

บทที่ 5

การแสดงผลและการรับข้อมูล



การแสดงผลและการรับข้อมูล

ความหมายของการแสดงผล

การแสดงผล หมายถึง การสั่งให้คอมพิวเตอร์นำข้อมูล และผลลัพธ์ที่มีอยู่ในหน่วยความจำไปแสดงผลออกที่อุปกรณ์แสดงผล (Output Device) ของคอมพิวเตอร์ การแสดงผลที่อุปกรณ์แสดงผลอาจมีเพียงอุปกรณ์เดียว หรือหลายๆ อุปกรณ์พร้อมกันก็ได้ เช่น แสดงผลที่จอภาพ เครื่องพิมพ์ (Printer) ลำโพง แผ่นดิสก์

แสดงผลออกทางหน้าจอ เป็นการทำงานพื้นฐานหรือเรียกได้ว่าเป็นส่วนหนึ่งในการทำงานของทุกโปรแกรม คือ การแสดงผลข้อมูลออกทางจอภาพ โดยในภาษาซีนั้นการแสดงผลข้อมูลออกทางจอภาพใช้คำสั่งดังนี้

๑ คำสั่ง printf()

คำสั่ง printf() เป็นคำสั่งพื้นฐานในการแสดงผลข้อมูลทุกชนิดออกทางหน้าจอไม่ว่าจะเป็นจำนวนเต็ม (int) ทศนิยม (float) ข้อความ (string) หรืออักขระ (char) นอกจากนี้คำสั่งยังมีความยืดหยุ่นสูง โดยผู้เขียนสามารถกำหนดหรือจัดรูปแบบการแสดงผลให้มีระเบียบหรือเหมาะสมตามความต้องการได้อีกด้วย มีรูปแบบการเขียนคำสั่งดังนี้

```
printf("format",variable);
```

โดยที่

format

คือข้อความที่ต้องการแสดงผลออกทางหน้าจอ โดยข้อความนี้ต้องเขียนไว้ในเครื่องหมาย Double Quote (" ") ข้อความที่สามารถแสดงผลได้มีอยู่ ๒ ประเภท คือ ข้อความธรรมดา และค่ารหัสควบคุมที่ใช้แสดงค่าในตัวแปรออกมา

variable

คือข้อมูลที่จะแสดงผล ซึ่งอาจเป็นตัวแปรหรือนิพจน์ที่ต้องการนำค่าไปแสดงผล ถ้ามีมากกว่า ๑ ตัว ให้ใช้เครื่องหมายคอมม่า (,) คั่นระหว่างแต่ละตัว

รหัสควบคุมรูปแบบ (Format Code)

รหัสควบคุมรูปแบบใช้สำหรับการควบคุมการแสดงผลตัวแปร หรือนิพจน์ออกทางหน้าจอ โดยรหัสควบคุมรูปแบบในภาษาซีมีหลายชนิดด้วยกัน การนำไปใช้จะต้องพิจารณาให้เหมาะสมกับค่าของข้อมูลที่จะแสดงผลด้วย รหัสควบคุมรูปแบบแสดงได้ดังนี้

รหัสควบคุมรูปแบบ	การนำไปใช้งาน
%d	แสดงผลค่าของตัวแปรชนิดตัวเลขจำนวนเต็ม
%u	แสดงผลค่าของตัวแปรชนิดตัวเลขจำนวนเต็มบวก
%f	แสดงผลค่าของตัวแปรชนิดตัวเลขทศนิยม
%c	แสดงผลอักขระ ๑ ตัว (char)
%s	แสดงผลข้อความ (อักขระมากกว่า ๑ ตัว)

ตัวอย่างการใช้คำสั่ง printf() แสดงผลข้อความธรรมดาออกทางหน้าจอดังนี้

```
#include <stdio.h>
#include <conio.h>
main()
{
    printf("Hello World \n");
    printf("My School");
    getch();
}
```

ผลลัพธ์โปรแกรม

```
Hello World
My school
```

ตัวอย่างการใช้คำสั่ง printf() นำค่าของตัวแปรหรือนิพจน์ มาแสดงออกทางหน้าจอได้ดังนี้

```
#include <stdio.h>
#include <conio.h>
main()
{
    int x = 32;
    printf("Variable Value = %d",x);
    getch();
}
```

ผลลัพธ์โปรแกรม

Variable Value = 32

อักขระควบคุมการแสดงผล (Carriage Control)

