

259201 Computer Programming for Engineers

การรับและแสดงผลในภาษาซี Input/Output Processing in C

Outline

- คำสั่งในภาษา C (Commands in C)
- คำสั่งกำหนดค่า (Assignment Statement)
- ฟังก์ชันการรับและแสดงผลข้อมูล (Input/Output Functions)

คำสั่งในภาษา C

- คำสั่งในภาษา C สามารถแบ่งกว้างๆออกเป็น 2 ลักษณะ
 - คำสั่งที่เป็นคำสั่งจริงๆ เช่น คำสั่งกำหนดค่าในตัวแปร, คำสั่งควบคุม โปรแกรมแบบมีเงื่อนไขหรือการทำงานซ้ำ เป็นต้น
 - คำสั่งที่ติดต่อกับผู้ใช้งาน เช่น คำสั่งในการรับข้อมูล, คำสั่งในการแสดงผลข้อมูล เป็นต้น

คำสั่งกำหนดค่า (Assignment Statement)

- เป็นคำสั่งของภาษา C ไว้สำหรับสั่งให้คอมพิวเตอร์ทำงานต่างๆ ได้แก่
 - การคำนวณ
 - การกำหนดค่าเริ่มต้นให้กับตัวแปร
 - การย้ายค่าตัวแปรจากที่หนึ่งไปยังอีกที่หนึ่ง
- รูปแบบคำสั่ง:

<Variable> = <expression>;

คำสั่งกำหนดค่า (Assignment Statement) (ต่อ)

- ตัวอย่าง

<pre>main() { int x; x = 5; }</pre>	<pre>main() { int a, b, c, d; a = b = c = d = 485; }</pre>
<pre>main() { int a = 5, b = 3, v; v = a*b / 2 - 10; }</pre>	<pre>main() { int a, b; a = 5; b = a; }</pre>

คำสั่งในการรับและแสดงผลข้อมูล

- ในภาษา C การรับและการแสดงผลข้อมูลจะอาศัยการเรียกใช้ฟังก์ชันมาทำงาน
 - ฟังก์ชันมาตรฐาน ได้แก่ ฟังก์ชัน printf(), scanf() etc.
 - ฟังก์ชันที่เก็บไว้ใน I/O library ได้แก่ putchar(), getchar(), getch() etc. โดยเวลาเรียกใช้ฟังก์ชันเหล่านี้จะต้องบอกให้ C compiler รับรู้ โดยใช้คำสั่ง #include เซคเตอร์ไฟล์ชื่อ stdio.h เข้ามาก่อน

การแสดงผลออกทางหน้าจอด้วย printf ()

- รูปแบบคำสั่ง: **printf("control", value);**
- **control** คือส่วนที่ใช้ควบคุมการแสดงผลใน 3 รูปแบบได้แก่
 - 1) ข้อความธรรมดา
 - 2) รหัสควบคุมรูปแบบ(Place Holder or Format code) เช่น %d, %f
 - 3) อักขระควบคุมการแสดงผล (\n, \t)
- โดยส่วนเหล่านี้จะต้องเขียนไว้ภายในเครื่องหมาย " ... " (Double quote)

การแสดงผลออกทางหน้าจอด้วย printf () (ต่อ)

- **value** คือ ค่าของตัวแปร นิพจน์ หรือสิ่งที่ต้องการจะแสดงผล
- ตัวอย่างการใช้เช่น


```
printf("Hello World.\n");
printf("Sum = %d", a + b);
printf("Sum is %d = %d + %d", a + b,a,b);
```

รหัสรูปแบบในภาษา C (Place Holder & Format Code)

- ใช้เพื่อควบคุมการแสดงผลตัวแปร นิพจน์ออกทางหน้าจอ โดยพิจารณาให้สอดคล้องกับทั้งชนิดและค่าของข้อมูลที่ต้องการจะแสดงผลออก

