

บทที่ 7

การใช้งานฟังก์ชัน

7.1 บทนำ

การเขียนโปรแกรมขนาดใหญ่ประกอบด้วยการทำงานหลายส่วนและมีความซับซ้อน นักพัฒนาโปรแกรมจึงแบ่งการทำงานของโปรแกรมออกเป็นส่วนย่อยหรือมอดูล (Module) โดยแต่ละมอดูลมีการทำงานที่เป็นอิสระ ทำให้สามารถควบคุมการดำเนินงานและตรวจสอบการทำงานได้ง่าย สำหรับในภาษาซีเรียกมอดูลว่าฟังก์ชัน การทำงานของฟังก์ชันแบ่งเป็น 2 ประเภทได้แก่ ฟังก์ชันที่เก็บไว้ในคลังมาตรฐานของภาษาซีและฟังก์ชันที่สร้างขึ้นโดยนักพัฒนาโปรแกรม เนื้อหาในบทนี้ได้อธิบายถึงการใช้งานฟังก์ชันที่เกี่ยวข้องกับการพัฒนาโปรแกรมไมโครคอนโทรลเลอร์โดยโปรแกรม C51

7.2 ฟังก์ชันในภาษาซี

ฟังก์ชันในภาษาซีสามารถแบ่งได้ 2 ประเภทคือฟังก์ชันมาตรฐานภาษาซีซึ่งเก็บไว้ในคลังมาตรฐานของภาษาซีและฟังก์ชันที่สร้างขึ้นโดยนักพัฒนาโปรแกรม มีรายละเอียดดังนี้

7.2.1 ฟังก์ชันมาตรฐานภาษาซี

ฟังก์ชันมาตรฐานภาษาซีอยู่ในไลบรารีภาษาซีมาตรฐาน (C Standard Library) ซึ่งประกอบด้วยฟังก์ชันต่างๆ ที่รองรับการเขียนโปรแกรมที่เกี่ยวข้อง เช่น ฟังก์ชันสำหรับการคำนวณทางคณิตศาสตร์ การจัดการกับข้อความ การจัดการกับข้อมูลนำเข้าและส่งออก เป็นต้น โดยการใช้งานฟังก์ชันประเภทนี้ นักพัฒนาโปรแกรมต้องรวมไลบรารีที่ต้องการใช้งานไว้เป็นไฟล์ส่วนหัว (Header file) ของโปรแกรมที่พัฒนาขึ้น เพื่อให้ตัวแปลโปรแกรมรู้ว่าฟังก์ชันที่ต้องการใช้อยู่ในไลบรารีตัวใด

ในที่นี้อธิบายฟังก์ชันการจัดการกับข้อมูลนำเข้าและส่งออก (Standard input/output) ที่ใช้ในการเขียนโปรแกรมไมโครคอนโทรลเลอร์ ซึ่งเป็นฟังก์ชันที่อยู่ในไลบรารีมาตรฐาน ชื่อ `stdio` โดยมีรูปแบบการเขียนโปรแกรม เป็นดังนี้

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

โดยที่ `stdio.h` เป็นชื่อไฟล์ส่วนหัวของไลบรารีมาตรฐาน `stdio` และมีฟังก์ชันการจัดการกับข้อมูลนำเข้าและส่งออกที่สำคัญได้แก่

1. ฟังก์ชัน printf () ใช้ในการแสดงผลข้อมูลหรือค่าตัวแปรออกมาทางจอภาพ โดยมีรูปแบบคำสั่งดังนี้

```
printf (control, argument list);
```

โดยที่

control คือข้อความที่ต้องการให้แสดงผลออกมา

argument list คือชุดตัวแปร ค่าคงที่ หรือ นิพจน์ที่ต้องการนำมาแสดงผล

รายละเอียดของฟังก์ชัน printf อธิบายได้ ดังนี้

1) ส่วนการควบคุม (control) จะต้องเขียนภายใต้เครื่องหมายอัฒประกาศ (“ ”) ซึ่งจะเป็นข้อความที่ต้องการให้แสดงผลออกมาหรือเป็นรหัสรูปแบบที่ใช้ในการแสดงผลซึ่งทุกรหัสรูปแบบจะต้องอยู่ตามหลังเครื่องหมาย ‘%’ รหัสรูปแบบที่นิยมใช้แสดงดังตารางที่ 7.1

