บทที่ 3 โครงสร้างของโปรแกรมภาษาซี

เนื้อหา

3.1 เริ่มต้นกับภาษาซี

3.2 การคอมไพล์และลิงค์โปรแกรมในภาษาซี

3.3 องค์ประกอบพื้นฐานของภาษาซี

3.4 คำสงวน

3.5 ชนิดข้อมูลพื้นฐาน

3.6 ตัวแปร

3.7 ค่าคงที่

3.8 ตัวดำเนินการ

3.9 นิพจน์

ผลการเรียนรู้ที่คาดหวัง

1. บอกความหมายของข้อมูลและชนิดของข้อมูลที่ใช้ในการเขียนโปรแกรมได้
2. อธิบายและเปรียบเทียบตัวแปรและค่าคงที่ที่ใช้ในการเขียนโปรแกรมได้
3. บอกความหมายของนิพจน์และสามารถเขียนนิพจน์เพื่อใช้งานได้
4. อธิบายตัวดำเนินการที่ใช้ในการเขียนโปรแกรมและสามารถใช้ตัวดำเนินการต่างๆ ได้ตามความเหมาะสม

3.1 เริ่มต้นกับภาษาซี

ภาษาซีเป็นภาษาโปรแกรมระดับสูง ที่ใช้สำหรับเขียนโปรแกรมประยุกต์ต่างๆ เช่นเดียวกันกับภาษาปาสคาล ภาษาเบสิก เป็นต้น นอกจากนี้ภาษาซียังใช้สำหรับเขียนโปรแกรมระบบ และโปรแกรมสำหรับควบคุมฮาร์ดแวร์บางส่วนที่ภาษาโปรแกรมระดับสูงหลายภาษาไม่สามารถทำได้ ดังนั้นภาษาซีจึงจัดเป็นภาษาโปรแกรมในระดับกลางด้วย

ก่อนที่โปรแกรมภาษาซีจะถูกรัน (run) จะต้องถูกแปลงให้อยู่ในรูปของอ๊อบเจกต์โค้ด (object code) โดยการคอมไพล์ (compile) โปรแกรมภาษาซีที่เขียนโดยใช้คำสั่งตามมาตรฐานของ ANSI C สามารถนำไปคอมไพล์และรันที่เครื่องคอมพิวเตอร์ต่างระบบกันได้

โปรแกรมที่เขียนขึ้นโดยใช้ภาษาโปรแกรมต่างๆ นั้นเราเรียกว่า รหัสต้นฉบับ(source code) ซึ่งอยู่ในรูปของข้อความตามหลักการเขียนโปรแกรมของภาษาโปรแกรมที่สามารถอ่าน และทำความเข้าใจได้โดยมนุษย์เท่านั้น ดังนั้น

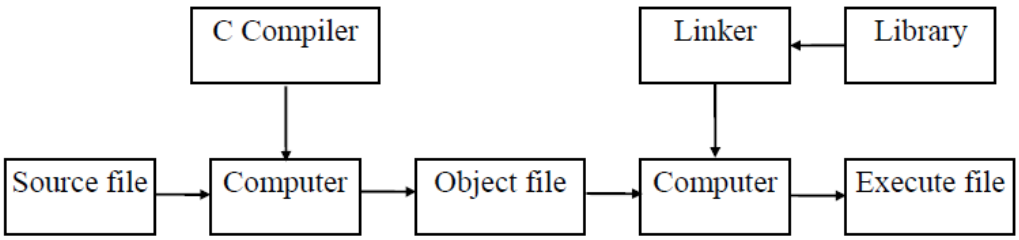
3.2 การคอมไพล์และลิงค์โปรแกรมในภาษาซี

การสร้างโปรแกรมที่สามารถใช้งานได้ขึ้นมาโปรแกรมหนึ่ง ในภาษาซีมีขั้นตอนดังนี้

1) สร้างตัวโปรแกรมที่เป็นตัวอักษรหรือเรียกว่า **ซอร์สไฟล์ (Source file)** โดยมีนามสกุลเป็น .c หรือ .cppขึ้นมาก่อน โดยใช้โปรแกรมที่สามารถเขียนไฟล์ที่เก็บอักขระ (Editor) ใดๆ ก็ได้ อักษรหรืออักขระใดๆ นั้นจะต้องอยู่ในรูปแบบของการโปรแกรมภาษา (ขั้นตอนนี้คือการสร้างโปรแกรมที่เป็นภาษามนุษย์นั่นเอง)

