

วิชา การพัฒนาโปรแกรม (ภาษาซี)

รหัสวิชา ว31293

หน่วยการเรียนรู้ที่ 4 การรับและแสดงผลข้อมูล

สาระการเรียนรู้

1. ตัวดำเนินการ (Operator)
2. ตัวถูกดำเนินการ (Operand)
3. นิพจน์ (Expression)
4. เครื่องหมายและตัวดำเนินการทางคณิตศาสตร์
5. การเปลี่ยนชนิดข้อมูล

ผลการเรียนรู้

สามารถสร้างโปรแกรมที่มีขั้นตอนการรับและแสดงผลข้อมูลได้อย่างถูกต้อง

จุดประสงค์การเรียนรู้

1. สามารถเลือกใช้งานฟังก์ชันเพื่อการรับข้อมูลและการแสดงผลข้อมูลได้
2. สามารถเขียนโปรแกรมเพื่อการรับข้อมูล และแสดงผลข้อมูลได้



ใบความรู้ที่ 4 เรื่อง การรับและแสดงผลข้อมูล

จัดทำโดย นางพรพนารัตน์ ชมภูษ

ในภาษา C มีคำสั่งรับและแสดงผลอยู่หลายคำสั่งด้วยกัน ซึ่งสามารถเลือกใช้งานตามความเหมาะสม ในการเรียกใช้งานคำสั่งรับและแสดงผลข้อมูลในฟังก์ชันฟังก์ชันพื้นฐาน ต้องประกาศใช้งาน Header file ที่ชื่อว่า `stdio.h` และ `conio.h` ก่อน โดยในบทนี้จะขอกล่าวถึงคำสั่งรับและแสดงผลข้อมูลดังต่อไปนี้

- ฟังก์ชัน `printf()`
- ฟังก์ชัน `scanf()`
- ฟังก์ชัน `getchar()`
- ฟังก์ชัน `putchar()`
- ฟังก์ชัน `getch()`
- ฟังก์ชัน `getche()`



คำสั่งการแสดงผลข้อมูล

1. การแสดงผลออกทางจอภาพด้วยคำสั่ง `printf()`

ฟังก์ชัน `printf()` เป็นฟังก์ชันที่ใช้สำหรับแสดงผลข้อมูล ไม่ว่าจะเป็นข้อมูลที่เป็นตัวเลข จำนวนเต็ม ทศนิยม อักขระ ข้อความ ค่าตัวแปร หรือค่าของนิพจน์ และในการแสดงผลข้อมูลด้วย `printf()` จะมีรูปแบบวิธีการใช้ฟังก์ชัน `printf()` ในการควบคุมการแสดงผล โดยมีวิธีการทำงานใช้งานอยู่ 6 แบบ คือ

- 1) การแสดงผลด้วยข้อความธรรมดา
- 2) การแสดงผลด้วยรหัสควบคุมรูปแบบการแสดงผล
- 3) การแสดงผลด้วยการใช้อักขระควบคุมการแสดงผล
- 4) การควบคุมขนาดพื้นที่การแสดงผล
- 5) การควบคุมจำนวนอักขระในการแสดงผล
- 6) การควบคุมการแสดงผลตัวเลขหลังจุดทศนิยม

ข้อสังเกต ในส่วนที่ใช้ในการควบคุมรูปแบบการแสดงผล จะต้องเขียนไว้ภายในเครื่องหมาย “ ” (Double quote) เสมอ

1) การแสดงผลด้วยข้อความ

เป็นการเรียกใช้งานฟังก์ชัน printf() เพื่อแสดงข้อความออกทางหน้าจอ สามารถใช้งานคำสั่งในรูปแบบคำสั่งดังนี้

รูปแบบ

```
printf("ข้อความ");
```

ตัวอย่าง

```
printf("Chaibadanwittaya");
```

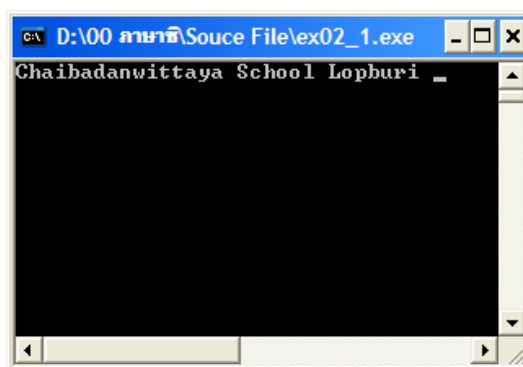


กิจกรรมเสริมทักษะ

ให้นักเรียนทดลองเขียนโปรแกรมดังต่อไปนี้ โดยบันทึกโปรแกรม ชื่อ ex05_1.cpp แล้วสังเกตผลลัพธ์ที่ได้

```

1 #include <stdio.h>
2 #include <conio.h>
3 main()
4 {
5     printf("Chaibadanwittaya ");
6     printf("School ");
7     printf("Lopburi ");
8     getche();
9 }
```



รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของโปรแกรมex05_1.exe

2) การแสดงผลด้วยรหัสควบคุมรูปแบบการแสดงผล

เมื่อมีการใช้งานตัวแปรในการเก็บข้อมูลในการเขียนโปรแกรม แล้วจำเป็นต้องแสดงผลค่าของตัวแปร ในภาษา C จะต้องใส่รหัสควบคุมรูปแบบ (Format Code) ให้กับตัวแปรก่อนอ้างถึงตัวแปรนั้น ๆ การแสดงผลด้วยรหัสควบคุมรูปแบบการแสดงผลมีรูปแบบการใช้งานดังนี้

รูปแบบ

```
printf("format code", variable1 , variable2 ,...);
```

โดยรหัสควบคุมรูปแบบการแสดงผลข้อมูล (Format Code) ชนิดต่าง ๆ ในภาษา C มีดังต่อไปนี้

ตาราง แสดงรหัสควบคุมรูปแบบการแสดงผลข้อมูล (Format Code)

รหัสควบคุม	การนำไปใช้งาน
%d	แสดงผลค่าของตัวแปรชนิดจำนวนเต็มฐานสิบ
%u	แสดงผลตัวเลขจำนวนเต็มบวก
%o	แสดงผลออกมาในรูปแบบของเลขฐาน 8
%x	แสดงผลออกมาในรูปแบบของเลขฐาน 16
%f	แสดงผลค่าของตัวแปรชนิดจำนวนทศนิยม (6 หลัก)
%e	แสดงผลตัวเลขทศนิยมออกมาในรูปแบบของ E หรือ e ยกกำลัง
%c	แสดงผลอักขระ 1 ตัว (Char)
%s	แสดงผลข้อความ (String)
%p	แสดงผลตัวชี้ตำแหน่ง (Pointer)

ตัวอย่าง

แบบที่ 1 แสดงผลแบบรวมบรรทัดเดียว

```
printf(“%s %d %c”, name , salary , sex);
```

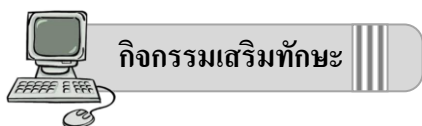
แบบที่ 2 แสดงผลแบบแยกบรรทัดแสดงพร้อมข้อความอธิบาย

```
printf(“My name is : %s ”, name);
```

```
printf(“Salary is : %d ”, salary);
```

```
printf(“Sex : %c ”, sex);
```

หมายความว่า รหัสควบคุมรูปแบบการแสดงผลข้อมูลในรูปแบบของ %s จะแสดงผลข้อมูลที่เป็นข้อความของตัวแปร name ส่วน %d จะแสดงผลค่าของตัวแปรชนิดจำนวนเต็มฐานสิบ และ %c จะแสดงผลอักขระ 1 ตัว ของตัวแปร sex



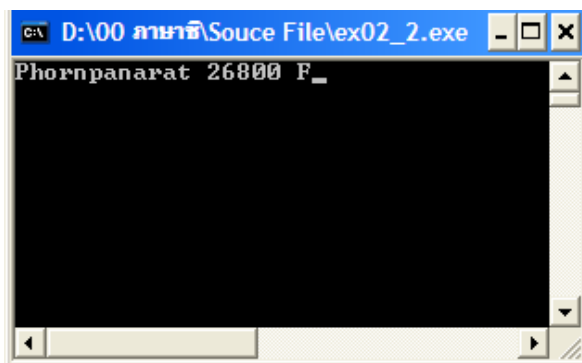
กิจกรรมเสริมทักษะ

ให้นักเรียนทดลองเขียนโปรแกรมดังต่อไปนี้ โดยบันทึกโปรแกรม ชื่อ ex05_2.cpp สังเกตผลลัพธ์ที่ได้