ตารางที่ 7.1 รหัสรูปแบบที่ใช้ในฟังก์ชัน printf

รหัสรูปแบบ	ชนิดตัวแปร	ลักษณะการแสดงผลออกจอภาพ
%d	int	ใช้แสดงข้อมูลเลขจำนวนเต็มฐานสิบ
%ld	long int	ใช้แสดงข้อมูลเลขจำนวนเต็มฐานสิบช่วงยาว
%u	unsigned int	ใช้แสดงข้อมูลเลขจำนวนเต็มฐานสิบแบบไม่มีเครื่องหมาย
%c	char	ใช้แสดงข้อมูลตัวอักษร
%s	string	ใช้แสดงข้อมูลตัวแปรสตริงหรือชุดตัวอักษร
%o	int	ใช้แสดงข้อมูลเลขฐานแปด
%x	int	ใช้แสดงข้อมูลเลขฐานสิบหก
%f	float	ใช้แสดงข้อมูลเลขทศนิยม

ที่มา: นवलندا สงวนวงศ์ทอง, 2557

2) ส่วนของอาร์กิวเมนต์ (Argument list) จะเป็นชุดตัวแปร ค่าคงที่ หรือ นิพจน์ที่ต้องการนำมาแสดงผล ถ้ามีมากกว่า 1 ค่าจะแยกออกจากกันโดยใช้เครื่องหมายจุลภาค (,) ซึ่งส่วนของอาร์กิวเมนต์นี้จะมีก็ต่อเมื่อในส่วนการแสดงผลข้อมูลมีการใช้รหัสรูปแบบ แต่ถ้าในส่วนของการแสดงผลข้อมูลเป็นการแสดงข้อความธรรมดาที่ไม่จำเป็นต้องมีส่วนของอาร์กิวเมนต์

3) รหัสพิเศษที่แทรกลงไป ค่าคงที่ที่สตริง (Escape sequence) เพื่อใช้ควบคุมการแสดงผลของตัวอักษรในลักษณะต่างๆ โดยการเขียนต้องมีเครื่องหมาย ‘\’ นำหน้า รหัสควบคุมการแสดงผลได้จากตารางที่ 7.2

ตารางที่ 7.2 รหัสพิเศษต่างๆ ที่มีใช้ในภาษาซี

รหัสพิเศษ	หน้าที่
\a	เสียงดังออกลำโพงหนึ่งครั้ง
\b	เลื่อนตัวชี้ตำแหน่งไปทางซ้ายมือหนึ่งตัวอักษร
\f	ขึ้นหน้าใหม่
\n	ขึ้นบรรทัดใหม่
\r	เลื่อนตัวชี้ตำแหน่งไปทางซ้ายมือสุดของบรรทัด
\t	ตั้ง tab ในแนวนอน
\v	ตั้ง tab ในแนวตั้ง
\\	เครื่องหมาย \
\'	เครื่องหมาย '
\"	เครื่องหมาย "
\?	เครื่องหมาย ?

ที่มา: นวลดดา สงวนวงษ์ทอง, 2557

2. ฟังก์ชัน `scanf ()` ใช้ในการรับข้อมูลมาทางจอภาพ มีลักษณะคล้ายกับฟังก์ชัน `printf` โดยมีรูปแบบคำสั่งดังนี้

```
scanf (control, argument list);
```