2) คอมไพล์เลอร์ของภาษาซี (C Compiler) จะทำการแปลงซอร์สไฟล์ จากอักขระใดๆ ให้เป็นรหัสที่เครื่องคอมพิวเตอร์สามารถเข้าใจได้เก็บไว้ในอีกไฟล์หนึ่งเรียกว่าไฟล์วัตถุประสงค์ (Object file) ที่มีนามสกุล .obj (ขั้นตอนนี้เรียกว่า **การคอมไพล์** เป็นการแปลงภาษามนุษย์เป็นภาษาเครื่องนั่นเอง)

3) ตัวเชื่อม (Linker) จะทำการตรวจสอบว่าในโปรแกรมที่เขียนขึ้นนั้น มีการเรียกใช้งานฟังก์ชันมาตรฐานใดจากห้องสมุดของภาษาซี (C Library) บ้างหรือไม่ ถ้ามี ตัวเชื่อมจะทำการรวมเอาฟังก์ชันเหล่านั้นเข้ากับไฟล์วัตถุประสงค์ แล้วจะได้ไฟล์ที่สามารถทำงานได้ โดยมีนามสกุลเป็น .exe (ขั้นตอนนี้เรียกว่า **การลิงค์** เป็นการรวมฟังก์ชันสำเร็จรูปเข้าไป แล้วสร้างไฟล์ที่ทำงานได้)



รูป แสดงขั้นตอนการคอมไพล์และลิงค์โปรแกรมภาษาซี

3.3 องค์ประกอบพื้นฐานของภาษาซี

ภาษาซี (C Language) เกิดขึ้นจากการดัดแปลงภาษาบี (B Language) ภาษาซีเป็นภาษาที่คิดค้นโดยเดนนิส รินชี่(Dennis Ritchie) ที่ศูนย์วิจัยเบล(Bell Laboratories) ในสหรัฐอเมริการาวปี ค.ศ. 1972 และเป็นภาษาที่ใช้เขียนระบบปฏิบัติการยูนิกส์(UNIX Operating System) ซึ่งเป็นระบบปฏิบัติการในระบบเครือข่ายคอมพิวเตอร์ที่แพร่หลายในปัจจุบัน องค์ประกอบพื้นฐานของภาษาซีมีดังนี้

```
Comments ..... (1)
Preprocessor directives ..... (2)
Global declaration ..... (3)
main ( ) ..... (4)
{
        declaration statement;
        assignment statement;
        user-defined function call( ); ..... (5)
}
```

รูป องค์ประกอบพื้นฐานของภาษาซี

1) ส่วนอธิบายโปรแกรม (Comments)

ใช้สำหรับเขียนอธิบายการทำงานของโปรแกรม ซึ่งข้อความในส่วนนี้ เมื่อผ่านการแปลคำสั่ง ตัวแปรคำสั่งจะไม่สนใจว่าเป็นส่วนหนึ่งของโปรแกรม เพื่อให้เข้าใจและอ่านโปรแกรมง่ายขึ้น การเขียนส่วนอธิบายโปรแกรมจะใช้เครื่องหมาย // หรือ /* และ */ คร่อมข้อความที่ต้องการอธิบาย ดังนี้

```
// ..... ข้อความที่ต้องการอธิบาย
```

แต่ถ้าต้องการเขียนอธิบายหลายๆบรรทัดจะเขียนได้ดังนี้

```
/* ..................................................
   ..... ข้อความที่ต้องการอธิบาย.....
   ..................................................*/
```

การเขียนส่วนคำอธิบายโปรแกรมจะเขียนไว้ที่ไหนในโปรแกรมก็ได้ โดยมากจะนิยมเขียนอธิบายขั้นตอนการทำงานก่อนเขียนคำสั่ง หรือ อธิบายความหมายของคำสั่งไว้ข้างคำสั่งนั้นก็ได้

2) พรีโปรเซสเซอร์ไดเรกทีฟ (Preprocessor directives)

เป็นส่วนที่ทุกโปรแกรมต้องมี ใช้สำหรับเรียกไฟล์ที่โปรแกรมต้องการ และกำหนดค่าต่างๆ ซึ่งจะต้องเริ่มต้นด้วยเครื่องหมายไดเรกทีฟ (#) และตามด้วยชื่อที่ต้องการกำหนดค่า

ตัวอย่างไดเรกทีฟที่ใช้บ่อย

#include เป็นการบอกให้ตัวแปลคำสั่ง (compiler) อ่านไฟล์อื่นเข้ามาร่วมในการคอมไพล์มีรูปแบบดังนี้

```
#include<filename> หรือ  #include"filename"
```

