

การรับข้อมูลเข้าและแสดงผลข้อมูล

ในภาษาซีมีไลบรารีฟังก์ชัน (Alexander, A., Online) ที่เกี่ยวกับการรับข้อมูลเข้าและการแสดงผลข้อมูล ได้แก่ฟังก์ชันที่ทำการรับข้อมูลและส่งข้อมูลคือฟังก์ชัน `getch()` `getche()` `getchar()` และฟังก์ชัน `putchar()` ทำหน้าที่รับส่งตัวอักษรเข้า-ออกคอมพิวเตอร์ ส่วนฟังก์ชัน `gets()` ฟังก์ชัน `puts()` ทำหน้าที่รับ-แสดงสตริง และฟังก์ชัน `printf()` ฟังก์ชัน `scanf()` ความสามารถรับ-ส่งอักขระ ตัวเลข และสตริงได้ (Steven และ Lutfar, 2006) (Brian W. K., Online) เป็นต้น

การเรียกใช้ฟังก์ชันแต่ละฟังก์ชันต้องทราบว่าฟังก์ชันที่เรียกใช้ถูกนิยามหรือต้องทราบว่าอยู่ในไลบรารีใด ก่อนใช้งานฟังก์ชันนั้น ต้องสั่งให้โปรแกรมภาษาซีเชื่อมต่อ (ลิงค์) ด้วยคำสั่ง `#include<fileheader.h>` ก่อนเสมอ

การเรียกใช้ฟังก์ชันการรับข้อมูลเข้าและการแสดงผลข้อมูล สามารถเรียกได้ทุกจุดของโปรแกรม โดยการพิมพ์ชื่อฟังก์ชัน ตามด้วยรายการอาร์กิวเมนต์ที่อยู่ในวงเล็บ โดยอาร์กิวเมนต์จะเก็บข้อมูลที่ต้องการส่งให้ฟังก์ชันรับข้อมูลเข้าและแสดงผลลัพธ์ตามความต้องการของผู้เขียน โปรแกรม ฟังก์ชันบางฟังก์ชันไม่ต้องการอาร์กิวเมนต์ก็สามารถทำงานได้

ฟังก์ชัน getch();

เป็นฟังก์ชันที่ใช้ในการรอรับการป้อนแป้นพิมพ์ (Key Board) 1 ตัวอักษร และจะไม่แสดงตัวอักษรที่ป้อนบนจอภาพ ฟังก์ชันนี้ถูกนิยามไว้ใน conio.h

รูปแบบที่ 1 getch();

หมายถึง หยุดรอการป้อนแป้นพิมพ์ 1 ตัวอักษร

รูปแบบที่ 2 ตัวแปร = getch();

หมายถึง หยุดรอการป้อนแป้นพิมพ์ 1 ตัวอักษรมาเก็บไว้ที่ตัวแปร

ฟังก์ชัน getche();

เป็นฟังก์ชันที่ใช้ในการรอรับการป้อนแป้นพิมพ์ 1 ตัวอักษรและจะแสดงตัวอักษรที่ป้อนบนจอภาพ ฟังก์ชันนี้ถูกนิยามไว้ใน conio.h

รูปแบบ getche();

หมายถึง หยุดรอการป้อนแป้นพิมพ์ 1 ตัวอักษรมาเก็บไว้ที่ตัวแปรและแสดงผล

ฟังก์ชัน `getchar()`;

เป็นฟังก์ชันที่ใช้ในการรอรับการป้อนเป็นพิมพ์ และจะแสดงตัวอักขระที่ป้อนบนจอภาพ ฟังก์ชันนี้ถูกนิยามไว้ใน `conio.h`

รูปแบบ `char var = getchar();`

หมายถึง หยุดรอการป้อนเป็นพิมพ์ตัวอักษรมาเก็บไว้ในตัวแปร `var`

ฟังก์ชัน `getchar()`;

เป็นฟังก์ชันที่ใช้ในการรอรับการป้อนเป็นพิมพ์ และจะแสดงตัวอักขระที่ป้อนบนจอภาพ ฟังก์ชันนี้ถูกนิยามไว้ใน `conio.h`

รูปแบบ `char var = getchar();`

หมายถึง หยุดรอการป้อนเป็นพิมพ์ตัวอักษรมาเก็บไว้ในตัวแปร `var`

ฟังก์ชัน `putchar()`;

เป็นฟังก์ชันที่ใช้ในการแสดงตัวอักขระบนจอภาพ ฟังก์ชันนี้ถูกนิยามไว้ใน `conio.h`

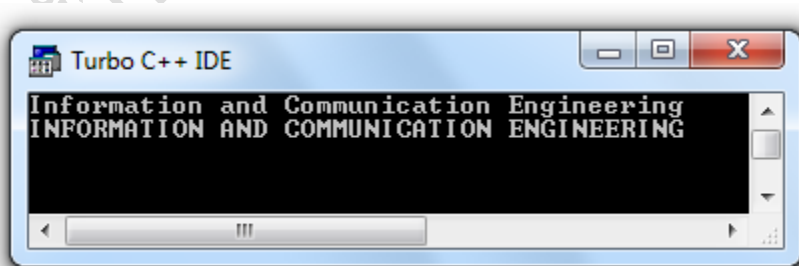
รูปแบบ `putchar(char var);`

หมายถึง แสดงตัวอักขระในตัวแปร `var` ออกสู่หน้าจอภาพ

ตัวอย่าง โปรแกรมที่ 4.1

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char letter[80];
    int count, tag;
    for(count=0; (letter[count]=getchar()) != '\n';++count);
    tag = count;
    for(count=0; count<tag;++count)
        putchar(toupper(letter[count]));
    getch();
}
```

ผลการทำงานของ โปรแกรม



ฟังก์ชัน printf();

เป็นฟังก์ชันที่ใช้แสดงข้อมูลที่เป็น ตัวเลข ตัวอักษร และสตริง ออกสู่หน้าจอภาพ หมายความว่า ฟังก์ชัน printf() ทำหน้าที่ย้ายข้อมูลจากหน่วยความจำของเครื่องคอมพิวเตอร์สู่หน่วยแสดงผลคอมพิวเตอร์ และฟังก์ชัน printf() ถูกนิยามไว้ใน stdio.h

รูปแบบมาตรฐาน	printf(control string, arg1, arg2, ... argn);
รูปแบบที่ 1	printf("ข้อมูล");
ตัวอย่างเช่น	printf("Hello C++...!");
ผลการรันโปรแกรม	แสดงข้อมูลที่เป็นข้อความ Hello C++...! แสดงสู่ออกทางหน้าจอภาพ
รูปแบบที่ 2	printf("รหัสควบคุมข้อมูล", อาร์กิวเมนต์);
ตัวอย่างเช่น	A = 55; printf("%d",A); //อาร์กิวเมนต์เป็นตัวแปร A
หมายถึง	แสดงข้อมูลค่าของตัวแปร A ออกมาแสดงผลในรูปแบบของเลขจำนวนเต็มตาม การกำหนดของรหัสควบคุมข้อมูล (%d)
ผลการรันโปรแกรม	แสดง 55 ออกทางจอภาพ

รูปแบบที่ 3