รายละเอียดของฟังก์ชัน `scanf` อธิบายได้ ดังนี้

- 1) ส่วนการควบคุมเป็นรหัสรูปแบบที่ใช้ในการรับข้อมูล ต้องเขียนอยู่ภายใต้เครื่องหมายอัฒประกาศและมีรหัสรูปแบบเหมือนกับรูปแบบที่ใช้ในคำสั่ง `printf`
- 2) ส่วนของอาร์กิวเมนต์เป็นตัวแปรที่รับค่าจากแป้นพิมพ์มาเก็บไว้ กรณีที่มีมากกว่าหนึ่งค่าต้องแยกกันด้วยเครื่องหมายจุลภาค และเนื่องจากฟังก์ชัน `scanf ()` เป็นการรับข้อมูลมาเก็บในหน่วยความจำดังนั้นในการใช้งานจึงต้องมีอักขระ '&' นำหน้าตัวแปรเสมอ

3. ฟังก์ชัน `getchar ()` ใช้รับข้อมูลเข้ามาทางแป้นพิมพ์ทีละ 1 ตัวอักษร โดยต้องกดปุ่ม `enter` ทุกครั้งเมื่อสิ้นสุดข้อมูล และข้อมูลที่ป้อนจะปรากฏให้เห็นบนหน้าจอภาพด้วย มีรูปแบบคำสั่งดังนี้

```
getchar ();
```

4. ฟังก์ชัน `putchar ()` ใช้เป็นฟังก์ชันที่ให้คอมพิวเตอร์แสดงผลบนจอภาพทีละ 1 ตัวอักษรโดยมีรูปแบบคำสั่ง ดังนี้

```
putchar ( );
```

การศึกษาการทำงานของไมโครคอนโทรลเลอร์ MCS-51 โดยอาศัยโปรแกรม Keil μ Vision สามารถแสดงผลการทำงานของฟังก์ชันการจัดการกับข้อมูลนำเข้าและส่งออกผ่านทาง Serial Windows ในโหมดดีบั๊กได้ โดยการทำงานดังกล่าวจะรับส่งข้อมูลผ่านพอร์ตอนุกรม RS-232 ซึ่งได้อธิบายรายละเอียดไว้ในบทที่ 9 เรียบร้อยแล้ว ดังนั้นก่อนที่จะมีการใช้งานฟังก์ชันการจัดการกับข้อมูลนำเข้าและส่งออก ต้องมีการกำหนดค่าต่างๆ ในเรจิสเตอร์ที่เกี่ยวข้องกับการใช้งานพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 ด้วยฟังก์ชันดังตัวอย่างโปรแกรมที่ 7.1

ตัวอย่างที่ 7.1 โปรแกรมการกำหนดค่าเรจิสเตอร์ต่างๆ ที่เกี่ยวข้องกับการใช้งานพอร์ตอนุกรม

```
1 void init_serial (void)
2 {
3     SCON = 0x50;
4     PCON &= 0x7F;
5     TMOD |= 0x20;
6     TH1 = 0xFA;
7     TR1 = 1;
8     TI = 1;
9 }
```

- บรรทัดที่ 3 กำหนดค่าเรจิสเตอร์ SCON เพื่อควบคุมการทำงานพอร์ตอนุกรมโหมด 1
- บรรทัดที่ 4 เคลียร์บิต SMOD ของเรจิสเตอร์ PCON
- บรรทัดที่ 5 กำหนดค่ารีโหลดอัตโนมัติให้กับไทม์เมอร์ 1
- บรรทัดที่ 6 กำหนดอัตราบอด 96,000 bps ที่ความถี่ 12 MHz
- บรรทัดที่ 7 เริ่มการทำงานของไทม์เมอร์ 1
- บรรทัดที่ 8 กำหนดบิต TI เพื่อเริ่มต้นส่งข้อมูล