รหัสรูปแบบ	การนำไปใช้
%d	สำหรับแสดงผลเลขจำนวนเต็ม (int, short, unsigned short, long, unsigned long)
%u	สำหรับแสดงผลเลขจำนวนเต็มบวก (unsigned short, unsigned long)
%o	สำหรับแสดงผลออกมาในแบบเลขฐานแปด
%x	สำหรับแสดงผลออกมาในแบบเลขฐานสิบหก
%f	สำหรับแสดงผลทศนิยม (float, double, long double)
%e	สำหรับแสดงผลตัวเลขทศนิยมออกมาในแบบของ e หรือ ยกกำลัง (float, double, long double)
%c	สำหรับแสดงผลอักขระ 1 ตัว (char)
%s	สำหรับแสดงผลข้อความ หรือ อักขระมากกว่า 1 ตัว (string)
%p	สำหรับแสดงผลตัวชี้ตำแหน่ง(pointer)

รหัสรูปแบบในภาษา C (ต่อ)

- การแสดงผลในรูปแบบตัวเลขทศนิยม เราสามารถกำหนดจำนวนหลักหลังจุดทศนิยมได้ โดยใช้รูปแบบ เช่น %nf คือการแสดงผลออกมาเป็นทศนิยมที่มีตัวเลขหลังจุด n ตัว เช่น %.3f จะแสดงตัวเลขหลังจุดทศนิยม 3 หลัก อย่างเช่น 12.345
- %me คือการแสดงผลออกมาในรูปแบบทศนิยมเลขยกกำลังในรูป e และมีตัวเลขหลังจุดทศนิยม n ตัว เช่น %.3e จะแสดงตัวเลขหลังจุดทศนิยม 3 หลัก อย่างเช่น 1.256e + 02

อักขระควบคุมการแสดงผล (Carriage Control)

- ช่วยในการจัดข้อความที่จะแสดงให้เป็นระเบียบ โดยต้องใส่ไว้ภายในเครื่องหมาย " ... " มีดังต่อไปนี้

อักขระควบคุมการแสดงผล	ความหมาย
\n	ขึ้นบรรทัดใหม่
\t	เว้นช่องว่างเป็นระยะ 1 tab (6 ตัวอักษร)
\r	กำหนดให้ cursor ไปอยู่ที่ต้นบรรทัด
\f	เว้นช่องว่างเป็นระยะ 1 หน้าจอ
\b	ลบอักขระตัวสุดท้ายออก

การแสดงผลออกทางหน้าจอด้วย putchar ()

- ในภาษา C นอกเหนือจากการเรียกฟังก์ชัน printf () พร้อมกับกำหนดรหัสรูปแบบ %c เพื่อใช้สำหรับการแสดงผลอักขระออกทางหน้าจอแล้ว เรายังมีฟังก์ชันที่ใช้สำหรับแสดงผลอักขระโดยเฉพาะ ได้แก่ ฟังก์ชัน putchar ()
- รูปแบบคำสั่ง: **putchar(char);**
- **char** เป็นตัวแปรชนิด char หรืออักขระที่เขียนภายในเครื่องหมาย ' ... '

การแสดงผลออกทางหน้าจอด้วย putchar() (ต่อ)

- ตัวอย่างการใช้เช่น

```
#include<stdio.h>
int main( )
{
    char first = 'O';
    putchar(first);
    putchar('k');
    return 0;
}
```

การแสดงผลออกเป็นข้อความด้วย puts()

- เราสามารถใช้งานฟังก์ชัน puts() ในการแสดงข้อความออกทางหน้าจอ
- รูปแบบคำสั่ง: **puts(str);**
 - str** คือ ตัวแปรที่เก็บข้อมูลชนิดข้อความ หรือข้อความที่เขียนอยู่ในเครื่องหมาย “...” เช่น puts(“easy & fun”);

การรับข้อมูลจากคีย์บอร์ดด้วย scanf()