การจัดข้อความที่จะแสดงออกมาทางหน้าจอให้ดูเป็นระเบียบสวยงาม เช่น ขึ้นบรรทัดใหม่ หลังแสดงข้อความ หรือเว้นช่องระหว่างข้อความ โดยใช้อักขระควบคุมการแสดงผลร่วมกับคำสั่ง printf() ดังต่อไปนี้

อักขระควบคุมการแสดงผล	ความหมาย
\n	ขึ้นบรรทัดใหม่
\t	เว้นช่องว่างเป็นระยะ ๑ แท็บ (๖ ตัวอักษร)
\r	กำหนดให้เคอร์เซอร์ไปอยู่ต้นบรรทัด
\f	เว้นช่องว่างเป็นระยะ ๑ หน้าจอ
\b	ลบอักขระสุดท้ายออก ๑ ตัว

การนำอักขระควบคุมการแสดงผลมาใช้ ต้องเขียนอักขระควบคุมการแสดงผลไว้ภายในเครื่องหมาย Double Quote (" ")

ตัวอย่างการใช้อักขระควบคุมการแสดงผล แสดงดังต่อไปนี้

```
#include <stdio.h>
#include <conio.h>
main()
{
    int a = 32 , b =8;
    printf ("Hello Student \n");
    printf ("Sum of %d + %d = \t %d \n",a,b,a+b);
    getch();
}
```

อธิบายโปรแกรม

- บรรทัดที่ ๑ : เป็นการบอกให้คอมไพเลอร์นำเฮดเดอร์ไฟล์ชื่อ `stdio.h` เข้ามาร่วมในการแปลผลด้วยเมื่อมีการเรียกใช้งานฟังก์ชัน `printf()` เพื่อแสดงข้อมูลออกทางจอภาพ และเนื่องจากส่วนของการประกาศฟังก์ชัน `printf()` ถูกบรรจุอยู่ในเฮดเดอร์ไฟล์ `stdio.h` ดังนั้นจึงจำเป็นต้องนำเฮดเดอร์ไฟล์ `stdio.h` เข้ามาร่วมในการแปลผลด้วย
- บรรทัดที่ ๓ : คือฟังก์ชัน `main()` ซึ่งเป็นฟังก์ชันหลักของโปรแกรม การทำงานของโปรแกรมภาษาซีจะเริ่มต้นที่ฟังก์ชันนี้
- บรรทัดที่ ๔ : เครื่องหมาย { ระบุจุดเริ่มต้นของฟังก์ชัน `main()`
- บรรทัดที่ ๕ : สร้างตัวแปรจำนวนเต็ม `a` และ `b` ให้มีค่า 32 และ 8 ตามลำดับ
- บรรทัดที่ ๖ : เรียกใช้ฟังก์ชัน `printf()` ซึ่งเป็นฟังก์ชันมาตรฐานของภาษาซีทำหน้าที่แสดงผลข้อมูลออกทางจอภาพ ในที่นี้จะแสดงข้อความ `Hello Student` แล้วขึ้นบรรทัดใหม่
- บรรทัดที่ ๗ : เรียกใช้ฟังก์ชัน `printf()` แสดงผลข้อความ `Sum of 32 + 8 = 40` ออกทางจอภาพ โดยแสดงค่าตัวแปร `a` แทนที่ตำแหน่งของ `%d` ตัวแรก แสดงค่าของตัวแปร `b` แทนที่ตำแหน่งของ `%d` ตัวที่สอง เว้นช่องว่างเป็นระยะ ๑ แท็บ เมื่อเจอรหัสควบคุม `\t` และแสดงค่าของนิพจน์ `a+b` แทนที่ตำแหน่ง `%d` ตัวสุดท้าย แล้วขึ้นบรรทัดใหม่เพราะมีการใช้ `\n`
- บรรทัดที่ ๘ : เครื่องหมาย } ระบุจุดสิ้นสุดของฟังก์ชัน `main()`

ผลลัพธ์โปรแกรม

```
Hello Student
Sum of 32 + 8 = 40
```

๒ คำสั่ง `putchar()`

คำสั่ง `putchar()` เป็นคำสั่งใช้สำหรับแสดงผลอักขระโดยเฉพาะออกทางหน้าจอ มีรูปแบบการเขียนคำสั่งดังนี้

```
putchar(char);
```

โดยที่ `char` เป็นตัวแปรชนิดอักขระ (`char`) ที่เขียนภายในเครื่องหมาย Single Quote (' ')