7.2.2 ฟังก์ชันที่สร้างขึ้นโดยนักพัฒนาโปรแกรม

นักพัฒนาโปรแกรมสามารถเขียนฟังก์ชันการทำงานที่จะเรียกใช้ในส่วนต่างๆ ของโปรแกรม โดยฟังก์ชันการทำงานดังกล่าวถูกเขียนไว้ในฟังก์ชันเพียงครั้งเดียวเท่านั้น แต่สามารถเรียกใช้งานได้หลายครั้ง ตัวแปรที่ประกาศหรือคำสั่งที่เรียกในฟังก์ชันใดๆ มีขอบเขตการใช้งานอยู่ในฟังก์ชันนั้นๆ เท่านั้น นั่นหมายความว่าฟังก์ชันอื่นๆ รวมทั้งฟังก์ชัน `main()` จะไม่ทราบการทำงานภายในหรือคำสั่ง

ต่างๆ ในฟังก์ชันนั้น โดยฟังก์ชันจะถูกเรียกให้ทำงานก็ต่อเมื่อมีการเรียกใช้ฟังก์ชัน ซึ่งการเรียกใช้ฟังก์ชันต้องระบุชื่อและค่าอาร์กิวเมนต์ที่ส่งให้กับฟังก์ชันนั้น เพื่อใช้ในการทำงานตามที่ได้ออกแบบไว้

7.3 การประกาศและเรียกใช้งานฟังก์ชัน

การกำหนดฟังก์ชันในโปรแกรมประกอบด้วย 3 ส่วนหลักได้แก่ ต้นแบบของฟังก์ชัน (Function prototype) การกำหนดฟังก์ชัน และการเรียกใช้งานฟังก์ชัน (Function call)

7.3.1 ต้นแบบของฟังก์ชัน

ต้นแบบของฟังก์ชันเป็นส่วนที่กำหนดให้คอมพิวเตอร์รู้ถึงคุณสมบัติต่างๆ ของฟังก์ชัน ได้แก่

1. ชนิดของข้อมูลที่ได้จากการทำงานของฟังก์ชัน
2. จำนวนพารามิเตอร์ที่โปรแกรมส่งให้ฟังก์ชัน
3. ชนิดของพารามิเตอร์
4. ลำดับของพารามิเตอร์

ดังนั้นตัวแปลโปรแกรมจะตรวจสอบการเรียกใช้ฟังก์ชันจากต้นแบบของฟังก์ชัน ซึ่งมีรูปแบบดังนี้

```
return-type-value function-name (arg_type arg_name, arg_type arg_name,...)
หรือ
return-type-value function-name (arg_type, arg_type, arg_type,...)
```

โดยที่

return-type-value	คือชนิดของข้อมูลที่ต้องการส่งค่าผ่านกลับออกไปยังจุดที่เรียกใช้งาน เช่น int หรือ float หากไม่ส่งค่ากลับให้ใช้ void
function-name	คือชื่อของฟังก์ชันที่สร้างขึ้น
arg type	คือชนิดของอาร์กิวเมนต์
arg name	คือชื่อของอาร์กิวเมนต์ อาจใส่หรือไม่ใส่ก็ได้

ตัวอย่างของต้นแบบฟังก์ชัน เช่น ฟังก์ชันชื่อ maximum รับอาร์กิวเมนต์ 3 ตัว คือ x, y และ z ที่มีชนิดข้อมูลเป็น int และคืนผลลัพธ์เป็นข้อมูลชนิด int แสดงได้ดังนี้

```
int maximum (int x, int y, int z);
```

7.3.2 การกำหนดฟังก์ชัน

รูปแบบของการกำหนดฟังก์ชัน สามารถกำหนดได้ ดังนี้

```
return-type-value function-name (arg_type arg_name, arg_type arg_name,...)
{
    declarations;
    statements;
}
```

โดยที่

declarations คือส่วนของการประกาศตัวแปรภายในฟังก์ชัน

statement คือส่วนของคำสั่งที่กระทำในฟังก์ชันนั้น

ตัวอย่างการกำหนดฟังก์ชันชื่อ maximum เพื่อหาค่าสูงสุด แสดงได้ดังโปรแกรมที่ 7.2

ตัวอย่างที่ 7.2 โปรแกรมคำนวณค่าสูงสุดจากข้อมูลจำนวน 3 ค่า

```
01 int maximum(int x,int y,int z)
02 {
03     int max = x;
04     if (y > max)
05         max = y;
06     if (z > max)
07         max = z;
08     return max;
09 }
```