เช่น #include<dos.h>

อ่านไฟล์ dos.h จากไดเร็คทรอรีที่กำหนด

#include"sample.h"

อ่านไฟล์ sample.h จากไดเร็คทรอรีปัจจุบันหรือที่กำหนด

- การใช้เครื่องหมาย <...> คร่อมชื่อ เพื่อบอกให้คอมไพเลอร์ค้นหาไฟล์จากไดเร็คทรอรีที่กำหนดใน Option directory แต่ถ้าใช้เครื่องหมาย "..." จะเป็นการกำหนดให้ค้นหาจากไดเร็คทรอรีปัจจุบัน และไดเร็คทรอรีที่กำหนดเส้นทางไว้

- ชื่อฟังก์ชันการทำงานพื้นฐานของภาษาซี ตัวแปร ค่าคงที่ และมาโครพื้นฐาน โดยปกติจะรวมกันเป็นกลุ่มในไฟล์ที่เรียกว่าไฟล์หัว (Header Files) คือไฟล์ ที่มีนามสกุล .h ไฟล์หัวแต่ละไฟล์จะเก็บโมดูลแยกกันเป็นส่วนการทำงานแต่ละส่วนเพื่อให้ผู้เขียนโปรแกรมสามารถเรียกใช้งานได้ เช่น ในตัวอย่างที่ 1 จะเห็นว่ามีการเรียกใช้ไฟล์ stdio.h ซึ่งในไฟล์ดังกล่าวมีการกำหนดคำสั่ง printf ซึ่งถูกเรียกใช้ในฟังก์ชัน main() และ ฟังก์ชัน

process แต่ถ้าเราต้องการเรียกใช้ฟังก์ชัน clrscr() ซึ่งไม่มีในไฟล์นี้ แต่มีในไฟล์ conio.h เราก็จะต้อง include ไฟล์ conio.h เข้ามาด้วย

ตัวอย่างไดเรกทีฟการใช้งาน

#include	Include text from a file
#define	Define a macro
#undef	Undefined a macro
#if	Test if a compile-time condition holed.
#ifdef	Test if a symbol is defined
#ifndef	Test if a symbol is not defined
#else	Indicate alternative if a test file fail.
#line	Give a line number for compiler message.

#define ทำหน้าที่ใช้กำหนดค่าคงที่ ที่เป็นชื่อแทน คำ นิพจน์ คำสั่ง หรือคำสั่งหลายคำสั่ง มีรูปแบบการใช้งานดังนี้

```
#define  ชื่อตัวแปร (ชื่อที่ใช้แทน)  ค่าที่ต้องการกำหนด
```

เช่น #define TEN 10 *กำหนดตัวแปร TEN แทนค่า 10*

#define PI 3.141592654 *กำหนดตัวแปร PI แทนค่า 3.141592654*

3) ส่วนประกาศ (Global Declaration)

ใช้ในการประกาศชื่อตัวแปร หรือฟังก์ชันที่ต้องใช้ในโปรแกรม โดยทุกๆ ส่วนของโปรแกรมสามารถจะเรียกใช้ข้อมูลส่วนนี้ได้

```
#include<stdio.h>
int x;                          //global declaration
void main(){
        x = 7;
        printf("%d",x);
}
```

4) ส่วนฟังก์ชันหลัก (main program)

เป็นส่วนที่โปรแกรมภาษาซีทุกโปรแกรมต้องมี ในฟังก์ชัน main จะประกอบด้วยคำสั่งต่างๆ ที่ใช้ในการสั่งงานโปรแกรมเรียงต่อๆ กัน โดยแต่ละประโยคคำสั่งจะจบลงด้วยเครื่องหมาย semi colon (;)

```
#include<stdio.h>
void main(){
        int a,b;
        scanf("%d",&a);
        scanf("%d",&b);          // main program
        printf("%d",a+b);
}
```

5) ส่วนของการเรียกใช้ฟังก์ชันที่ผู้ใช้สร้างขึ้นเอง (User defined function call)

เป็นส่วนของการเรียกใช้ฟังก์ชันที่ผู้ใช้สร้างขึ้นเอง ซึ่งนอกเหนือจากฟังก์ชันโปรแกรมภาษาซีมีไว้ให้

```
#include<stdio.h>
int sum(int a,int b){
        return(a+b);
}
void main(){
        printf("%d",sum(5,9));        // user defined function call
}
```