- ฟังก์ชัน scanf() ใช้สำหรับการรับข้อมูลจากผู้ใช้งาน เพื่อนำข้อมูลมาประมวลผลเข้ากับโปรแกรม โดยผู้เขียนโปรแกรมจะกำหนดให้ผู้ใช้อกรอกข้อมูลผ่านทางคีย์บอร์ด
- รูปแบบคำสั่ง: **scanf(“format”, &variable);**
 - format** เป็นการใช้ format code เพื่อกำหนดชนิดข้อมูลที่จะรับเข้ามา โดยชนิดของรหัสรูปแบบตรงกับชนิดของข้อมูลที่จะรับเข้ามา (&variable)
 - variable** คือ ตัวแปรที่ใช้เก็บค่าข้อมูลที่จะรับเข้ามา โดยต้องมีเครื่องหมาย & หน้าชื่อตัวแปร ยกเว้นตัวแปรที่เก็บข้อมูลชนิดข้อความ ที่สามารถใช้ชื่อตัวแปรได้โดยไม่ต้องมีเครื่องหมาย & นำหน้า

การรับข้อมูลจากคีย์บอร์ดด้วย scanf() (ต่อ)

- ในการรับหรือแสดงข้อมูลมากกว่าหนึ่งตัว เราสามารถใช้รูปแบบคิดกันได้ เช่น %c%c%c โดยไม่ต้องเว้นช่องว่าง เพราะตัวแปลภาษาสามารถแยกแยะข้อมูลได้อยู่แล้ว หรือจะมีอะไรมาแสดงก็ได้ เช่น %s/%s/%s
- ตัวอย่างการใช้เช่น

```
#include<stdio.h>
int main( )
{
    char ch1, ch2, ch3;

    printf("Enter three characters: ");
    scanf("%c%c%c", &ch1, &ch2, &ch3);
    printf("The characters are %c, %c, and %c\n", ch1, ch2, ch3);

    return 0;
}
```

การรับข้อมูลที่ละอักขระด้วย getchar()

- ฟังก์ชัน getchar() ใช้รับอักขระจากคีย์บอร์ด โดยสามารถรับได้ครั้งละหนึ่งอักขระเท่านั้น
- รูปแบบคำสั่ง: **variable = getchar();**
 - variable** คือชื่อตัวแปรชนิดอักขระ ที่จะใช้เก็บค่าของอักขระที่รับเข้ามาจากทางคีย์บอร์ด

- ตัวอย่างการใช้เช่น

```
#include<stdio.h>
int main( )
{
    char c;
    printf("Enter a letter: ");
    c = getchar();
    printf("You entered: %c", c);
    return 0;
}
```

การรับข้อมูลที่ละอักขระด้วย getch()

- ฟังก์ชัน getch() ใช้รับอักขระจากคีย์บอร์ด ได้ครั้งละหนึ่งอักขระ เช่นเดียวกับฟังก์ชัน getchar() แต่จะแตกต่างกันตรงที่การรับข้อมูลด้วยฟังก์ชัน getch() เมื่อผู้ใช้กรอกข้อมูลเข้ามา 1 อักขระแล้ว โปรแกรมจะทำงานต่อทันทีโดยไม่ต้องกดปุ่ม Enter และอักขระที่กรอกจะไม่แสดงขึ้นมาให้เห็นบนหน้าจอ
- รูปแบบคำสั่ง: **variable = getch();**

การรับข้อมูลเป็นข้อความด้วย gets()

- ฟังก์ชัน gets() ใช้รับข้อมูลที่เป็นข้อความผ่านทางคีย์บอร์ด
- รูปแบบคำสั่ง: **gets(str);**
 - str** คือ ตัวแปรที่จะใช้เก็บข้อความ ซึ่งเราต้องสร้างเตรียมไว้ก่อนที่จะเรียกฟังก์ชัน gets()

- ตัวอย่างการใช้เช่น

```
#include<stdio.h>
int main( )
{
    char message[30];
    printf("Enter a message: ");
    gets(message);
    printf("Your message is %s", message);
    return 0;
}
```