ตัวอย่างการแสดงผลโดยใช้คำสั่ง putchar()

```
#include <stdio.h>
#include <conio.h>
main()
{
    char first = 'O';
    putchar (first);
    putchar ('K');
    getch();
}
```

อธิบายโปรแกรม

- บรรทัดที่ ๑ : เป็นการบอกให้คอมไพเลอร์นำเฮดเดอร์ไฟล์ชื่อ stdio.h เข้ามาร่วมในการแปลผลด้วยเมื่อมีการเรียกใช้งานฟังก์ชัน putchar()
- บรรทัดที่ ๓ : ฟังก์ชัน main() ซึ่งเป็นฟังก์ชันหลักของโปรแกรม การทำงานของโปรแกรมภาษาซีจะเริ่มต้นที่ฟังก์ชันนี้
- บรรทัดที่ ๔ : เครื่องหมาย { ระบุจุดเริ่มต้นของฟังก์ชัน main()
- บรรทัดที่ ๕ : สร้างตัวแปร first เก็บอักษร O

- บรรทัดที่ ๖ : เรียกใช้ฟังก์ชัน putchar() แสดงอักษร O ที่เก็บไว้ในตัวแปร first ออกมาทางหน้าจอ
- บรรทัดที่ ๗ : เรียกใช้ฟังก์ชัน putchar() แสดงอักษร K ออกมาทางหน้าจอ
- บรรทัดที่ ๘ : เครื่องหมาย } ระบุจุดสิ้นสุดของฟังก์ชัน main()

ผลลัพธ์โปรแกรม



๗. คำสั่ง puts()

คำสั่ง puts() เป็นคำสั่งใช้สำหรับแสดงข้อความออกทางหน้าจอ โดยมีรูปแบบการเขียนคำสั่งดังนี้

```
puts(str);
```

โดยที่ str เป็นตัวแปรที่เก็บข้อมูลชนิดข้อความ (string) ที่เขียนภายในเครื่องหมาย Double Quote (" ")

ตัวอย่างการแสดงผลโดยใช้คำสั่ง puts()

```
#include <stdio.h>
#include <conio.h>
main()
{
    char message[ ] = "C Language";
    puts (message);
    puts ("Easy and Fun");
    getch();
}
```

อธิบายโปรแกรม

- บรรทัดที่ ๑ : เป็นการบอกให้คอมไพเลอร์นำเฮดเดอร์ไฟล์ชื่อ stdio.h เข้ามาร่วมในการแปลผลด้วยเมื่อมีการเรียกใช้งานฟังก์ชัน puts()
- บรรทัดที่ ๓ : คือฟังก์ชัน main() ซึ่งเป็นฟังก์ชันหลักของโปรแกรม การทำงานของโปรแกรมภาษาซีจะเริ่มต้นที่ฟังก์ชันนี้
- บรรทัดที่ ๔ : เครื่องหมาย { ระบุจุดเริ่มต้นของฟังก์ชัน main()
- บรรทัดที่ ๕ : สร้างตัวแปร message เก็บข้อความ C Language
- บรรทัดที่ ๖ : เรียกใช้ฟังก์ชัน puts() แสดงข้อความที่เก็บไว้ในตัวแปร message ออกมาทางหน้าจอหลังจากแสดงข้อความแล้วจะขึ้นบรรทัดใหม่ให้ด้วย
- บรรทัดที่ ๗ : เรียกใช้ฟังก์ชัน puts() แสดงข้อความ Easy and Fun ออกมาทางหน้าจอหลังจากนั้นจะขึ้นบรรทัดใหม่โดยเลื่อนเคอร์เซอร์ลงไปอยู่บรรทัดล่าง
- บรรทัดที่ ๘ : เครื่องหมาย } ระบุจุดสิ้นสุดของฟังก์ชัน main()

ผลลัพธ์โปรแกรม

```
C Language
Easy and Fun
```



คำสั่งรับข้อมูล

การทำงานของโปรแกรมส่วนใหญ่มักจะเป็นการเชื่อมโยงกับผู้ใช้แบบ ๒ ทิศทาง คือ ทั้งการแสดงผลการทำงานออกทางหน้าจอ และการรับข้อมูลจากผู้ใช้เข้ามาทางคีย์บอร์ด เพื่อร่วมในการประมวลผลของโปรแกรม ซึ่งการรับข้อมูลนั้นจะทำได้โดยใช้คำสั่งดังนี้