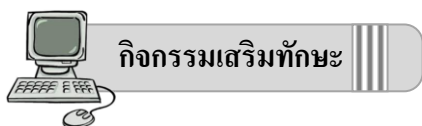
แบบที่ 1 แสดงผลแบบรวมบรรทัดเดียว

```

1 #include <stdio.h>
2 #include <conio.h>
3 char name[20]= "Phornpanarat";
4 char sex= 'F';
5 int salary=26800;
6 main()
7 {
8     printf("%s %d %c", name , salary , sex);
9     getch();
10 }
```



รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของโปรแกรม ex05_2.exe



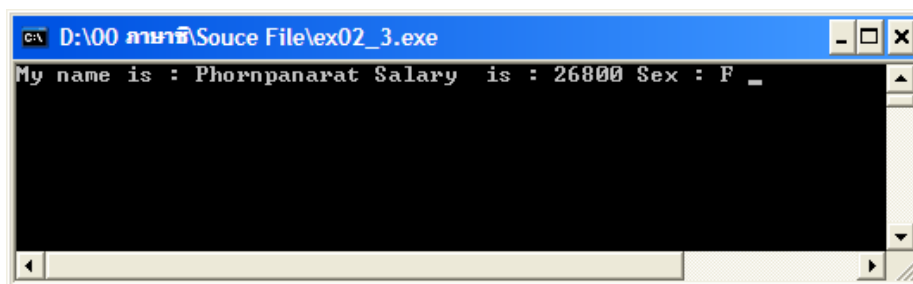
ให้นักเรียนทดลองเขียนโปรแกรมดังต่อไปนี้ โดยบันทึกโปรแกรม ชื่อ ex05_3.cpp สังเกตผลลัพธ์ที่ได้

แบบที่ 2 แสดงผลรวมกับข้อความอธิบาย

```

1 #include <stdio.h>
2 #include <conio.h>
3 char name[30]= "Phornpanarat",
4 char sex= 'F';
5 int salary=26800;
6 main()
7 {
8     printf("My name is %s ", name);
9     printf("Salary is %d ", salary);
10    printf("Sex %c ", sex);
11    getche( );
12 }

```



รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของโปรแกรม ex02_3.exe

3) การแสดงผลด้วยการใช้อักขระควบคุมการแสดงผล

การจัดรูปแบบการแสดงผลให้ดูเป็นระเบียบมากยิ่งขึ้น สามารถทำได้โดย การใช้อักขระควบคุมการแสดงผลร่วมกับคำสั่ง printf() ตัวอย่างเช่น การขึ้นบรรทัดใหม่ การเว้นระยะห่างระหว่างข้อความ เป็นต้น โดยมีรูปแบบการใช้งานดังนี้

รูปแบบ

```
printf("ข้อความ , อักขระควบคุม");
```

```
printf("ข้อความ , format code, อักขระควบคุม", variable );
```

ตัวอย่าง

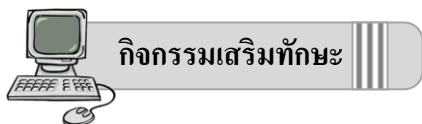
```
printf("Hollo..\n How are you?\n");
```

```
printf("My name is %s \n", name);
```

ในภาษา C มีอักขระควบคุมการแสดงผลหลายรูปแบบด้วยกัน ดังแสดงต่อไปนี้

ตาราง แสดงอักขระควบคุมการแสดงผล

อักขระควบคุม	การนำไปใช้งาน
\0 (ศูนย์)	ค่าว่าง (null)
\a	ส่งเสียงบี๊ป 1 ครั้ง
\b	ถอยหลังหนึ่งตัวอักษร
\f	ขึ้นหน้าใหม่ (form feed)
\n	ขึ้นบรรทัดใหม่ (new line)
\r	ย้ายเคอร์เซอร์กลับไปต้นบรรทัด
\t	กำหนดระยะห่างออกไป 1 แท็บแนวนอน (horizontal tab)
\v	กำหนดระยะห่างออกไป 1 แท็บแนวตั้ง (vertical tab)
\'	พิมพ์เครื่องหมาย '
\"	พิมพ์เครื่องหมาย "
\\	พิมพ์เครื่องหมาย \