```
printf("ข้อมูล รหัสควบคุมข้อมูล", อาร์กิวเมนต์);
```

ตัวอย่างเช่น

```
X = 999;
```

```
printf("Information in X = %d", X);
```

หมายถึง

แสดงข้อมูลคำว่า Information in X = ตามด้วยค่าของตัวแปร X ออกมา แสดงผลในรูปแบบของเลขจำนวนเต็ม ตามรหัสควบคุมข้อมูล (%d)

ผลการรันโปรแกรม

```
Information in X = 999
```

รหัสควบคุมข้อมูล

เป็นรหัสควบคุมข้อมูลซึ่งเป็นการแสดงรูปแบบ (Format Code) ที่ใช้ในการแสดงผลต่อหน้าจอภาพ โดยรูปแบบการสั่งจะต้องอยู่หลังเครื่องหมาย % รหัสควบคุมข้อมูลที่ใช้แสดง ดังตารางที่ 4.1

ตารางที่ 4.1 แสดงรหัสควบคุมข้อมูล

รหัสควบคุม	ความหมาย
%c	แสดงข้อมูลในรูปตัวอักษร 1 ตัว (character)
%d	แสดงข้อมูลในรูปตัวเลขจำนวนเต็ม (decimal)
%e	แสดงข้อมูลในรูปทศนิยม ที่มีเลขชี้กำลัง (exponential)
%f	แสดงข้อมูลในรูปของตัวเลขทศนิยม (float)
%g	แสดงข้อมูลในรูปเลขทศนิยม รูปแบบ e หรือ f ขึ้นอยู่กับข้อมูล โดยจะไม่แสดงเลขศูนย์ที่อยู่หลังเลขทศนิยมตัวสุดท้าย
%i	แสดงข้อมูลในรูปของตัวเลขจำนวนเต็มฐานสิบ รูปแบบคิดเครื่องหมาย (integer) คือแสดงตัวเลขเป็นชนิด signed
%o	แสดงข้อมูลในรูปของเลขฐาน 8 (octal)
%x	แสดงข้อมูลในรูปของเลขฐาน 16 (hexa decimal)
%u	แสดงข้อมูลในรูปจำนวนเต็มบวก หรือแสดงเลขจำนวนเต็มไม่คิดเครื่องหมาย (unsigned)
%s	แสดงข้อมูลในรูปของข้อความ (String) คือ แสดงข้อมูลที่เป็นตัวอักษรมากกว่าหนึ่งตัว
%p	แสดงข้อมูลในรูปที่อยู่ของตัวแปร (address)
%%	แสดงเครื่องหมาย %

รหัสควบคุมการแสดงผล

รหัสควบคุมการแสดงผล สามารถเขียนแทรกในคำสั่งของ printf(); ตามความต้องการของนักเขียนโปรแกรม ซึ่งสามารถเขียนก่อน ส่วนของรหัสควบคุมหรือ หลังรหัสควบคุมเป็นการจัดรูปแบบการแสดงผลข้อมูลสู่หน้าจอภาพให้เป็นระเบียบและสื่อความหมายระหว่างตัวโปรแกรมกับผู้ใช้โปรแกรม ได้ถูกต้อง รหัสควบคุมการแสดงผล ตารางที่ 4.2

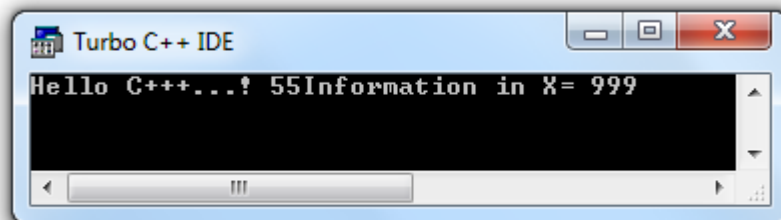
ตารางที่ 4.2 รหัสควบคุมการแสดงผล

รหัส	ความหมาย
\n	แสดงขึ้นบรรทัดใหม่ (new line)
\t	แสดงตัวอักษรย่อหน้ากระดาษ (Tab)
\r	สั่งให้ตัวชี้ตำแหน่ง (Cursor) กลับไปอยู่ที่ต้นบรรทัด
\f	สั่งให้เว้นการแสดงผลไป 1 หน้า (feed)
\b	สั่งให้ตัวชี้ตำแหน่ง (Cursor) ถอยหลังกลับไปลบตัวอักษร 1 ตำแหน่ง (back space)

ตัวอย่างโปรแกรม 4.2 เป็นการสั่งให้แสดงข้อมูลที่เป็นข้อความ โดยตรง การแสดงข้อมูลที่เป็นตัวเลขจำนวนเต็มสิบ โดยผ่านการแสดงตัวแปร การแสดงข้อความก่อนแสดงข้อมูลที่เป็นตัวเลขจำนวนเต็มโดยผ่านการควบคุมด้วยรหัสควบคุม แต่ยังไม่มีการควบคุมการแสดงผล หรือไม่มีการจัดรูปแบบการแสดงผล


```
#include<stdio.h>
#include<conio.h>
void main()
{
    int A=55; X=999;
    clrscr();
    printf("Hello C+++...! ");
    printf("%d",A);
    printf("Information in X= %d",X);
    getch();
}
```

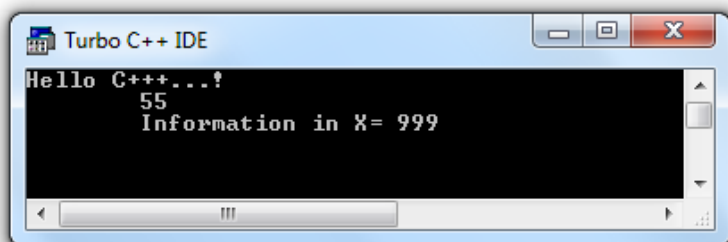
ผลการทำงานของโปรแกรม



ตัวอย่าง โปรแกรม 4.3 เป็นการสั่งให้แสดงข้อมูลโดยมีการจัดรูปแบบการแสดงผล

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int A=55, X=999;
    clrscr();
    printf("Hello C+++...! \n");
    printf("\t%d\n",A);
    printf("\tInformation in X= %d",X);
    getch();
}
```