๑. รับข้อมูลจากคีย์บอร์ดด้วยคำสั่ง scanf()

ในภาษาซี การรับข้อมูลจากคีย์บอร์ดสามารถทำได้โดยการเรียกใช้ฟังก์ชัน scanf() ซึ่งเป็นฟังก์ชันมาตรฐานสำหรับรับข้อมูลจากคีย์บอร์ด โดยสามารถรับข้อมูลได้ทุกประเภท ไม่ว่าจะเป็นจำนวนเต็ม ทศนิยม อักขระ หรือข้อความ โดยมีรูปแบบคำสั่งดังนี้

```
scanf("format", &variable);
```

โดยที่

format

เป็นการใช้รหัสควบคุมรูปแบบ (Format Code) เพื่อกำหนดชนิดของข้อมูลที่จะรับเข้ามาจากคีย์บอร์ด โดยรหัสควบคุมรูปแบบใช้ชุดเดียวกับคำสั่ง printf()

variable

คือตัวแปรที่จะใช้เก็บค่าข้อมูลที่รับเข้ามาจากคีย์บอร์ด โดยชนิดของตัวแปรจะต้องตรงกับรหัสควบคุมรูปแบบที่กำหนดไว้ นอกจากนี้หน้าชื่อของตัวแปรจะต้องนำหน้าด้วยเครื่องหมาย & ยกเว้นตัวแปรชนิดข้อความ (String) เท่านั้นที่ไม่ต้องนำหน้าด้วยเครื่องหมาย &

ตัวอย่างการใช้คำสั่ง scanf()

```
#include <stdio.h>
#include <conio.h>
main()
{
    int age;
    printf("How old are you?");
    scanf("%d",&age);
    printf("you are %d years old. \n",age);
    getch();
}
```

อธิบายโปรแกรม

- บรรทัดที่ ๑ : เป็นการบอกให้คอมไพเลอร์นำเฮดเดอร์ไฟล์ชื่อ `stdio.h` เข้ามาร่วมในการแปลผลด้วยเมื่อมีการเรียกใช้งานฟังก์ชัน `printf()` เพื่อแสดงข้อความออกทางหน้าจอ และฟังก์ชัน `scanf()` รับข้อความจากคีย์บอร์ด
- บรรทัดที่ ๓ : คือฟังก์ชัน `main()` ซึ่งเป็นฟังก์ชันหลักของโปรแกรม การทำงานของโปรแกรมภาษาซีจะเริ่มต้นที่ฟังก์ชันนี้
- บรรทัดที่ ๔ : เครื่องหมาย { ระบุจุดเริ่มต้นของฟังก์ชัน `main()`
- บรรทัดที่ ๕ : สร้างตัวแปร `age` เพื่อเก็บค่าจำนวนเต็ม
- บรรทัดที่ ๖ : เรียกใช้ฟังก์ชัน `printf()` แสดงข้อความ `How old are you?` ออกมาทางหน้าจอ
- บรรทัดที่ ๗ : เรียกใช้ฟังก์ชัน `scanf()` เพื่อรับค่าจากคีย์บอร์ดเข้ามา และใช้รหัสควบคุมแบบ `%d` แสดงว่าข้อมูลที่รับเข้ามาเป็นเลขจำนวนเต็มซึ่งจะถูกนำไปเก็บไว้ที่ตัวแปร `age`
- บรรทัดที่ ๘ : เรียกใช้ฟังก์ชัน `printf()` เพื่อแสดงข้อความที่กำหนด และข้อความที่รับค่าจากตัวแปร `age`
- บรรทัดที่ ๑๐ : เครื่องหมาย } ระบุจุดสิ้นสุดของฟังก์ชัน `main()`

ผลลัพธ์โปรแกรม

เมื่อรันโปรแกรม จะเห็นข้อความ `How old are you?` ปรากฏขึ้นมาพร้อมกับเคอร์เซอร์รอให้กรอกข้อมูล

```
How old are you ? _
```

เมื่อกรอกตัวเลขจำนวนเต็ม ๑ จำนวน ในตัวอย่างกำหนดให้เท่ากับ ๒๕ แล้วกดปุ่ม Enter โปรแกรมจึงจะทำงานต่อไป โดยแสดงออกมาดังตัวอย่าง

```
How old are you ? 25
You are 25 years old.
```