3.4 คำสงวน

คำสงวน หมายถึง คำที่สงวนไว้สำหรับเรียกใช้ตามวัตถุประสงค์ที่กำหนดไว้เฉพาะ เช่น คำที่ใช้ในคำสั่งควบคุม และชนิดของข้อมูล เป็นต้น คำสงวนของภาษาซีเช่น

auto	default	float	register	struct	volatile	break
do	far	return	switch	while	case	double
goto	short	typedef	char	else	if	signed
union	const	enum	int	sizeof	unsigned	continue
extern	long	static	void			

3.5 ชนิดข้อมูลพื้นฐาน

ชนิดข้อมูลพื้นฐานมีด้วยกัน 5 ประเภทคือ

1) char (1 ไบต์)　　2) int (2 ไบต์)　　3) float (4 ไบต์)

4) double (8 ไบต์)　　5) void (0 ไบต์)

ชนิดข้อมูล	ความหมาย	ไบต์ (bytes)	พิสัย (range)
char	อักษรหรืออักขระ	1	-128 ถึง 127
Int	จำนวนเต็ม	2	-32,768 ถึง 32,767
float	จำนวนจริง (เลขทศนิยม)	4	3.4E + 38 (7 ตำแหน่ง)
double	จำนวนจริงละเอียด 2 เท่า	8	1.7E + 308 (15 ตำแหน่ง)
void	ไม่ให้ค่าใด ๆ	0	ไม่ให้ค่า

ตารางชนิดข้อมูลพื้นฐานของภาษา C

ชนิดข้อมูลในตารางที่ 1 ที่ได้กล่าวมาแล้วนั้น ตัวแปรจะมี 4 ประเภทเท่านั้นที่สามารถเก็บข้อมูลเหล่านี้ เพื่อความคล่องตัว และสามารถเก็บข้อมูลนอกเหนือจากนี้ได้ ในภาษาซีจึงได้สร้างตัวแปรชนิดข้อมูลแบบ คิดเครื่องหมาย(signed) ไม่คิดเครื่องหมาย (unsigned), ยาว (long) และสั้น (short) เพิ่มเติมขึ้นมา ดังตารางที่ 2

ชนิดข้อมูล	ความหมาย	ไบต์ (bytes)	พิสัย (range)
char	คิดเครื่องหมาย	1	-128 ถึง 127
int	คิดเครื่องหมาย	2	-32,768 ถึง 32,767
short	คิดเครื่องหมาย	2	-32,768 ถึง 32,767
short int	คิดเครื่องหมาย	2	-32,768 ถึง 32,767
long	คิดเครื่องหมาย	4	-2,147,483,648 ถึง 2,147,483,647
long int	คิดเครื่องหมาย	4	-2,147,483,648 ถึง 2,147,483,647
Unsigned Char	ไม่คิดเครื่องหมาย	1	0 ถึง 255
unsigned	ไม่คิดเครื่องหมาย	2	0 ถึง 65,535
unsigned int	ไม่คิดเครื่องหมาย	2	0 ถึง 65,535
Unsigned short	ไม่คิดเครื่องหมาย	2	0 ถึง 65,535
Unsigned Long	ไม่คิดเครื่องหมาย	4	0 ถึง 4,294,967,295
signed Char	คิดเครื่องหมาย	1	-128 ถึง 127
signed	คิดเครื่องหมาย	2	-32,768 ถึง 32,767
signed int	คิดเครื่องหมาย	2	-32,768 ถึง 32,767
signed short	คิดเครื่องหมาย	2	-32,768 ถึง 32,767
signed Long	คิดเครื่องหมาย	4	-2,147,483,648 ถึง 2,147,483,647

ตารางชนิดของตัวแปรในภาษา C

3.6 ตัวแปร

ตัวแปร (Variables) คือชื่อเรียกแทนพื้นที่เก็บข้อมูลในหน่วยความจำ การกำหนดตัวแปรทำได้ 2 แบบ

1) กำหนดไว้นอกกลุ่มคำสั่ง หรือฟังก์ชัน เรียกตัวแปรนี้ว่า Global Variable กำหนดไว้นอกฟังก์ชันใช้งานได้ทั้งโปรแกรม มีค่าเริ่มต้นเป็น 0 (กรณีไม่ได้กำหนดค่าเริ่มต้น)

2) กำหนดไว้ในกลุ่มคำสั่ง หรือฟังก์ชัน เรียกตัวแปรนี้ว่า Local Variable กำหนดไว้ภายในฟังก์ชันใช้งานได้ภายในฟังก์ชันนั้น และไม่ถูกกำหนดค่าเริ่มต้นโดยอัตโนมัติ