ผลการทำงานของ โปรแกรม

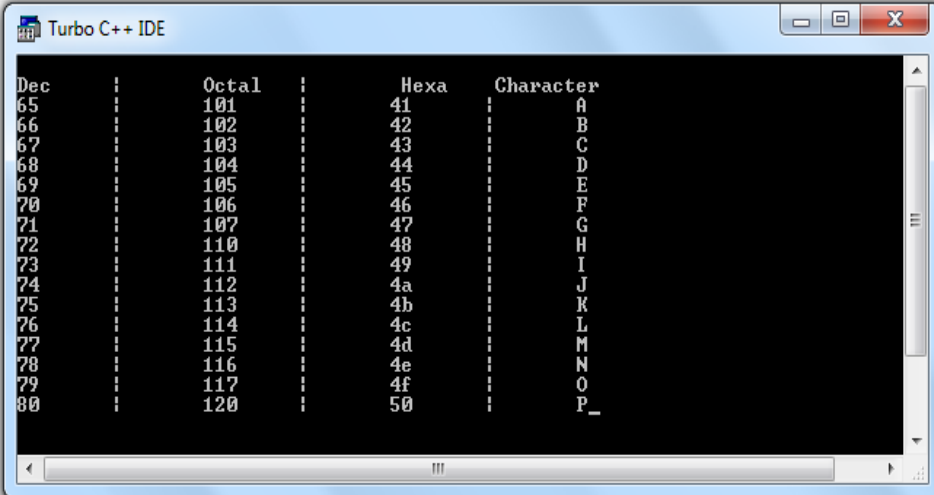


```
Turbo C++ IDE
Hello C+++...!
  55
  Information in X= 999
```

ตัวอย่างโปรแกรม 4.4 เป็นโปรแกรมที่นำรหัสการควบคุมข้อมูล (%d, %o, %x, %c) มาทำการสร้างข้อมูล ใช้ในการแสดงค่าของ ตัวเลขจำนวนเต็ม ตัวเลขฐานแปด ตัวเลขฐานสิบหก ตัวอักษร (ASCII Code) ของแต่ละค่าตามลำดับ และรหัสการควบคุมการแสดงผล (\n, \t) มาทำการสร้างระบบการแสดงผลที่สมบูรณ์

```
#include<stdio.h>
#include<conio.h>
#define number 80
void main()
{
    clrscr();
    unsigned char Tascii;
    printf("\nDec \t| \tOctal \t| \tHexa \t Character");
    for(Tascii = 65; Tascii<=number;++Tascii)
    printf(" \n%d\t|\t%o\t|\t%x\t|\t%c",Tascii,Tascii,Tascii,Tascii);
    getch();
}
```

ผลการทำงานของโปรแกรม



```

Turbo C++ IDE
Dec      |      |      |      |      |      |
65      |      |      |      |      |      |
66      |      |      |      |      |      |
67      |      |      |      |      |      |
68      |      |      |      |      |      |
69      |      |      |      |      |      |
70      |      |      |      |      |      |
71      |      |      |      |      |      |
72      |      |      |      |      |      |
73      |      |      |      |      |      |
74      |      |      |      |      |      |
75      |      |      |      |      |      |
76      |      |      |      |      |      |
77      |      |      |      |      |      |
78      |      |      |      |      |      |
79      |      |      |      |      |      |
80      |      |      |      |      |      |
Octal    |      |      |      |      |      |
101     |      |      |      |      |      |
102     |      |      |      |      |      |
103     |      |      |      |      |      |
104     |      |      |      |      |      |
105     |      |      |      |      |      |
106     |      |      |      |      |      |
107     |      |      |      |      |      |
110     |      |      |      |      |      |
111     |      |      |      |      |      |
112     |      |      |      |      |      |
113     |      |      |      |      |      |
114     |      |      |      |      |      |
115     |      |      |      |      |      |
116     |      |      |      |      |      |
117     |      |      |      |      |      |
120     |      |      |      |      |      |
Hexa     |      |      |      |      |      |
41      |      |      |      |      |      |
42      |      |      |      |      |      |
43      |      |      |      |      |      |
44      |      |      |      |      |      |
45      |      |      |      |      |      |
46      |      |      |      |      |      |
47      |      |      |      |      |      |
48      |      |      |      |      |      |
49      |      |      |      |      |      |
4a      |      |      |      |      |      |
4b      |      |      |      |      |      |
4c      |      |      |      |      |      |
4d      |      |      |      |      |      |
4e      |      |      |      |      |      |
4f      |      |      |      |      |      |
50      |      |      |      |      |      |
Character|      |      |      |      |      |
A       |      |      |      |      |      |
B       |      |      |      |      |      |
C       |      |      |      |      |      |
D       |      |      |      |      |      |
E       |      |      |      |      |      |
F       |      |      |      |      |      |
G       |      |      |      |      |      |
H       |      |      |      |      |      |
I       |      |      |      |      |      |
J       |      |      |      |      |      |
K       |      |      |      |      |      |
L       |      |      |      |      |      |
M       |      |      |      |      |      |
N       |      |      |      |      |      |
O       |      |      |      |      |      |
P       |      |      |      |      |      |
_       |      |      |      |      |      |

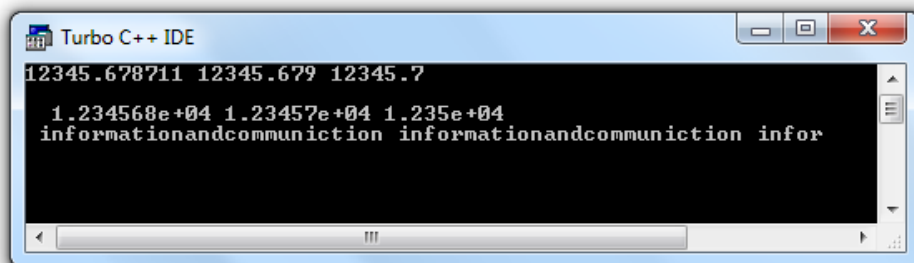
```

การแสดงผลข้อมูลโดยใช้ฟังก์ชัน `printf()` ในรูปแบบที่สามารถกำหนดขนาดข้อมูลที่แสดงได้ทั้งจำนวนตัวเลขก่อนจุดทศนิยมและหลังจุดทศนิยม เป็นการแสดงความเที่ยง (precision) โดยการกำหนดจำนวนหลักการแสดงผลตัวเลขหลังจุดทศนิยม ถ้ากำหนดจำนวนที่น้อยกว่า การแสดงผลจะทำการปัดเศษทศนิยม ดังตัวอย่างที่ 4.5

ตัวอย่างโปรแกรม 4.5 การกำหนดจำนวนตัวเลขก่อนและหลังจุดทศนิยม

```
#include<stdio.h>
#include<conio.h>
void main()
{
    float a=12345.6789;
    char text[30]={'i','n','f','o','r','m','a','t','i','o','n','a',
'n','d','c','o','m','m','u','n','i','c','a','t','i','o','n'};
    clrscr();
    printf("%5f%7.3f%5.1f\n\n ",a,a,a);
    printf("%e %.5e %.3e\n ",a,a,a);
    printf("%7s %9s %.5s",text,text,text);
    getch();
}
```