๒. รับข้อมูลที่ละตัวอักษรด้วยคำสั่ง `getchar()`

ฟังก์ชัน `getchar()` เป็นฟังก์ชันที่ใช้สำหรับรับข้อมูลชนิดอักขระเข้ามาจากคีย์บอร์ด โดยรับข้อมูลได้ครั้งละ ๑ ตัวอักษรเท่านั้น และกดปุ่ม Enter จึงจะมีการแสดงผลทางหน้าจอ ซึ่งมีรูปแบบคำสั่งดังนี้

```
variable = getchar();
```

โดยที่ `variable` คือ ชื่อของตัวแปรชนิดอักขระ ซึ่งจะใช้เก็บค่าของอักขระที่รับเข้ามาจากคีย์บอร์ด

ตัวอย่างการใช้คำสั่ง `getchar()`

```
1 #include <stdio.h>
2 main()
3 {
4     char answer;
5     clrscr();
6     printf("Enter character (Y or N) : ");
7     answer = getchar();
8     printf("You insert : %c",answer);
9 }
```

อธิบายโปรแกรม

บรรทัดที่ ๑ : เป็นการบอกให้คอมไพเลอร์นำเฮดเดอร์ไฟล์ชื่อ `stdio.h` เข้ามาร่วมในการแปลผลเมื่อมีการเรียกใช้งานฟังก์ชัน `printf()` เพื่อแสดงข้อความออกทางหน้าจอ และฟังก์ชัน `getchar()` รับข้อความจากคีย์บอร์ดครั้งละ ๑ ตัวอักษร

บรรทัดที่ ๒ : คือฟังก์ชัน `main()` ซึ่งเป็นฟังก์ชันหลักของโปรแกรม การทำงานของโปรแกรมภาษาซีจะเริ่มต้นที่ฟังก์ชันนี้

บรรทัดที่ ๓ : เครื่องหมาย { ระบุจุดเริ่มต้นของฟังก์ชัน `main()`

บรรทัดที่ ๔ : สร้างตัวแปร `answer` เพื่อเก็บค่าอักขระ

บรรทัดที่ ๕ : เรียกใช้ฟังก์ชัน `clrscr()` เพื่อลบข้อความบนหน้าจอ

- บรรทัดที่ ๖ : เรียกใช้ฟังก์ชัน printf() แสดงข้อความ Enter character (Y or N) :
ออกมาทางหน้าจอ
- บรรทัดที่ ๗ : เรียกใช้ฟังก์ชัน getchar() เพื่อรับค่าจากคีย์บอร์ด ๑ ตัวอักษร
โดยเก็บไว้ในตัวแปร answer
- บรรทัดที่ ๘ : เรียกใช้ฟังก์ชัน printf() เพื่อแสดงข้อความที่กำหนด และตัวอักษร
ที่รับค่าจากตัวแปร answer
- บรรทัดที่ ๙ : เครื่องหมาย } ระบุจุดสิ้นสุดของฟังก์ชัน main()

ผลลัพธ์โปรแกรม

เมื่อรันโปรแกรม จะเห็นข้อความ Enter character (Y or N) : ปรากฏขึ้นมาพร้อมกับ
เคอร์เซอร์รอรับตัวอักษร

```
Enter character (Y or N) : _
```

เมื่อกรอกตัวอักษร Y หรือ N เพียงตัวใดตัวหนึ่งเท่านั้น แล้วกดปุ่ม Enter โปรแกรมจะ
แสดงออกมาดังตัวอย่าง

```
Enter character (Y or N) : Y  
You insert : Y
```

๓. รับข้อมูลที่ละตัวอักษรด้วยคำสั่ง getch()

คำสั่ง getch() เป็นอีกหนึ่งฟังก์ชันที่รับข้อมูลชนิดอักขระจากคีย์บอร์ด ได้ครั้งละ ๑ ตัว
อักษรเหมือนกับฟังก์ชัน getchar() แต่การรับข้อมูลด้วยฟังก์ชัน getch() เมื่อผู้ใช้กรอกข้อมูล ๑
ตัวอักษรแล้วโปรแกรมจะทำงานต่อทันทีโดยไม่ต้องกดปุ่ม Enter และตัวอักษรที่กรอกเข้ามาจะไม่
แสดงขึ้นมาให้เห็นบนหน้าจอ โดยมีรูปแบบการเขียนฟังก์ชันดังนี้

```
variable = getch();
```

โดยที่ variable คือ ชื่อของตัวแปรชนิดอักขระ ซึ่งจะใช้เก็บค่าของอักขระที่รับเข้ามา
จากคีย์บอร์ด