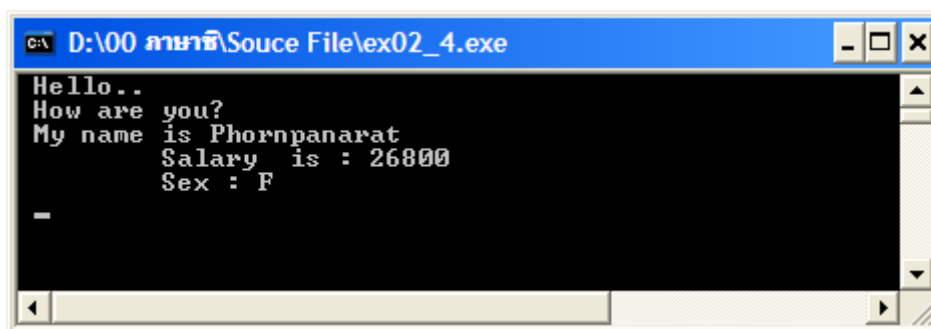


ให้นักเรียนทดลองเขียนโปรแกรมดังต่อไปนี้ โดยบันทึกโปรแกรม ชื่อ ex05_4.cpp สังเกตผลลัพธ์ที่ได้

```

1  #include <stdio.h>
2  #include <conio.h>
3  char name[30]= "Phornpanarat";
4  char sex= 'F';
5  int salary=26800;
6  main()
7  {
8      printf(" Hello.. \n How are you? \n");
9      printf(" My name is %s \n", name);
10     printf(" Salary is : %d \n \t ", salary);
11     printf(" Sex : %c \n \t \a ", sex);
12     getch();
13 }

```



รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของโปรแกรม ex05_4.exe

4) การควบคุมพื้นที่การแสดงผล

ตามปกติในการแสดงผลตัวแปรภาษา C จะเตรียมพื้นที่ให้พอดีกับข้อมูลที่จะแสดงผล
ตัวอย่าง กำหนดให้แสดงข้อความคำว่า HELLO และจำนวนตัวเลข 123.456 ดังรูป

H	E	L	L	O
---	---	---	---	---

1	2	3	.	4	5	6
---	---	---	---	---	---	---

แต่เราสามารถเพิ่มพื้นที่ในการแสดงผลได้ตามต้องการ โดยการจองพื้นที่หน่วยความจำให้มากขึ้นตามความต้องการ โดยจะเว้นพื้นที่ทางด้านซ้ายเอาไว้ และโปรแกรมจะแสดงผลชิดด้านขวาดังรูป

ตัวอย่าง กำหนดให้แสดงข้อความ HELLO ในพื้นที่ขนาด 8 ช่อง และจำนวนตัวเลข 123.456 ในพื้นที่ขนาด 12 ช่อง โปรแกรมจะแสดงผล ดังรูป

				H	E	L	L	O											
												1	2	3	.	4	5	6	

วิธีการกำหนดขนาดพื้นที่การแสดงผล

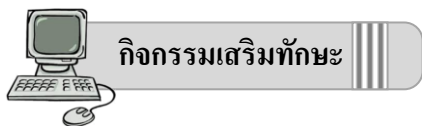
ในภาษา C การกำหนดขนาดพื้นที่การแสดงผลสามารถทำได้โดย ใส่ตัวเลขขนาดของพื้นที่ที่ต้องการไว้หลังเครื่องหมาย % ในรหัสควบคุมรูปแบบ (Format Code) โดยมีรูปแบบการใช้งานดังต่อไปนี้

รูปแบบ

```
printf("% ขนาดพื้นที่ รหัสควบคุมรูปแบบ \n", ข้อความ);
printf("% ขนาดพื้นที่ รหัสควบคุมรูปแบบ \n", ตัวเลข);
printf("% ขนาดพื้นที่ รหัสควบคุมรูปแบบ \n", ตัวแปร);
```

ตัวอย่าง

```
printf("%10d \n", 0123456789);
printf("%8s \n", "Hello");
printf("%f \n", 3.141);
printf("%4c \n", "Y");
```

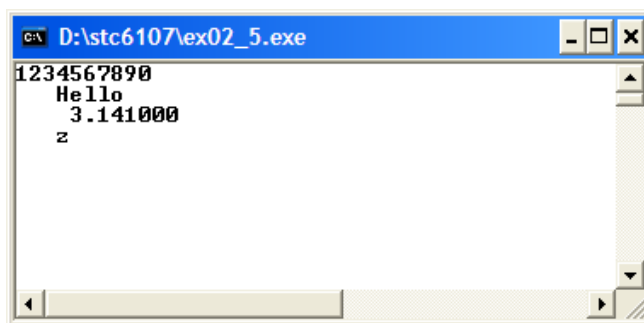


ให้นักเรียนทดลองเขียนโปรแกรมดังต่อไปนี้ โดยบันทึกโปรแกรม ชื่อ ex05_5.cpp สังเกตผลลัพธ์ที่ได้

```

1 #include <stdio.h>
2 #include <conio.h>
3 main( )
4 {
5     printf(“%10d \n”, 1234567890);
6     printf(“%8s \n”, “Hello”);
7     printf(“%12.f \n”, 3.141);
8     printf(“%4c \n”, ‘z’);
9     getch( );
    }