ผลการทำงานของโปรแกรม



```
Turbo C++ IDE
12345.678711 12345.679 12345.7
1.234568e+04 1.23457e+04 1.235e+04
informationandcommunication informationandcommunication infor
```

จากผลการทำงานโปรแกรม 4.5 การแสดงผลจากฟังก์ชัน `printf("%5f%7.3f%5.1f\n",a,a,a)`; ในรหัสควบคุม `%5f...a` เป็นการกำหนดให้แสดงค่าของตัวแปร `a` ตัวเลขจำนวน 5 ตำแหน่งก่อนจุดทศนิยม รหัสควบคุม `%7.3f...a` เป็นการกำหนดให้แสดงค่าของตัวแปร `a` ตัวเลขจำนวน 7 ตำแหน่งก่อนจุดทศนิยม หลังจุดทศนิยม 3 ตำแหน่ง และรหัสควบคุม `%5.1f...a` เป็นการกำหนดให้แสดงค่าของตัวแปร `a` ตัวเลขจำนวน 5 ตำแหน่งก่อนจุดทศนิยม หลังจุดทศนิยม 1 ตำแหน่ง

ส่วนฟังก์ชัน `printf("%e%.5e%.3e\n",a,a,a)`; ในรหัสควบคุม `%e...a` เป็นการกำหนดให้แสดงค่าของตัวแปร `a` แสดงข้อมูลในรูปแบบทศนิยม ที่มีเลขชี้กำลัง (exponential) ตามจำนวนจริง รหัสควบคุม `%.5e...a` เป็นการกำหนดให้แสดงค่าของตัวแปร `a` แสดงข้อมูลในรูปแบบทศนิยม ที่มีเลขชี้กำลัง โดยให้แสดงเลขหลังจุดทศนิยม 5 ตำแหน่ง และรหัสควบคุม `%.3e...a` เป็นการกำหนดให้แสดงค่าของตัวแปร `a` แสดงข้อมูลในรูปแบบทศนิยม ที่มีเลขชี้กำลัง โดยให้แสดงเลขหลังจุดทศนิยม 3 ตำแหน่ง

ส่วนฟังก์ชัน `printf("%7s%9s%.5s",text,text,text)`; ในรหัสควบคุม `%7s...text` เป็นการกำหนดให้แสดงค่าของตัวแปร `text` แสดงข้อมูลในรูปแบบตัวอักษรจางจำนวนพื้นที่แสดง 7 ตัวอักษร รหัสควบคุม `%9s...text` เป็นการกำหนดให้แสดงค่าของตัวแปร `text` แสดงข้อมูลในรูปแบบตัวอักษรจางจำนวนพื้นที่แสดง 9 ตัวอักษร และรหัสควบคุม `%.5s...text` เป็นการกำหนดให้แสดงค่าของตัวแปร `text` แสดงข้อมูลในรูปแบบตัวอักษรจำนวนแสดง 5 ตัวอักษร

ฟังก์ชัน scanf();

เป็นฟังก์ชันที่ใช้ในติดต่อกับผู้ใช้งานโปรแกรมในการรับข้อมูลจากแป้นพิมพ์ (Key-Board) เข้ามาเก็บไว้ในที่อยู่ของตัวแปร เพื่อนำไปใช้ในการคำนวณและการประมวลผล ถูกนิยามไว้ใน stdio.h

รูปแบบ	scanf(“รหัสของชนิดข้อมูล”, ที่อยู่ของตัวแปร);
ตัวอย่าง	scanf(“%d”, &A);
หมายถึง	รอรับค่าข้อมูลที่เป็นเลขจำนวนเต็มมาเก็บไว้ในที่อยู่ของตัวแปร A &A หมายถึง ที่อยู่ของตัวแปร A (Address A)
ตัวอย่าง	scanf(“%f”, &score);
หมายถึง	รอรับค่าข้อมูลที่เป็นเลขทศนิยมมาเก็บไว้ในที่อยู่ของตัวแปร score

ฟังก์ชัน scanf() มีการควบคุมข้อมูลจากการป้อนคีย์บอร์ด โดยใช้รหัสควบคุม ตามตารางที่ 4.3 ซึ่งมีข้อสังเกตว่า มีส่วนที่เหมือนและแตกต่างกับรหัสควบคุมของคำสั่ง printf() บางรหัส

ตารางที่ 4.3 แสดงรหัสควบคุมการรับข้อมูลจากคีย์บอร์ดของคำสั่ง scanf()

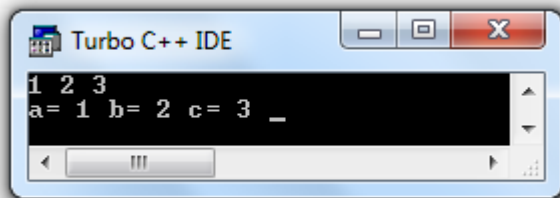
รหัสควบคุม	ความหมาย
%c	รับข้อมูลเป็นตัวอักขระ 1 ตัว
%d	รับข้อมูลเป็นตัวเลขจำนวน
%e	รับข้อมูลเป็นตัวเลขทศนิยม
%f	รับข้อมูลเป็นตัวเลขทศนิยม
%g	รับข้อมูลเป็นตัวเลขทศนิยม
%h	รับข้อมูลเป็นตัวเลขจำนวนเต็มแบบ short
%i	รับข้อมูลเป็นตัวเลขจำนวนเต็ม เลขฐานสิบหก เลขฐานแปด
%o	รับข้อมูลเป็นตัวเลขฐานแปด
%x	รับข้อมูลเป็นตัวเลขฐานหก
%s	รับข้อมูลเป็นข้อความ จำนวนข้อมูลที่เป็นตัวอักขระมากกว่าหนึ่งตัวใช้ในการเก็บอักขระนัล \0 ให้กับสตริงอัตโนมัติ
%u	รับข้อมูลเป็นตัวเลขจำนวนเต็มแบบ unsigned

การป้อนข้อมูลให้กับที่อยู่ตัวแปรของฟังก์ชัน scanf() โดยมาตรฐานใช้อักขระไวท์สเปซในการแยกข้อมูลเข้าของแต่ละฟิลด์ดังกล่าวอย่างโปรแกรม 4.6

ตัวอย่าง โปรแกรม 4.6 การป้อนข้อมูลโดยฟังก์ชัน scanf()

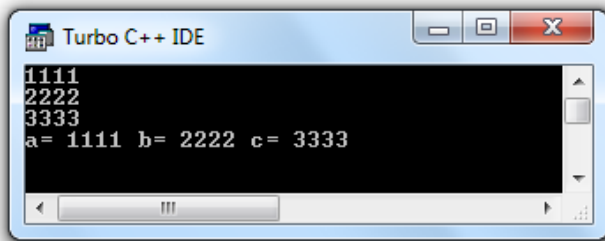
```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b,c;
    scanf("%d %d %d",&a,&b,&c);
    printf("a= %d b= %d c= %d ",a,b,c);
    getch();
}
```