บรรทัดที่ 1 กำหนดฟังก์ชันชื่อ maximum ซึ่งประกอบด้วยอาร์กิวเมนต์ 3 ค่า คือ x, y และ z

บรรทัดที่ 3 กำหนดค่าตัวแปร max เท่ากับค่า x

บรรทัดที่ 4 คำสั่ง if ตรวจสอบเงื่อนไขว่า y มากกว่า max หรือไม่

บรรทัดที่ 5 หากเงื่อนไขเป็นจริง กำหนดให้ max เท่ากับ y

บรรทัดที่ 6 คำสั่ง if ตรวจสอบเงื่อนไขว่า z มากกว่า max หรือไม่

บรรทัดที่ 7 หากเงื่อนไขเป็นจริง กำหนดให้ max เท่ากับ z

บรรทัดที่ 8 คืนค่า max เมื่อฟังก์ชันทำงานเสร็จ

7.3.3 การเรียกใช้งานฟังก์ชัน

การเรียกใช้งานฟังก์ชันในโปรแกรม C51 ทำได้โดยกำหนดค่าให้ฟังก์ชัน โดยอาร์กิวเมนต์จะ
ได้รับค่าที่กำหนดให้และส่งไปให้ฟังก์ชันทำงาน แสดงได้ดังตัวอย่างโปรแกรมที่ 7.3

ตัวอย่างที่ 7.3 โปรแกรมเรียกใช้งานฟังก์ชัน maximum

```

1  #include <REGX52.H>
2  #include <stdio.h>
3
4  void init_serial(void);
5  int maximum(int ,int ,int );
6
7  void main (void)
8  {
9      int a, b, c;
10     init_serial ();
11     printf ("Enter three intergers:");
12     scanf ("%d %d %d", &a, &b, &c);
13     printf("Maximum is: %d\n", maximum(a,b,c));
14     while (1);
15 }

```

- บรรทัดที่ 4 ประกาศฟังก์ชันชื่อ init_serial เพื่อรับส่งข้อมูลผ่านพอร์ตอนุกรม
- บรรทัดที่ 5 ประกาศฟังก์ชันชื่อ maximum เพื่อคำนวณค่าสูงสุดจากข้อมูลจำนวน 3 ค่า
- บรรทัดที่ 9 ประกาศตัวแปรชนิดจำนวนเต็ม a, b และ c
- บรรทัดที่ 10 เรียกใช้งานฟังก์ชัน init_serial
- บรรทัดที่ 11 ฟังก์ชัน printf แสดงข้อความ “Enter three intergers”
- บรรทัดที่ 12 ฟังก์ชัน scanf เพื่อรับข้อมูลจำนวนเต็ม 3 ค่า
- บรรทัดที่ 13 ฟังก์ชัน printf แสดงผลลัพธ์จากการเรียกใช้ฟังก์ชัน maximum

ผลลัพธ์ของการรันโปรแกรม

UART #1
Enter three intergers:10 20 30
Maximum is: 30

ผลลัพธ์การทำงานอธิบายได้ดังนี้ ขั้นตอนแรกให้ผู้ใช้ป้อนข้อมูลชนิดตัวเลขจำนวนเต็มทั้งหมด 3 ค่า จากนั้นจะเรียกใช้ฟังก์ชัน maximum ซึ่งฟังก์ชันนี้มีอาร์กิวเมนต์ 3 ตัว ทำการส่งค่าข้อมูลที่รับมาทั้ง 3 ค่าให้กับฟังก์ชัน เมื่อฟังก์ชันรับค่าตัวเลขเข้ามาจะเก็บไว้ในพารามิเตอร์ x, y และ z ตามลำดับ จากนั้นนำข้อมูลในตัวแปรทั้ง 3 ตัวมาคำนวณหาค่าสูงสุดและแสดงผลผ่านพอร์ตอนุกรม (UART #1) โดยเมื่อเสร็จโปรแกรมแล้วจะหยุดการทำงานโดยฟังก์ชัน while