การประกาศตัวแปร มีลักษณะดังนี้

```
ชนิดข้อมูล ชื่อตัวแปร[,ชื่อตัวแปร];
```

กฎในการตั้งชื่อ มีดังนี้

1. ประกอบด้วยตัวอักษร ตัวอักษรปนตัวเลข หรือปนเครื่องหมาย underscore (_) ได้
2. ตัวแรกจะต้องเป็นตัวอักษรหรือเครื่องหมาย underscore (_) เท่านั้น
3. ถ้าใช้ underscore เป็นส่วนของชื่อจะต้องอยู่ระหว่างตัวอักษรหรือตัวเลขเสมอ
4. มีความยาวได้ตั้งแต่ 1 ตัวอักขระ ไปจนถึง 32 ตัวอักขระ (เฉพาะเทอร์โบซีแต่บางเครื่องอาจได้น้อยหรือมากกกว่านี้)
5. ห้ามตั้งชื่อซ้ำกับคำสงวน (Reserved word)
6. ชื่อที่ตั้งขึ้นแล้วเขียนเป็นตัวเล็ก ตัวใหญ่หรือตัวใหญ่ปนตัวเล็กจะถือว่าเป็นคนละชื่อกันหมด เช่น count, COUNT, Count จะถือเป็น 3 ชื่อตัวแปรที่แตกต่างกัน

ชื่อตัวแปรที่ถูก	*ชื่อตัวแปรที่ผิด*
Count	1count
Num12	num !
m_mum	m...mun

ข้อสังเกต การกำหนดชนิดของตัวแปร มีสิ่งที่ควรพิจารณาอยู่ 2 ประการคือ ตัวแปรนั้น จะต้องสามารถรับค่าได้ทุกค่าโดยไม่เกินขอบเขตของข้อมูลชนิดนั้น และตัวแปรจะต้องไม่ใช้หน่วยความจำมากเกินความจำเป็น เช่น ถ้าข้อมูลไม่เกินขอบเขตของ int ก็ไม่ควรกำหนดตัวแปรให้เป็น float

3.7 ค่าคงที่

ค่าคงที่ (Constant) คือค่าของข้อมูลที่มีจำนวนแน่นอน เป็นค่าที่ไม่สามารถเปลี่ยนแปลงได้ขณะรันโปรแกรม

การประกาศค่าคงที่ มีลักษณะดังนี้

```
const ชนิดของข้อมูล ชื่อค่าคงที่ = ค่า;
```

ตัวอย่าง const float Pi = 3.1415; หมายความว่า Pi เป็นค่าคงที่ชนิด float ซึ่งมีค่าเท่ากับ 3.1415

การประกาศค่าคงที่แบ่งตามชนิดต่างๆ ได้ดังนี้

1) ค่าคงที่ชนิดจำนวนเต็ม

ค่าคงที่ชนิดจำนวนเต็ม (integer constant) เป็นค่าคงที่ชนิดตัวเลขซึ่งอาจจะเป็นค่าบวก 0 หรือ ค่าลบ ค่าคงที่ชนิดนี้มี 3 ประเภท คือ ค่าคงที่อินทีเจอร์ฐานสิบ ฐานแปดและฐานสิบหก ซึ่งมีรายละเอียดดังนี้

- ค่าคงที่ชนิดจำนวนเต็มฐานสิบ (decimal integer constant) หมายถึงชุดของเลขฐานสิบ (0 ถึง 9) ซึ่งเลขตัวแรกไม่เป็น 0 ดังตัวอย่างต่อไปนี้

```
const int price = 1000;
const int WeekDay = 7;
```

- ค่าคงที่ชนิดจำนวนเต็มฐานแปด (octal integer constant) หมายถึงชุดของเลขฐานแปด (0 ถึง 7) ซึ่งเลขตัวแรกเป็น 0 ดังตัวอย่างต่อไปนี้

```
const int Oct1 = 01000;
const int Oct2 =03 7;
```

- ค่าคงที่ชนิดจำนวนเต็มสิบหก (hexadecimal integer constant) หมายถึงชุดของเลขฐานสิบหก (0 ถึง 9 และ A ถึง F) ซึ่งนำหน้าด้วย 0X หรือ 0x ดังตัวอย่างต่อไปนี้

```
const int Hex1 = 0xF78;
const int Hex2 =0xFF;
```