ผลการทำงาน โปรแกรม



จากผลการทำงานโปรแกรม 4.6 ใช้การเว้นวรรคหรืออักขระไวท์สเปซ คั่นระหว่างการป้อนเลข 1 เลข 2 และเลข 3 เพื่อกำหนดค่าข้อมูลจากคีย์บอร์ดให้ที่อยู่ตัวแปรแต่ละตัว a b และ c ตามลำดับ

การป้อนข้อมูลสามารถใช้การการป้อนป้อน enter แทนการใช้การเว้นวรรคหรืออักขระไวท์สเปซ คั่นระหว่างการป้อนเลข 1 เลข 2 และเลข 3 ให้กับที่อยู่ตัวแปรแต่ละตัว a b และ c ตามลำดับ

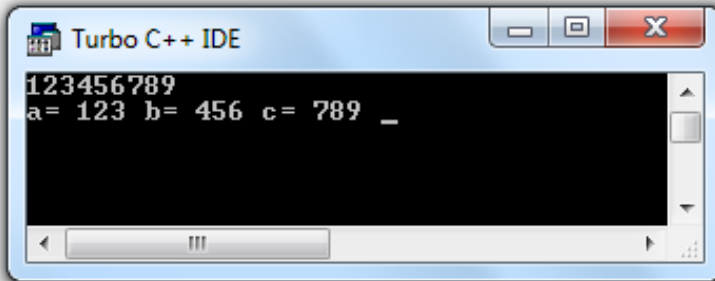


หรือการกำหนดจำนวนของตัวอักขระที่ใช้ในการป้อนข้อมูล จากตัวโปรแกรมของใช้ฟังก์ชัน scanf() ได้ โปรแกรม 4.7 เช่น การกำหนดจำนวนตัวเลขที่ต้องการในป้อน 3 ตัวอักขระ สามารถกำหนดด้วย %3d

ตัวอย่าง โปรแกรม 4.7 แสดงการกำหนดจำนวนอักขระในการป้อนฟังก์ชัน scanf()

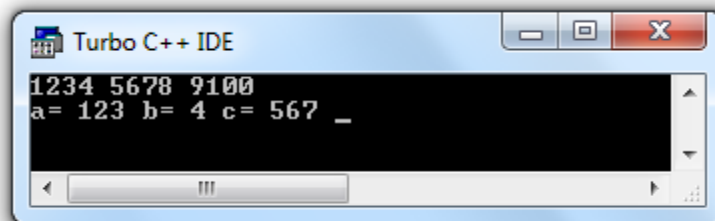
```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int a,b,c;
    scanf("%3d %3d %3d",&a,&b,&c);
    printf("a= %d b= %d c= %d ",a,b,c);
    getch();
}
```

ผลการทำงานโปรแกรม



```
Turbo C++ IDE
123456789
a= 123 b= 456 c= 789 _
```

แต่ถ้าผู้ใช้งานโปรแกรมป้อนผิดพลาด คือใช้การกำหนดจำนวนอักขระในการป้อน เช่น 1234 5678 9100 ส่งผลทำให้การรับข้อมูลผิดพลาดได้ดังนี้



```
Turbo C++ IDE
1234 5678 9100
a= 123 b= 4 c= 567 _
```

จากผลการทำงานโปรแกรมพบว่า ข้อมูลที่ตัวแปร a เก็บเป็นค่า 123 b เก็บค่า 4 และ c เก็บค่า 567 ส่วนค่า 89100 ถูกตัดทิ้งเพราะ โปรแกรมมีการกำหนดรับค่า 3 อักขระแรก เป็นของตัวแปร a ส่วนที่เหลือเป็นของตัวแปร b คือ 4 เท่านั้นเพราะหลังการป้อนเลข 4 มีการเว้นวรรค ส่งผลให้โปรแกรมตีความว่าจบการป้อนค่าชุดที่สอง ส่วนตัวแปร c รับค่าอีก 3 อักขระที่ป้อนคือ 567 เท่านั้นส่วนเหลือตัวโปรแกรมไม่สามารถรับได้ จึงไม่สนใจ

การกำหนดของรหัสควบคุมในโปรแกรมภาษาซียอมให้มีการกำหนดเพิ่มอาร์กิวเมนต์เพิ่มเติมได้ เช่น ต้องการกำหนดให้โปรแกรมรับจำนวนเต็มแบบ short หรือ unsigned สามารถกำหนดด้วย %hd ถ้ากำหนดให้รับจำนวนเต็มแบบ long หรือ unsigned long สามารถกำหนดด้วย %ld หรือถ้าต้องการกำหนดให้รับเลขทศนิยมมีความละเอียดสองเท่า สามารถกำหนดด้วย %Lf ดังตัวอย่างโปรแกรมที่ 4.8

ตัวอย่างโปรแกรม 4.8 การกำหนดการรับค่าเพิ่มอาร์กิวเมนต์เพิ่มเติม โดยทำการกำหนดตัวแปรที่ชนิด double 2 ตัวแปรคือ ad และ bd ในการรับค่าด้วย ฟังก์ชัน scanf("%lf %lf",&ad,&bd); และการกำหนดตัวแปรที่ชนิด short 2 ตัวแปรคือ as และ bs ในการรับค่าด้วย ฟังก์ชัน scanf("%hd %hd",&as,&bs); จะพบว่าถ้าทำการป้อนค่าให้แปรชนิดที่ short จำนวน 2 ไบท์ (บิท) ซึ่งมีค่าตัวเลขที่สามารถรับค่าได้อยู่ระหว่าง -32768 ถึง +32767 เท่านั้น ถ้ามีการป้อนค่าเกินไปหนึ่งค่า ข้อมูลจะทำการกำหนดค่าให้เป็นค่าติดลบค่าแรกของค่าคือ -32768 ดังผลการทำงานของโปรแกรม

```
#include<stdio.h>
#include<conio.h>
void main()
{

clrscr();

double ad,bd;

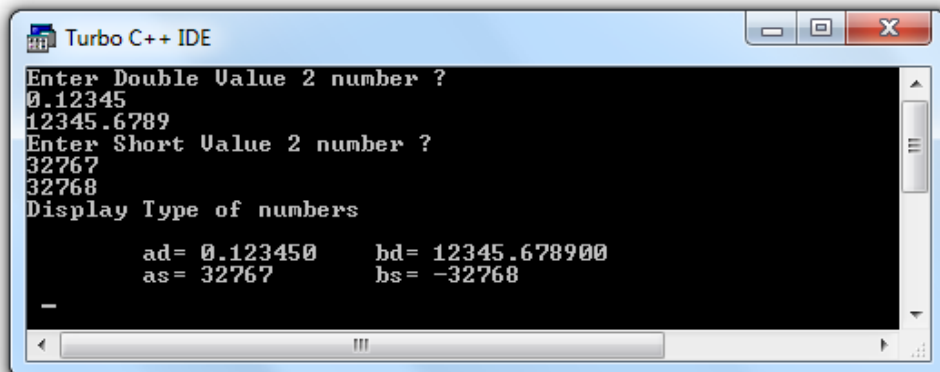
short as,bs;

printf("Enter Double Value 2 number ?\n");
```