ตัวอย่างการใช้คำสั่ง getch()

```
1  #include <stdio.h>
2  main()
3  {
4      char n;
5      clrscr();
6      printf("Enter character (Y or N) : ");
7      n = getch();
8      printf("You insert : %c",n);
9  }
```

อธิบายโปรแกรม

- บรรทัดที่ ๑ : เป็นการบอกให้คอมไพเลอร์นำเฮดเดอร์ไฟล์ชื่อ `stdio.h` เข้ามาร่วมในการแปลผลด้วยเมื่อมีการเรียกใช้งานฟังก์ชัน `printf()` เพื่อแสดงข้อความออกทางหน้าจอ และฟังก์ชัน `getch()` รับข้อความจากคีย์บอร์ดครั้งละ ๑ ตัวอักษร
- บรรทัดที่ ๒ : คือฟังก์ชัน `main()` ซึ่งเป็นฟังก์ชันหลักของโปรแกรม การทำงานของโปรแกรมภาษาซีจะเริ่มต้นที่ฟังก์ชันนี้
- บรรทัดที่ ๓ : เครื่องหมาย { ระบุจุดเริ่มต้นของฟังก์ชัน `main()`
- บรรทัดที่ ๔ : สร้างตัวแปร `n` เพื่อเก็บค่าอักขระ
- บรรทัดที่ ๕ : เรียกใช้ฟังก์ชัน `clrscr()` เพื่อลบข้อความบนหน้าจอ
- บรรทัดที่ ๖ : เรียกใช้ฟังก์ชัน `printf()` แสดงข้อความ `Enter character (Y or N) :` ออกมาทางหน้าจอ
- บรรทัดที่ ๗ : เรียกใช้ฟังก์ชัน `getch()` เพื่อรับค่าจากคีย์บอร์ด ๑ ตัวอักขระโดยเก็บไว้ที่ตัวแปร `n`
- บรรทัดที่ ๘ : เรียกใช้ฟังก์ชัน `printf()` เพื่อแสดงข้อความที่กำหนด และตัวอักษรที่รับค่าจากตัวแปร `n`
- บรรทัดที่ ๙ : เครื่องหมาย } ระบุจุดสิ้นสุดของฟังก์ชัน `main()`

ผลลัพธ์โปรแกรม

เมื่อรันโปรแกรม จะเห็นข้อความ Enter character (Y or N) : ปรากฏขึ้นมาพร้อมกับเคอร์เซอร์รับตัวอักษร

```
Enter character (Y or N) : _
```

เมื่อกรอกตัวอักษร Y หรือ N เพียงตัวใดตัวหนึ่งเท่านั้น โปรแกรมก็จะทำงานต่อโดยไม่ต้องกด Enter

```
Enter character (Y or N) : You insert : Y
```

๔. รับข้อมูลเป็นข้อความด้วย gets()

```
gets(str);
```

คำสั่ง gets() เป็นอีกหนึ่งฟังก์ชันที่รับข้อมูลเป็นข้อความจากคีย์บอร์ด ซึ่งเป็นฟังก์ชันสำหรับข้อความโดยเฉพาะ โดยมีรูปแบบการเขียนฟังก์ชันดังนี้

โดยที่ str เป็นตัวแปรที่ใช้สำหรับเก็บข้อความ ซึ่งจะต้องสร้างเตรียมไว้ก่อนที่จะเรียกใช้งานฟังก์ชัน gets()

```
1 #include <stdio.h>
2 main()
3 {
4     char msg[20];
5     clrscr();
6     printf("Enter your message: ");
7     gets(msg);
8     printf("Your message is : %s",msg);
9 }
```

ตัวอย่างการใช้คำสั่ง gets()