2) ค่าคงที่ชนิดจำนวนจริง

ค่าคงที่ชนิดจำนวนจริง (floating point constant) เป็นค่าคงที่ชนิดตัวเลขที่มีทศนิยม ค่าคงที่ชนิดนี้มี 2 รูปแบบดังนี้

- เป็นชุดของตัวเลขที่ประกอบด้วยเลข 0 ถึง 9 และจุดทศนิยม ดังตัวอย่างต่อไปนี้

```
const float Pi = 3.1415;
const float InterestRate =0.65;
```

- เป็นเลขยกกำลังซึ่งมีส่วนประกอบ 2 ส่วนคือ ส่วนหนึ่งเป็นอินทีเจอร์หรือ float อีกส่วนหนึ่งเป็นตัวคูณสิบยกกำลังซึ่งแสดงด้วยอักษร E (หรือ e) ตามด้วยเลขยกกำลังซึ่งเป็นอินทีเจอร์บวกหรือลบดังตัวอย่างต่อไปนี้

```
const float A = 1.0E2; // หมายถึง 1.0 * 102 ซึ่ง = 100.0
```

const float A = 1E+3; // หมายถึง 1 * 103 ซึ่ง = 1000.0

const float A = 1E-2; // หมายถึง 1 * 10 - 2 ซึ่ง = 0.01

3) ค่าคงที่ชนิดอักษร

ค่าคงที่ชนิดอักษร(character constant) เป็นค่าคงที่ที่ประกอบด้วยอักษรหนึ่งตัวอยู่ภายในเครื่องหมาย (‘ ‘) ดังตัวอย่างต่อไปนี้

const char A = ‘a’;

const char B =’b’;

ค่าคงที่ชนิดอักษรมีชนิดเป็น char ค่าของค่าคงที่จะเป็นตัวเลขซึ่งมีค่าเท่ากับรหัส ASCII ของอักษรนั้น เช่น ‘A’ = 65, ‘B’ = 66

ค่าคงที่ชนิดอักษรอีกประเภทหนึ่งมีชื่อว่า เอสเคปซีเควนซ์ (escape sequence) ซึ่งเป็นค่าคงที่ที่ประกอบด้วยเครื่องหมาย \ และอักษรอยู่ภายในเครื่องหมาย (‘ ‘) เช่น ‘ \n ’ , ‘ \t ’ ค่าคงที่ประเภทนี้มีจำนวนจำกัดและมีความหมายเฉพาะดังแสดงในตารางที่ 3

เอสเคปซีเควนซ์	ชื่อ	ความหมาย
\a	Alarm bell	เตรียมพร้อม (เสียงระฆัง)
\b	Backspace	เลื่อนกลับ
\f	Formfeed	ขึ้นหน้าใหม่
\n	Newline	ขึ้นบรรทัดใหม่
\r	Return	เคอเซอร์กลับไปอยู่ที่ต้นบรรทัด
\t	Tab	เว้นระยะในแนวระดับ
****	Backslash	การกดเครื่องหมาย \\
\’	Single quote	การกดเครื่องหมาย \
\”	Double quote	การกดเครื่องหมาย “
\?	Question mark	การกดเครื่องหมาย ?

ตาราง แสดงค่าคงที่แบบต่าง ๆ

3.8 ตัวดำเนินการ

ตัวดำเนินการ (Operator) คือการใช้ในการกำหนดค่าให้ตัวแปร มีตัวดำเนินการ 3 ประเภทใหญ่ ๆ ดังนี้

1) ตัวดำเนินการเลขคณิต (arithmetic operators)

2) ตัวดำเนินการเปรียบเทียบและ (relational operators)

3) ตัวดำเนินการตรรกะ (logical operators)

1) **ตัวดำเนินการเลขคณิต (arithmetic operators)** เป็นตัวดำเนินการทางด้านคณิตศาสตร์ ได้แก่ เครื่องหมายที่ใช้ในการบวก ลบ คูณ หาร ตัวเลข และอื่น ๆ ดังในตารางที่ 4

ตัวดำเนินการ	ความหมาย	ตัวอย่าง
-	การลบ	X – Y
+	การบวก	X + Y
*	การคูณ	X * Y
/	การหาร	X / Y
%	การหารจะเอาเฉพาะเศษไว้	11%3 = 3 เศษ 2 ดังนั้น 2 เป็นผลลัพธ์
- -	การลดค่าลงครั้งละ 1	X- -หรือ - -X เหมือนกับ X=X-1
+ +	การเพิ่มค่าครั้งละ1	X+ +หรือ + +X เหมือนกับ X=X+1