ต่อจากด้านบน

```
scanf("%lf %lf",&ad,&bd);
printf("Enter Short Value 2 number ?\n");
scanf("%hd %hd",&as,&bs);
printf("Display Type of numbers\n\n");
printf("\tad= %lf \tbd= %lf\n ",ad,bd);
printf("\tas= %hd \tbs= %hd\n ",as,bs);
getch();
}
```

ผลการทำงานโปรแกรม



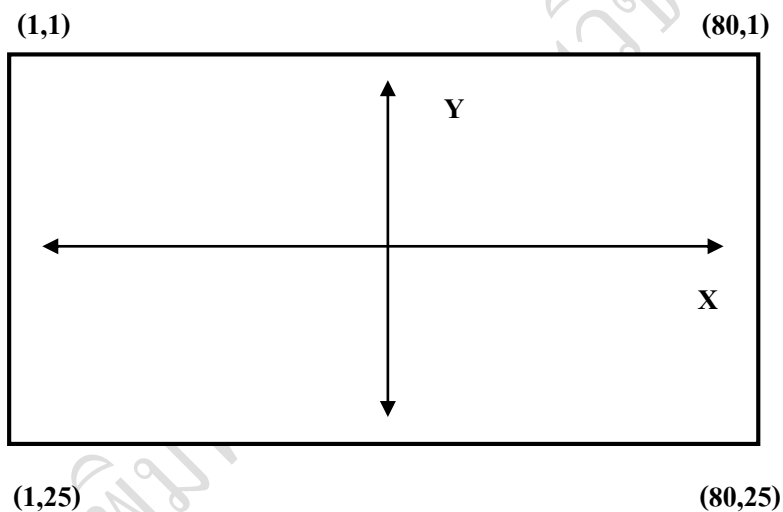
The screenshot shows the Turbo C++ IDE window with the following output:

```
Turbo C++ IDE
Enter Double Value 2 number ?
0.12345
12345.6789
Enter Short Value 2 number ?
32767
32768
Display Type of numbers

      ad= 0.123450      bd= 12345.678900
      as= 32767        bs= -32768
```

ตำแหน่งของเคอร์เซอร์ (Cursor Position)

ในการควบคุมการแสดงผลข้อมูลบนหน้าจอภาพของเครื่องคอมพิวเตอร์ในโหมดตัวอักษร (Text Mode) ตามมาตรฐานหน้าจอภาพประกอบด้วย 25 บรรทัด (Rows) และแต่ละบรรทัดจะมี 80 คอลัมน์ (Columns) เมื่อ X แทนค่าคอลัมน์ และ Y แทนค่าบรรทัด ดังนั้น X จะมีค่าตั้งแต่ 1 ถึง 80 ส่วน Y จะมีค่าตั้งแต่ 1 ถึง 25 ภาพที่ 4.1



รูปที่ 4.1 แสดงการแบ่งจำนวนตัวอักษรหน้าจอภาพในโหมดตัวอักษร

ฟังก์ชัน gotoxy();

เป็นฟังก์ชันที่ใช้กำหนดตำแหน่งในการแสดงผลได้ตามของแกน x ด้านคอลัมน์ และแกน y ด้านแถวหรือบรรทัด โดยเมื่อกำหนดตำแหน่งที่ต้องการแล้ว ต้องใช้ฟังก์ชัน printf(); หรือ cprintf(); สั่งให้แสดงผล ซึ่งฟังก์ชัน gotoxy() ถูกนิยามไว้ใน conio.h

รูปแบบ	gotoxy(ตำแหน่งแกน x, ตำแหน่งแกน y);
ตัวอย่างเช่น	gotoxy(10,5); printf("ICE iTech TRU");
หมายถึง	สั่งให้แสดงข้อความ ICE iTech TRU ที่ตำแหน่งแกน x หรือคอลัมน์นับจากขอบจอภาพด้านซ้ายมือมา 10 ตำแหน่ง และที่ตำแหน่งแกน y หรือ บรรทัดที่ 5 นับจากขอบด้านบนของจอภาพลงมา

ฟังก์ชัน clrscr();

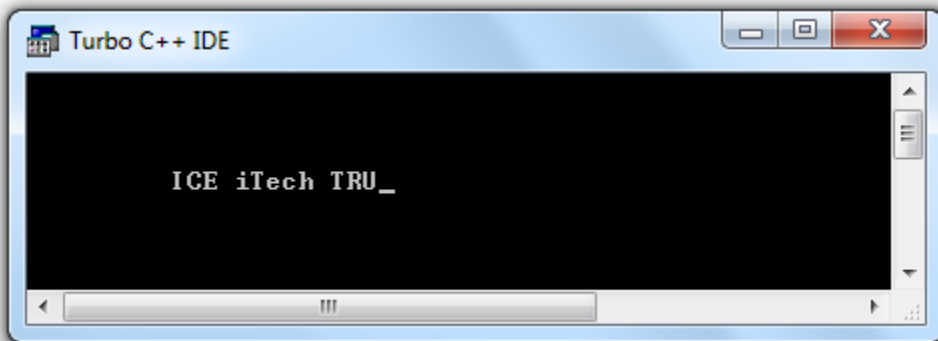
ในการเขียนโปรแกรมมีความจำเป็นต้องการลบผลการทำงาน โปรแกรมที่ทำงานเดิม ฟังก์ชัน clrscr() ทำหน้าที่การลบข้อความบนจอภาพ ถูกนิยามไว้ใน conio.h

รูปแบบ	clrscr();
---------------	-----------

ตัวอย่างโปรแกรมที่ 4.9

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    gotoxy(10,5);
    printf("ICE iTech TRU");
    getch();
}
```

ผลการทำงานโปรแกรม



ฟังก์ชัน `textcolor()`;

ในโหมดตัวอักษรสามารถกำหนดการแสดงผลบนหน้าจอภาพได้ด้วย เป็นฟังก์ชัน `textcolor()` โดยการทำงานต้องใช้ร่วมกับฟังก์ชัน `cprintf()`; ถูกนิยามไว้ใน `conio.h`

รูปแบบ	<code>textcolor (ค่าสี);</code>
หมายถึง	การใช้งานต้องกำหนดค่าสีของฟังก์ชัน <code>textcolor()</code> ก่อนทำการประมวลผลโดย <code>ค่าสี</code> จะสามารถกำหนดได้ 16 สี สามารถกำหนดเป็นตัวเลข หรือเป็น ชื่อสี ภาษาอังกฤษ ได้ ตารางที่ 4.4

ตารางที่ 4.4 แสดงค่าสีฟังก์ชัน `textcolor()`

ค่าสี	สีที่แสดง
0 (BLACK)	ดำ
1 (BLUE)	น้ำเงิน
2 (GREEN)	เขียว
3 (CYAN)	ฟ้า
4 (RED)	แดง
5 (MAGENTA)	ม่วง
6 (BROWN)	น้ำตาล
7 (LIGHTGRAY)	เทาสว่าง
8 (DARKGRAY)	เทาดำ

ค่าสี	สีที่แสดง
9 (LIGHTBLUE)	น้ำเงินสว่าง
10 (LIGHTGREEN)	เขียวสว่าง
11 (LIGHTCYAN)	ฟ้าสว่าง
12 (LIGHTRED)	แดงสว่าง
13 (LIGHTMAGENTA)	ม่วงสว่าง
14 (YELLOW)	เหลือง
15 (WHITE)	ขาว

ตัวอย่าง เช่น