```



รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของโปรแกรม ex05_5.exe

5) การควบคุมจำนวนอักขระในข้อความที่จะให้แสดงผล

การแสดงผลข้อความโดยใช้รหัสควบคุมรูปแบบ %s เราสามารถกำหนดได้ว่า ต้องการจะแสดงข้อความทั้งหมด หรือเลือกเฉพาะบางส่วนของข้อความ โดยกำหนดจำนวนอักขระที่ต้องการแสดงผลได้ โดยมีรูปแบบการทำงานดังต่อไปนี้

รูปแบบ

%*n*s

%*m.n*s

n หมายถึง จำนวนอักขระที่ต้องการให้แสดง

m หมายถึง จำนวนพื้นที่ที่ต้องการใช้ในการแสดงผล

ถ้าหากต้องการกำหนดตำแหน่งข้อความให้ชิดซ้าย ก็สามารถทำได้โดยใส่เครื่องหมาย –ไว้ด้านหน้า ดังรูปแบบต่อไปนี้

รูปแบบ

`%-m.n`

ตัวอย่าง

```
printf("%.9d ", "Chaibadanwittaya");
printf("%12.9d ", "Chaibadanwittaya");
printf("%-12.9d ", "Chaibadanwittaya");
```



ให้นักเรียนทดลองเขียนโปรแกรมดังต่อไปนี้ โดยบันทึกโปรแกรม ชื่อ ex05_6.cpp
สังเกตผลลัพธ์ที่ได้

```
1 #include <stdio.h>
2 #include <conio.h>
3 main()
4 {
5     printf("%s\n", "12345678901234567890");
6     printf("%s\n", "Chaibadanwittaya");
7     printf("%.9s\n", "Chaibadanwittaya");
8     printf("%12.9s\n", "Chaibadanwittaya");
9     printf("%-12.9s\n", "Chaibadanwittaya");
    getch();
}
```

```

C:\> D:\stc6107\ex02_6.exe
12345678901234567890
Chaibadanwittaya
Chaibadan
Chaibadan
Chaibadan

```

รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของโปรแกรม ex05_6.exe

6) การควบคุมการแสดงผลตัวเลขหลังจุดทศนิยม

การแสดงผลตัวเลขทศนิยม ตามปกติถ้าไม่ได้กำหนดค่าใด ๆ เพิ่มเติม เมื่อใช้รหัสควบคุมรูปแบบ `%f` โปรแกรมจะแสดงผลตัวเลขหลังจุดทศนิยมทั้งหมด 6 ตำแหน่ง ดังนั้นหากต้องการควบคุมการแสดงผลตัวเลขหลังทศนิยมสามารถทำได้โดยกำหนดตามรูปแบบการใช้คำสั่งดังนี้

รูปแบบ

`%.nf`

ตัวอย่าง

```
printf("%.2f ", 3.14);
```

```
printf("%.2f ",3 );
```



กิจกรรมเสริมทักษะ

ให้นักเรียนทดลองเขียนโปรแกรมดังต่อไปนี้ โดยบันทึกโปรแกรม ชื่อ `ex05_7.cpp` สังเกตผลลัพธ์ที่ได้

```
1 #include <stdio.h>
2 #include <conio.h>
3 main( )
4 {
5     printf("%f\n", 3.14);
6     printf("%f\n", -3.14);
7     printf("%.2f\n", 3.14);
8     printf("%.3f\n", -3.14);
9     getche( );
}
```

```
D:\stc6107\ex02_7.exe
3.140000
-3.140000
3.14
-3.140
```

รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของโปรแกรม `ex05_7.exe`

2. การแสดงผลออกทางจอภาพด้วยคำสั่ง putchar()

