าของนิพจน์ด้านล่างโดยกำหนดให้มีจากการประกาศตัวแปร และกำหนดค่าตัวแปรดังนี้

   int i = 8, j =5, k;

   float x = 0.005, y = -0.01, z;

   char a, b, c =’c’, d = ‘d’;

   จงคำนวณหาค่าของผลลัพธ์ดังนี้

   ก. k = (i +j)

   ข. z = (x + y)

   ค. i = j

   ง. k =(x + y)

   จ. k = c

   ฉ. z = i / j

3. จงคำนวณหาค่าของนิพจน์ดังต่อไปนี้

ก. a = b = c

ข. i = j = 1.1

ค. z =k =x

ง. k =z = x

จ. i +=2

ช. y -=x

4. จงคำนวณหาค่าของนิพจน์ดังต่อไปนี้

ก. x *=2

ข. i /=y

ค. i %=y

ง. i += (j -2)

จ. k = (j == 5)? i : j

ช. k = (j > 5)? i : j

5. จงคำนวณหาค่าของนิพจน์ดังต่อไปนี้

ก. z = (x >= 0)? x : 0

ข. z = (y >= 0)? y : 0

ค. a = (c < d)? c : d

ง. i = (j > 0)? j : 0

……

เอกสารอ้างอิง

ศรัณย์ อินทโกสุม (2539). ทฤษฎีและตัวอย่างโจทย์การเขียนโปรแกรมด้วยภาษาซี กรุงเทพฯ : แมคกรอฮิล อินเตอร์เนชั่นแนล เอ็นเตอร์ไพรส์, อิงค์.

ธันวา ศรีประโมง (2539). การเขียนโปรแกรมภาษาซีสำหรับวิศวกรรม. พิมพ์ครั้งที่ 4. กรุงเทพฯ : มหาวิทยาลัยเทคโนโลยีมหานคร.

วิจักษณ์ ศรีสัจจะเลิศวาจา และดุษฎี ประเสริฐธิติพงษ์ ออนไลน์ : www.satit.su.ac.th/soottin.

Alexander, A. **Tutorial** Online : **http://www.cprogramming.com**.

Brian, W. K. Programming in C: A Tutorial Online : http://www.lysator.liu.se/c/bwktutor.html.

Steven, H. & Lutfar, R. (2006). Art of Programming Contest: C Programming | Data Structure | Algorithms (ACM supported), 2nd Edition.