```
textcolor(1);
cprintf("ICE iTech TRU");
```

หมายถึง

จะแสดงข้อความ ICE iTech TRU เป็นสีน้ำเงิน

ฟังก์ชัน `cprintf()`;

ฟังก์ชัน `cprintf()` เป็นฟังก์ชันแสดงข้อความหรือตัวอักษรเป็นสีออกทางจอภาพตามค่าสีที่กำหนดในฟังก์ชัน `textcolor()` รูปแบบการทำงานเหมือนกับฟังก์ชัน `printf()` ซึ่งฟังก์ชันทั้ง 2 ฟังก์ชันที่ถูกลิขิตไว้ใน `conio.h`

ฟังก์ชัน `textbackground()`;

ฟังก์ชัน `textbackground()` เป็นฟังก์ชันที่ใช้กำหนดสีพื้นของการแสดงข้อความตัวอักษร ซึ่งจะต้องใช้ร่วมกับฟังก์ชัน `textcolor()`; และ `cprintf()`; ถูกนิยามไว้ใน `conio.h` และ `graphics.h`

รูปแบบ	<code>textbackground(ค่าสี)</code>
ตัวอย่างเช่น	<code>textbackground(RED);</code> <code>textcolor(BLUE);</code> <code>cprintf("Test Color for Monitor");</code>
หมายถึง	แสดงข้อความ Test Color for Monitor เป็นสีน้ำเงิน และพื้นเป็นสีแดง

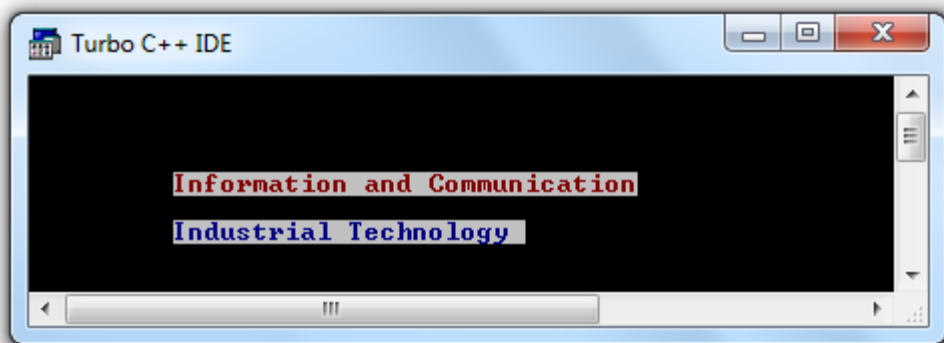
ตัวอย่าง โปรแกรมที่ 4.10

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main( )
{
    clrscr();
    gotoxy(10,5);
```

ต่อจากด้านบน

```
textbackground(WHITE);  
textcolor(RED);  
cprintf("Information and Communication");  
gotoxy(10,7);  
textbackground(WHITE);  
textcolor(BLUE);  
cprintf("Industrial Technology ");  
getch();  
}
```

ผลการทำงานโปรแกรม



สรุป

ในการเขียนโปรแกรมให้สามารถตอบโจทย์ ผู้ใช้งานโปรแกรมได้อย่างมีประสิทธิภาพ ปัจจัยสำคัญหนึ่งของสร้างโปรแกรมให้ตอบเงื่อนไขได้ ส่วนของโค้ดที่ทำหน้าในการรับข้อมูลเข้า ซึ่งมีฟังก์ชันที่สามารถนำมาใช้งานได้ 2 ประเภท ฟังก์ชันรับข้อมูลเข้าโดยไม่แสดงข้อมูลที่มีการป้อนหน้าจอภาพ และฟังก์ชันรับข้อมูลเข้าโดยแสดงข้อมูลที่มีการป้อนหน้าจอภาพ ในการควบคุมตำแหน่งการแสดงผลหน้าจอสามารถสั่งได้ด้วยรหัสควบคุมข้อมูลด้วยเครื่องหมาย % ตามด้วยรหัสควบคุม และเครื่องหมาย \ ตามด้วยรหัสควบคุม เป็นต้น

ในการควบคุมการแสดงผลข้อมูล เป็นตัวอักษรสี สามารถใช้ฟังก์ชัน `cprintf()` แทนฟังก์ชัน `printf()` ซึ่งข้อมูลเป็นการแสดงตัวอักษรปกติ (สีดำ) การควบคุมตำแหน่งทุกตำแหน่งระหว่าง ตัวอักษรที่สามารถแรก จนถึงตัวอักษรสุดท้าย(80) และสามารถกำหนดตำแหน่งบรรทัดแรก จนถึงบรรทัดสุดท้าย (25) ได้ด้วยฟังก์ชัน `gogxy()` เป็นต้น

แบบฝึกหัดท้ายบทที่ 4

ตอนที่ 1 จงเติมคำหรือข้อความในช่องว่างต่อไปนี้ให้ถูกต้อง

1. ฟังก์ชัน `getch()`, `getche()` มีความแตกต่างหรือสัมพันธ์กันอย่างไร
.....
2. ฟังก์ชัน `getchar()`, `putchar()` มีความแตกต่างหรือสัมพันธ์กันอย่างไร
.....
3. ฟังก์ชัน `printf()` ต้องนิยามร่วมกับเฮดเดอร์ใด
.....
4. ฟังก์ชัน `printf()` สามารถสั่งให้แสดงผลข้อความพร้อมกับตัวแปรได้หรือไม่
.....
5. ฟังก์ชัน `printf()` ถ้าต้องการให้แสดงค่าที่เป็นทศนิยมต้องใช้รหัสควบคุมใด
.....
6. ฟังก์ชัน `scanf()` มีรูปแบบการสั่งการอย่างไร
.....
7. ฟังก์ชัน `scanf()` สามารถสั่งให้แสดงผลข้อความพร้อมกับรับค่าเก็บในตัวแปรได้หรือไม่
.....
8. จงเขียนฟังก์ชัน `scanf()` ให้รับค่าที่เป็นตัวอักษร 1 ตัวเก็บในตัวแปรชื่อ `input`
.....
9. ถ้าต้องให้แสดงผลเลขฐาน 16 ผู้เขียนโค้ดควรต้องใช้ฟังก์ชันใด
.....
10. ถ้าต้องการรับข้อมูลครั้งละ 3 ตัวแปรสามารถทำได้หรือไม่
.....