ตาราง แสดงตัวดำเนินการของ C

นอกจากตัวดำเนินการบวก ลบ คูณ หาร ธรรมดาแล้ว ยังมีโมดูลัส (modulas) คือการหารเอาเฉพาะเศษดังในตารางที่ 4 และใน C มีวิธีการรวบรัดการใช้ตัวดำเนินการเลขคณิตดังนี้ เช่น รวมสัญลักษณ์ (+) และ (=) เข้าด้วยกันกลายเป็น (+=) หรือรวมสัญลักษณ์ (-) และ(=) เข้าด้วยกันกลายเป็น (-=) ดังตารางที่ 5

นิพจน์	ขั้นตอนการประมวลผล
y = x + a ++;	1. y = x + a 2. a = a + 1
y = x + ++ a;	1. a = a + 1 2. y = x + a
y += x ;	1. y = y + x
y += x ++ ;	1. y = y + x 2. x = x + 1
y -= 9 ;	1. y = y - 9
y *= 7 * x++ ;	1. y = y * 7 * x 2. x = x + 1
y /= x ;	1. y = y / x
y %= x ;	1. y = y % x

ตาราง แสดงตัวอย่างการดำเนินการของภาษา C

ขั้นตอนการทำงานของตัวดำเนินการใน C

ในบางครั้งนิพจน์ประกอบด้วยตัวดำเนินการมากมาย ทำให้เกิดความยุ่งยากในการพิจารณาขั้นตอนการทำงานของตัวดำเนินการ จึงได้ตั้งกฎเกี่ยวกับลำดับการทำงานก่อนหลังของตัวดำเนินการ (Operator) ดังตาราง 6

ตัวดำเนินการ	ลำดับที่
( )	1 (สูงสุด)
++ -- unary	2
* / %	3
+ -	4
+= -= *= = /= %=	5 (ต่ำสุด)

ตาราง แสดงลำดับการทำงานของตัวดำเนินการ

ตัวดำเนินการที่มีความสำคัญอยู่ในระดับเดียวกัน ให้ทำงานตามขั้นตอนจากซ้ายไปขวาเป็นหลักต่อไปนี้ เป็นตัวอย่างพอสังเขป ดังตารางที่ 7

ตัวอย่าง	
5 + 6 * 2	♣ ตัวดำเนินการ * อยู่ลำดับสูงกว่า + จึงต้องคูณเลขก่อนแล้วบวกเลข 5 ภายหลัง
2 * 3 – 14 / 7 + 5	♣ ตัวดำเนินการ * และ / อยู่ลำดับเดียวกันให้ทำจากซ้ายไปขวา คือ คูณเลข แล้วหารเลข ♣ ตัวดำเนินการ – และ + อยู่ในลำดับเดียวกันทำจากซ้ายมือก่อนคือ ลบเลข แล้วจึงบวกเลขในภายหลัง

ตาราง แสดงลำดับการทำงานของตัวดำเนินการ

2) **ตัวดำเนินการเปรียบเทียบและ (relational operators)** หมายถึง เครื่องหมายที่ใช้ในการเปรียบเทียบและตัดสินใจผลการเปรียบเทียบจะให้ค่าเป็น 1 และ เท็จจะให้ค่าเป็น 0 เครื่องหมายที่ใช้มีดังตารางที่ 8

เครื่องหมาย	ความหมาย	ตัวอย่าง
>	มากกว่า	A > B (A มากกว่า B)
>=	มากกว่าหรือเท่ากับ	A >= B (A มากกว่าหรือเท่ากับ B)
<	น้อยกว่า	A < B (A กว่า B)
<=	น้อยกว่าหรือเท่ากับ	A <= B (A กว่าหรือเท่ากับ B)
= =	เท่ากับ	A = = B (A กับ B)
! =	ไม่เท่ากับ	A ! = B (A ไม่กับ B)

ตาราง แสดงตัวดำเนินการเปรียบเทียบห

3) **ตัวดำเนินการตรรกะ (logical operators)** เครื่องหมายตรรกะมีจุดประสงค์ใช้ในการเปรียบเทียบเพื่อตัดสินใจซึ่งผลที่ได้จากการเปรียบเทียบจะได้ผลลัพธ์ 2 อย่าง คือ ถ้าได้ผลลัพธ์เป็นจริงจะมีค่าเป็น 1 และถ้าได้ผลลัพธ์เป็นเท็จจะมีค่า 0 เป็นเครื่องหมายตรรกะที่ใช้ในภาษา C มีดังนี้