ในการแสดงผลตัวอักษรหรืออักขระ ออกทางจอภาพ นอกจากการใช้คำสั่ง printf() พร้อมกับกำหนดรหัสควบคุมรูปแบบเป็น %c แล้ว เราสามารถเรียกคำสั่งสำหรับการแสดงผลตัวอักษรหรืออักขระ โดยเฉพาะได้อีกด้วย โดยใช้คำสั่ง putchar() ซึ่งมีรูปแบบการเรียกใช้คำสั่งดังนี้

รูปแบบ

```
putchar (variable ชนิดอักขระ);
```

ตัวอย่าง

```
putchar (ans);
```

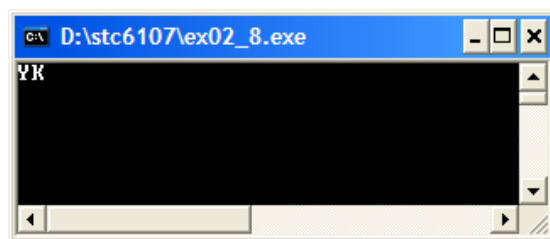


กิจกรรมเสริมทักษะ

ให้นักเรียนทดลองเขียนโปรแกรมดังต่อไปนี้ โดยบันทึกโปรแกรม ชื่อ ex05_8.cpp สังกะผลลัพ์ที่ได้

```

1 #include <stdio.h>
2 #include <conio.h>
3 char ans1='Y';
4 char ans2='OK';
5 main( )
6 {
7     putchar(ans1);
8     putchar(ans2);
9     getch( );
10 }
```



รูปภาพ แสดงผลลัพ์ที่ได้จากการทำงานของโปรแกรมex05_8.exe

3. การแสดงผลออกทางจอภาพด้วยคำสั่ง puts()

การแสดงผลในรูปแบบข้อความ นอกจากใช้คำสั่ง printf() และยังสามารถใช้คำสั่ง puts () ในการแสดงผลในรูปแบบข้อความ โดยมีรูปแบบการเรียกใช้คำสั่งดังนี้

รูปแบบ

puts(ตัวแปรชนิดตัวอักษรแบบชุด)

หรือ

puts(“ข้อความ”)

ตัวอย่าง

puts(name)

puts(“Phornpanarat”)

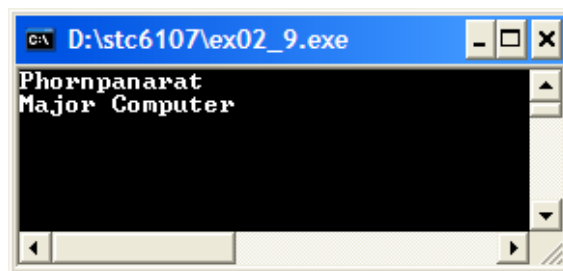


กิจกรรมเสริมทักษะ

ให้นักเรียนทดลองเขียนโปรแกรมดังต่อไปนี้ โดยบันทึกโปรแกรม ชื่อ ex05_9.cpp สังเกตผลลัพธ์ที่ได้

```

1 #include <stdio.h>
2 #include <conio.h>
3 char name[20]= “Phornpanarat”;
4 main()
5 {
6     puts(name);
7     puts(“Major Computer”);
8     getche();
9 }
```



รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของโปรแกรม ex05_9.exe



คำสั่งการรับข้อมูล

1. การรับข้อมูลจากคีย์บอร์ดด้วยคำสั่ง scanf()

ในภาษา C การรับข้อมูลจากคีย์บอร์ดสามารถทำได้โดยการเรียกใช้ฟังก์ชัน `scanf()` ซึ่งเป็นฟังก์ชันมาตรฐานสำหรับรับข้อมูลจากคีย์บอร์ด โดยสามารถรับข้อมูลได้ทุกประเภทไม่ว่าจะเป็นจำนวนเต็ม ทศนิยม หรือข้อความก็ตาม และก่อนจะใช้งานฟังก์ชันดังกล่าวที่ส่วนหัวของโปรแกรมจะต้องประกาศ Header File ที่ชื่อว่า `stdio.h` ก่อน ซึ่งรูปแบบการเรียกใช้คำสั่งดังนี้

รูปแบบ

```
scanf("Format Code", &variable);
```

```
scanf("ข้อความ Format Code", &variable);
```

Format code คือ รหัสควบคุมรูปแบบ เพื่อกำหนดรูปแบบและชนิดของข้อมูลที่จะรับเข้ามาจากคีย์บอร์ด โดยรหัสควบคุมรูปแบบจะใช้ชุดเดียวกับรหัสควบคุมรูปแบบของคำสั่ง `printf()`