ตอนที่ 2 จงทำเครื่องหมายกากบาท (x) ทับหน้าข้อที่ถูกต้องที่สุด

1. ในการแสดงผลหน้าจอภาพในโหมด Text อยากทราบว่าสามารถแสดงข้อมูลได้สูงสุดกี่บรรทัด

ก. 15	ข. 25
ค. 40	ง. 80
2. ในการแสดงผลหน้าจอภาพในโหมด Text อยากทราบว่าสามารถแสดงข้อมูลได้สูงสุดกี่ตัวอักษรในหนึ่งบรรทัด

ก. 15	ข. 25
ค. 40	ง. 80
3. คำสั่งใดทำหน้าที่ในการกำหนดตำแหน่งที่ต้องการแสดงข้อมูลบนหน้าจอภาพคอมพิวเตอร์

ก. textcolor()	ข. scanf()
ค. gotoxy()	ง. clrscr()
4. ถ้าต้องการลบข้อมูลบนหน้าจอภาพคอมพิวเตอร์ต้องใช้ข้อใด

ก. textcolor()	ข. scanf()
ค. gotoxy()	ง. clrscr()
5. ถ้าต้องการกำหนดสีให้กับข้อมูลที่เป็นตัวอักษรบนหน้าจอภาพคอมพิวเตอร์ต้องใช้ข้อใด

ก. textcolor()	ข. scanf()
ค. gotoxy()	ง. clrscr()

6. รูปแบบกำหนดสีให้กับข้อมูลที่เป็นตัวอักษรบนหน้าจอภาพเป็น
textcolor(CYAN) จะแสดงสีใด
- ก. ฟ้ำ
ข. ม่วง
ค. น้ำตาล
ง. เทาดำ
7. รูปแบบกำหนดสีให้กับข้อมูลที่เป็นตัวอักษรบนหน้าจอภาพเป็น
textcolor(DARKGRAY) จะแสดงสีใด
- ก. ฟ้ำ
ข. ม่วง
ค. น้ำตาล
ง. เทาดำ
8. cprintf() ต้องทำการนิยามกับเฮดเดอร์ใด
- ก. stdio.h
ข. conio.h
ค. graphic.h
ง. math.h
9. textbackground() เป็นฟังก์ชันทำหน้าที่ใด
- ก. กำหนดสีให้กับตัวอักษร
ข. กำหนดสีให้กับพื้นหลังของตัวอักษร
ค. กำหนดสีให้กับพื้นหลังของจอภาพ
ง. กำหนดสีให้กับ
10. textbackground(BROWN) เป็นฟังก์ชันทำหน้าที่ใด
- ก. ฟ้ำ
ข. ม่วง
ค. น้ำตาล
ง. เทาดำ
-

ตอนที่ 3 จงทำการวิเคราะห์คำถามและทำการเขียนอธิบายคำตอบตามที่ผู้อ่านเข้าใจโดยยึดความถูกต้องของเนื้อหาประกอบการบรรยาย

จากการโปรแกรมภาษาซี

```
#include<stdio.h>
```

```
int i, j, k;
```

1. ให้ผู้อ่านเขียนฟังก์ชันการรับค่าด้วย scanf() เก็บในตัวแปร i, j และ k ดังนี้
 - ก. รับค่าตัวแปร i, j และ k เป็นเลขจำนวนเต็มฐานสิบ
 - ข. รับค่าตัวแปร i เป็นเลขจำนวนเต็มฐานสิบ รับค่าตัวแปร j เป็นเลขจำนวนเต็มแปด และรับค่าตัวแปร k เป็นเลขจำนวนเต็มฐานสิบหก
2. จงบอกความหมายของสตริงควบคุมในฟังก์ชัน scanf() ดังนี้
 - ก. scanf("%10d %3hd %10f %10le", &w, &x, &y, &z)
 - ข. scanf("%12lx %5ho %15hu %15lu", &w, &x, &y, &z)
 - ค. scanf("%10D %hd %10f %10e", &w, &x, &y, &z)
 - ง. scanf("%5d %3d %2lf %10lf", &w, &x, &y, &z)

จากโปรแกรมภาษาซี

```
#include<stdio>
```

```
float m, n, o;
```

3. เขียนฟังก์ชัน printf() ให้แสดงค่าตัวแปร และนิพจน์โดยการแปลงค่าชนิดตัวแปร
 - ก. m n และ o โดยแต่ละตัวมีความกว้างของฟิลด์ที่แสดง 5 ตำแหน่ง มีทศนิยมไม่เกิน 3 ตำแหน่ง
 - ข. (m + n) และ (n - o) โดยแต่ละตัวมีความกว้างของฟิลด์ที่แสดง 7 ตำแหน่ง มีทศนิยมไม่เกิน 5 ตำแหน่ง
 - ค. sqrt(m + n) และ fabs(n - o) โดยแต่ละตัวมีความกว้างของฟิลด์ที่แสดง 8 ตำแหน่ง มีทศนิยมไม่เกิน 4 ตำแหน่ง

จากโปรแกรมภาษาซี

```
#include<stdio>
```

```
float m= 2.5, n=0.0005, o=3000;
```

4. จงเขียนผลลัพธ์ที่ได้จากฟังก์ชัน printf() ดังต่อไปนี้

ก. printf(“%f %f %f”,m,n,o)

ข. printf(“%2f %2f %2f”,m,n,o)

ค. printf(“%7f %7f %7f”,m,n,o)

ง. printf(“%8.2f %8.2f %8.2f”,m,n,o)

จากโปรแกรมภาษาซี

```
#include<stdio>
```

```
char tx[80];
```

5. โดย tx เก็บค่า information and communication engineering department ให้ผู้อ่านเขียนผลลัพธ์ของฟังก์ชัน printf() ดังต่อไปนี้

ก. printf(“%s”,tx)

ข. printf(“%10s”,tx)

ค. printf(“%.10s”,tx)

เอกสารอ้างอิง

ศรัณย์ อินทโกสุม (2539). ทฤษฎีและตัวอย่างโจทย์การเขียนโปรแกรมด้วยภาษาซี
กรุงเทพฯ : แมคกรอฮิล อินเทอร์เน็ต เนชั่นแนล เอ็นเตอร์ไพรส์, ینگค์.

ธันวาคม ศรีประโมง (2539). การเขียนโปรแกรมภาษาซีสำหรับวิศวกรรม. พิมพ์ครั้งที่
ที่ 4. กรุงเทพฯ : มหาวิทยาลัยเทคโนโลยีมหานคร.

วิจักขณ์ ศรีตังจะเลศวาจา และคุณผู้ ประเสริฐฐิติพงษ์ 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.