คำสั่ง && (AND) เป็นการนำเงื่อนไข 2 เงื่อนไขมาเปรียบเทียบกันซึ่งผลลัพธ์ที่ได้จะเป็นดังนี้

X	Y	X && Y
0	0	0
0	1	0
1	0	0
1	1	1

คำสั่ง | | (OR) เป็นการนำเงื่อนไข 2 เงื่อนไขมาเปรียบเทียบกันซึ่งผลลัพธ์ที่ได้จะเป็นดังนี้

X	Y	X | | Y
0	0	0
0	1	1
1	0	1
1	1	1

คำสั่ง ! (NOT) เป็นการทำนิเสธซึ่งผลลัพธ์ที่ได้จะเป็นดังนี้

X	! X
0	1
1	0

3.9 นิพจน์

นิพจน์ (Expression) คือการนำตัวแปร ค่าคงที่ มาสัมพันธ์กันโดยใช้เครื่องหมายอย่างใดอย่างหนึ่งเป็นตัวเชื่อมโดยมีกฎเกณฑ์ทั่ว ๆ ไปในการเขียนนิพจน์ของภาษา C มีดังนี้

1) เขียนตัวอักษรหลายตัวติดกันได้โดยไม่มีเครื่องหมายคั่น เช่น XY ถือเป็น 1 ตัวแปร

2) กรณีนิพจน์มีค่าของตัวแปรหรือค่าคงที่ต่างชนิดกันในนิพจน์เดียวกัน กลไกของภาษา C จะเปลี่ยนชนิดของข้อมูลที่มีขนาดเล็กให้เป็นชนิดของข้อมูลที่ใหญ่ขึ้น ดังนั้นจึงควรระวังในการตั้งตัวแปรเพื่อเก็บผลลัพธ์ที่ได้จากการดำเนินการของนิพจน์มีค่าของตัวแปรหรือค่าคงที่ต่างชนิดกัน ซึ่งตัวแปรที่ตั้งขึ้นควรเป็นชนิดของข้อมูลที่ใหญ่ที่สุดในนิพจน์นั้น ดังตัวอย่าง

- หากนิพจน์มี int กับ long กลไกของภาษา C จะเปลี่ยนชนิดของข้อมูลเป็น long
- หากนิพจน์มี int กับ double กลไกของภาษา C จะเปลี่ยนชนิดของข้อมูลเป็น double

นิพจน์คณิตศาสตร์	นิพจน์ภาษา C
$2X^2$	2*(X*X)
$10X + 3XY + 10Y^2$	10*X-3*X*Y+10*Y*Y

ตาราง แสดงตัวอย่างของนิพจน์ในภาษาซี

นิพจน์แบ่งออกเป็นนิพจน์ต่างๆ ดังนี้

1) นิพจน์คณิตศาสตร์ (Arithmetic Expression)

หมายถึงการนำตัวแปร ค่าคงที่ มาสัมพันธ์กันโดยใช้เครื่องหมายคณิตศาตร์เป็นตัวเชื่อมผลที่ได้จากนิพจน์แบบนี้จะเป็นตัวเลข ดังตารางที่ 10

	นิพจน์คณิตศาสตร์ในภาษา C
$3X + 5Y$ →	3*X + 5*Y
$X^2 - Y^2$ →	X*X - Y*Y

ตาราง แสดงนิพจน์ทางคณิตศาสตร์

2) นิพจน์ตรรกะ (Logical Expression)

หมายถึง การนำตัวแปร ค่าคงที่ หรือนิพจน์ มาสัมพันธ์กันโดยใช้เครื่องหมายเปรียบเทียบและเครื่องหมายตรรกเป็นตัวเชื่อมผลที่ได้จะเป็นจริง หรือเท็จ คือจะให้ค่าเป็น 1 หรือ 0 ออกมาเป็นผลลัพธ์ซึ่งสามารถนำไปคำนวณต่อได้ดังตารางที่ 11

โจทย์→ ถ้า I มีค่าเป็น 3 J มีค่าเป็น 5 A มีค่าเป็น 3 ถ้าเขียนนิพจน์ดังนี้	
I = = J	ผลลัพธ์ไม่จริง (0)
I = = A	ผลลัพธ์จริง (1)
I > J*5	ผลลัพธ์ไม่จริง (0)
I+3 > J –2 && a*2 > 10	ผลลัพธ์ไม่จริง (0)

ตาราง แสดงนิพจน์ตรรกะ