อธิบายโปรแกรม

- บรรทัดที่ ๑ : เป็นการบอกให้คอมไพเลอร์นำเฮดเดอร์ไฟล์ชื่อ `stdio.h` เข้ามาร่วมในการแปลผลด้วยเมื่อมีการเรียกใช้งานฟังก์ชัน `printf()` เพื่อแสดงข้อความออกทางหน้าจอ และฟังก์ชัน `gets()` รับข้อมูลชนิดข้อความจากคีย์บอร์ด
- บรรทัดที่ ๒ : คือฟังก์ชัน `main()` ซึ่งเป็นฟังก์ชันหลักของโปรแกรม การทำงานของโปรแกรมภาษาซีจะเริ่มต้นที่ฟังก์ชันนี้
- บรรทัดที่ ๓ : เครื่องหมาย { ระบุจุดเริ่มต้นของฟังก์ชัน `main()`
- บรรทัดที่ ๔ : สร้างตัวแปรชนิด `char` ชื่อ `msg[20]` เพื่อเก็บชนิดข้อความความยาว ๒๐ ตัวอักษร
- บรรทัดที่ ๕ : เรียกใช้ฟังก์ชัน `clrscr()` เพื่อลบข้อความบนหน้าจอ
- บรรทัดที่ ๖ : เรียกใช้ฟังก์ชัน `printf()` แสดงข้อความ `Enter your message: ออกมาทางหน้าจอ`
- บรรทัดที่ ๗ : เรียกใช้ฟังก์ชัน `gets()` เพื่อรับข้อความและเก็บไว้ที่ตัวแปร `msg` โดยผู้ใช้จะต้องเป็นผู้พิมพ์ข้อความและจะต้องกด `Enter`
- บรรทัดที่ ๘ : เรียกใช้ฟังก์ชัน `printf()` เพื่อแสดงข้อความ `Your message is : และข้อความที่รับค่าจากตัวแปร msg ออกทางหน้าจอ`
- บรรทัดที่ ๙ : เครื่องหมาย } ระบุจุดสิ้นสุดของฟังก์ชัน `main()`

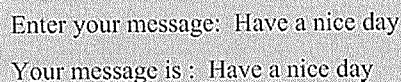
ผลลัพธ์โปรแกรม

เมื่อรันโปรแกรม จะเห็นข้อความ `Enter your message :` ปรากฏขึ้นมาพร้อมกับเคอร์เซอร์รอรับข้อความ



Enter your message: _

เมื่อผู้ใช้พิมพ์ข้อความว่า `Have a nice day` และกด `Enter` โปรแกรมก็จะทำงานโดยแสดงผลที่หน้าจอต่อมาว่า



Enter your message: Have a nice day
Your message is : Have a nice day

แบบฝึกหัดที่ 5.1

1. จงหาผลลัพธ์จากการเขียน โปรแกรมต่อไปนี้

```
#include<stdio.h>
#include<conio.h>
main()
{
    printf("computer\n");
    printf("THAILAND");
    getch();
}
```

ผลลัพธ์

2. จงหาผลลัพธ์จากการเขียน โปรแกรมต่อไปนี้

```
#include<stdio.h>
#include<conio.h>
main()
{
    int x=15,y=6;
    printf("sum\t");
    printf("%d+%d=%d",x,y,x*y);
    getch();
}
```

ผลลัพธ์

3. จงหาผลลัพธ์จากการเขียน โปรแกรมต่อไปนี้

```
#include<stdio.h>
#include<conio.h>
main()
{
    char name[10];
    printf("please enter your name:");
    scanf("%s",&name);
    printf("my name is: %s",name);
    getch();
}
```

ผลลัพธ์

4. จงหาผลลัพธ์จากการเขียนโปรแกรมต่อไปนี้

```
#include<stdio.h>
#include<conio.h>
main()
{
    int x=17,y=10;
    printf("\tX\tY\n");
    printf("\t%d\t%d",x,y);
    getch();
}
```

ผลลัพธ์

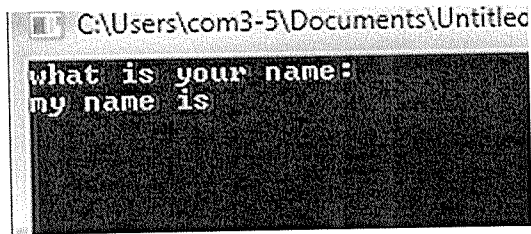
5. จงหาผลลัพธ์จากการเขียนโปรแกรมต่อไปนี้

```
#include<stdio.h>
#include<conio.h>
main()
{
    int x,y;
    x=10;
    printf("x1=%d\n",(x+2)+(5*4)+(9/8));
    y=20;
    printf("y1=%d\n",y+5-2+5-10);
    y=10;
    printf("y2=%d\n",(y*10)+12*10+5*10);
    getch();
}
```