&variable คือ ตัวแปรที่จะใช้ข้อมูลที่ได้รับเข้ามาทางคีย์บอร์ด โดยชนิดของตัวแปรจะต้องตรงกับรหัสควบคุมรูปแบบที่กำหนดไว้ นอกจากนี้หน้าชื่อของตัวแปรจะต้องนำหน้าด้วยเครื่องหมาย `&` (Address Operator) ยกเว้นตัวแปรชนิด *String* เท่านั้นที่ไม่ต้องนำหน้าด้วยเครื่องหมาย `&`

ตัวอย่าง

```
scanf("%s", name);
```

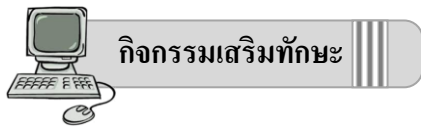
```
scanf("%d", &salary);
```

นอกจากนี้ยังสามารถรับข้อมูลได้มากกว่า 1 ค่าในครั้งเดียวกัน สามารถทำได้โดยกำหนดรูปแบบการรับค่าในคำสั่ง `scanf()` ให้มีรหัสควบคุมรูปแบบ และ ตัวแปรให้เพิ่มขึ้นมากกว่าอย่างละ 1 โดยรหัสควบคุมรูปแบบต้องตรงกับชนิดของตัวแปรแต่ละตัว ดังตัวอย่างต่อไปนี้

ตัวอย่าง

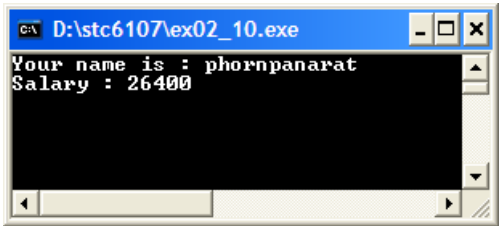
```
scanf("%s %d", name, &salary);
```

```
scanf("%d %d", &base, &high);
```

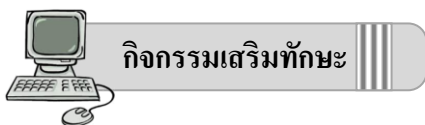


ให้นักเรียนทดลองเขียนโปรแกรมดังต่อไปนี้ โดยบันทึกโปรแกรม ชื่อ ex05_10.cpp

สังเกตผลลัพธ์ที่ได้

<pre> 1 #include <stdio.h> 2 #include <conio.h> 3 char name[20]; 4 int salary; 5 main() 6 { 7 printf("Your name is : "); 8 scanf("%s", name); 9 printf("Salary : "); 10 scanf("%d", &salary); 11 getche(); 12 }</pre>	 <p>รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของ โปรแกรม ex05_10.exe</p>
---	--

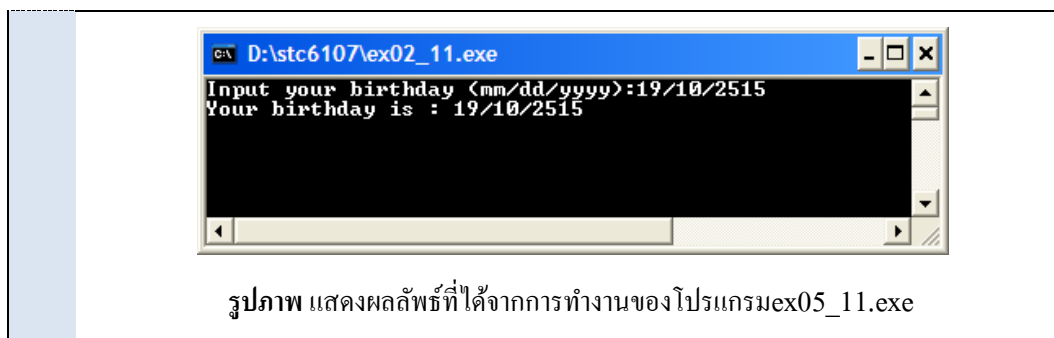
เมื่อสั่งให้โปรแกรม Run ขึ้นมาก็ทำการป้อนชื่อ และเงินเดือน ในการป้อนข้อมูลแต่ละค่า เมื่อป้อนเสร็จแล้วให้กดปุ่ม Enter ก็จะเข้าสู่การทำงานของคำสั่งถัดไป