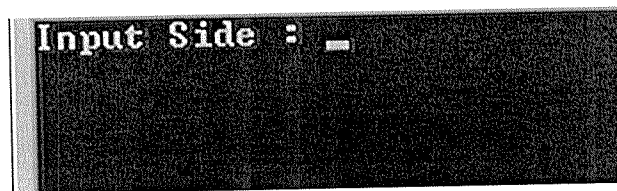
ผลลัพธ์

แบบฝึกหัดที่ 5.2

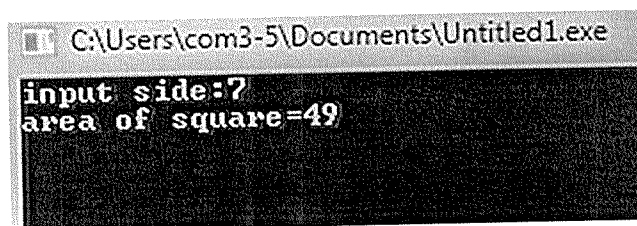
1. จงเขียน โปรแกรม ให้แสดงผลหน้าจอตามตัวอย่างข้างล่างนี้



2. จงเขียน โปรแกรม การคำนวณหาพื้นที่สี่เหลี่ยมจตุรัส โดยกำหนดตัวแปร *side* เป็นเลขจำนวนเต็ม แทนด้านของสี่เหลี่ยม สูตรในการหาพื้นที่รูปสี่เหลี่ยมจตุรัส คือ ด้านคูณด้าน และกำหนดให้รับค่าของตัวแปร *side* ผ่านทางแป้นพิมพ์ และแสดงค่าผลลัพธ์ผ่านทางจอภาพ ดังตัวอย่างหน้าจอข้างล่างนี้

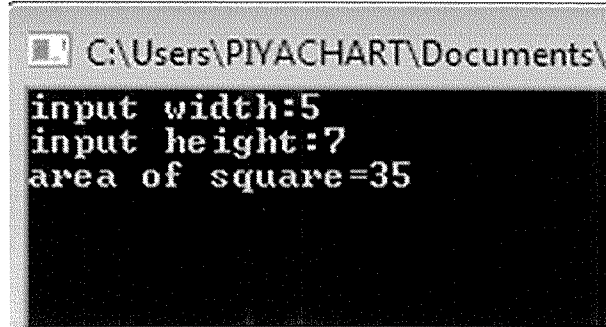


รูปที่ 1 แสดงการรับข้อมูลด้านของสี่เหลี่ยมผ่านทางแป้นพิมพ์



รูปที่ 2 แสดงตัวอย่างแสดงผลเมื่อป้อนข้อมูลด้าน (Side) เท่ากับ 7 และผลลัพธ์ที่ได้คือ 49

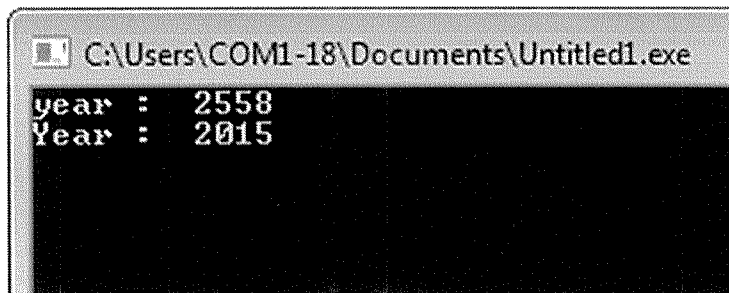
3. จงเขียนโปรแกรม การคำนวณหาพื้นที่สี่เหลี่ยมผืนผ้า โดยกำหนดตัวแปร *width* เป็นความกว้างของสี่เหลี่ยมผืนผ้า และกำหนดตัวแปร *height* เป็นความสูงของสี่เหลี่ยมผืนผ้า สูตรในการหาพื้นที่รูปสี่เหลี่ยมผืนผ้า คือ กว้างคูณยาว และกำหนดให้รับค่าของตัวแปร *width* และ *height* ผ่านทางแป้นพิมพ์ และแสดงค่าผลลัพธ์ผ่านทางจอภาพ ดังตัวอย่างหน้าจอข้างล่างนี้



```
C:\Users\PIYACHART\Documents\  
input width:5  
input height:7  
area of square=35
```

ตัวอย่างแสดงผลเมื่อป้อนข้อมูลความกว้างเท่ากับ 5 ความสูงเท่ากับ 7 และผลลัพธ์ที่ได้คือ 35

4. จงเขียนโปรแกรมรับค่าปี พ.ศ. ทางคีย์บอร์ด แล้วแสดงผลเป็นปี ค.ศ.



```
C:\Users\COM1-18\Documents\Untitled1.exe  
year : 2558  
Year : 2015
```