ให้นักเรียนทดลองเขียนโปรแกรมดังต่อไปนี้ โดยบันทึกโปรแกรม ชื่อ ex05_11.cpp

สังเกตผลลัพธ์ที่ได้

<pre> 1 #include <stdio.h> 2 #include <conio.h> 3 int dd , mm , yy; 4 main() 5 { 6 printf("Input your birthday (dd/mm/yyyy) : "); 7 scanf("%d/ %d/%d", &dd,&mm,&yy); 8 scanf("Your birthday is : %d/ %d/%d", &dd,&mm,&yy); 9 getche(); 10 }</pre>



2. การรับข้อมูลจากคีย์บอร์ดด้วยคำสั่ง getchar() และ getch()

คำสั่งรับข้อมูลประเภทอักขระหรือตัวอักษรจากคีย์บอร์ดโดยเฉพาะนั้น คือ คำสั่ง getchar() และ getch() ทั้ง 2 คำสั่งนี้ใช้รับข้อมูลประเภทอักขระ (char) โดยมีรูปแบบการเรียกใช้คำสั่งดังนี้

รูปแบบ

Variable = getchar();

Variable = getch();

Variable คือ ตัวแปรชนิด char ที่เก็บค่าอักขระหรือตัวอักษรที่รับเข้ามา

ตัวอย่าง

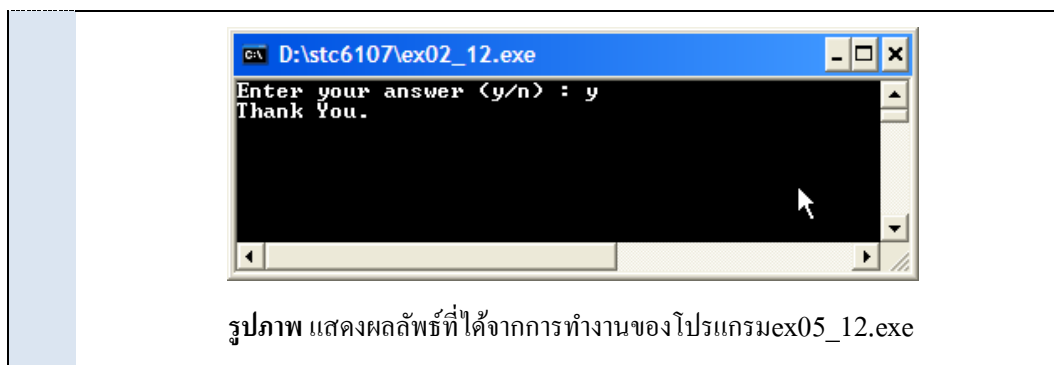
Ans=getchar();



ให้นักเรียนทดลองเขียนโปรแกรมดังต่อไปนี้ โดยบันทึกโปรแกรม ชื่อ ex05_12.cpp สังเกตผลลัพธ์ที่ได้

```

1 #include <stdio.h>
2 #include <conio.h>
3 char ans;
4 main()
5 {
6     printf("Enter your answer (y/n) : ");
7     ans=getchar( );
8     printf("Thank You.");
9     getche( );
10 }
```



3. การรับข้อมูลจากคีย์บอร์ดด้วยคำสั่ง gets()

นอกจากคำสั่งพิเศษในการรับข้อมูลชนิดอักขระหรือตัวอักษรแล้ว การรับข้อความจากคีย์บอร์ดก็มีคำสั่งพิเศษให้เรียกใช้เช่นกันนั่นก็คือ คำสั่ง gets() ซึ่งมีรูปแบบการใช้คำสั่งดังนี้

รูปแบบ

`gets(variable);`

variable คือ ตัวแปรสำหรับเก็บข้อความที่รับเข้ามาจากคีย์บอร์ด

ตัวอย่าง

`gets(name);`

`gets(telephone);`



กิจกรรมเสริมทักษะ

ให้นักเรียนทดลองเขียนโปรแกรมดังต่อไปนี้ โดยบันทึกโปรแกรม ชื่อ ex05_13.cpp สังเกตผลลัพธ์ที่ได้

```

1  #include <stdio.h>
2  #include <conio.h>
3  char name [20],telephone[10];
4  main()
5  {
6    printf("Enter your name : "); gets( name);
7    printf("Enter your telephone : "); gets( telephone);
8    printf("Your name is %s ans Telephone number is %s",name,telephone);
9    getche();
10 }
```