7.4 การส่งค่าให้ฟังก์ชัน

การเขียนโปรแกรมโดยใช้ฟังก์ชันที่ไม่มีการคืนค่ากลับนั้น เมื่อโปรแกรมทำงานเสร็จชนิดของข้อมูลที่ต้องการจะส่งค่าผ่านกลับออกไปยังจุดที่เรียกใช้งาน ต้องประกาศเป็นคำว่า void และไม่มีคำสั่ง return ในการคืนค่ากลับ อย่างไรก็ตามในการพัฒนาโปรแกรมโดยส่วนใหญ่อาศัยการทำงานของฟังก์ชันที่สามารถคืนค่ากลับได้ เนื่องจากโปรแกรมต้องการนำข้อมูลจากฟังก์ชันไปประมวลผลต่อ เช่น ฟังก์ชันหาผลลัพธ์ของเลขยกกำลัง ฟังก์ชันสำหรับหาค่าเฉลี่ย หรือฟังก์ชันสำหรับหาค่าสูงสุด เป็นต้น ซึ่งรูปแบบของฟังก์ชันที่มีการคืนค่ากลับ มีดังนี้

7.4.1 ฟังก์ชันที่ส่งค่าข้อมูลผ่านฟังก์ชัน

ฟังก์ชันที่ส่งค่าข้อมูลผ่านฟังก์ชันเป็นการส่งค่าให้ฟังก์ชันโดยการคัดลอกค่าจากอาร์กิวเมนต์ของฟังก์ชันไปเก็บไว้ในพารามิเตอร์ ซึ่งกระบวนการในฟังก์ชันจะไม่ส่งผลกระทบต่อค่าอาร์กิวเมนต์ การคืนค่ากลับมาจากฟังก์ชันจะคืนค่ากลับมาได้ค่าเดียวเท่านั้น แม้ว่าฟังก์ชันสามารถรับอาร์กิวเมนต์ได้หลายตัวก็ตาม โดยการคืนค่าต้องบอกตัวแปรโปรแกรมเพื่อให้ทราบถึงชนิดข้อมูล เช่น char หรือ int เป็นต้น การคืนค่ากลับจะใช้คำสั่ง return ตามด้วยค่าที่ต้องการซึ่งอาจเป็น ค่าคงที่ ตัวแปร หรือค่าจากการทำงานของนิพจน์ใดๆ

7.4.2 ฟังก์ชันที่ส่งค่าอ้างอิงหรือแอดเดรสผ่านฟังก์ชัน

การส่งค่าของตำแหน่งหน่วยความจำที่นำค่าไปเก็บไว้ในพารามิเตอร์ กระบวนการทำงานของฟังก์ชันในลักษณะนี้สามารถเปลี่ยนแปลงค่าที่เก็บไว้ในหน่วยความจำที่ตำแหน่งที่ส่งมาให้ได้ เมื่อฟังก์ชันทำงานเสร็จ อาร์กิวเมนต์อาจมีการเปลี่ยนแปลงค่าไปจากเดิมก่อนการเรียกใช้งานฟังก์ชัน

7.5 ตัวแปรชนิดโลคอลและโกลบอล

ตัวแปรชนิดโลคอล (Local Variable) คือ ตัวแปรที่ถูกสร้างขึ้นภายในฟังก์ชัน สามารถเรียกใช้งานได้เฉพาะภายในฟังก์ชันที่สร้างขึ้น และจะถูกทำลายลงเมื่อเสร็จสิ้นการทำงานของฟังก์ชันนั้นๆ ส่วนตัวแปรชนิดโกลบอล (Global Variable) คือ ตัวแปรที่ถูกสร้างขึ้นภายนอกฟังก์ชัน สามารถใช้งานได้ในทุกฟังก์ชันหรือทั้งโปรแกรม ยกเว้นฟังก์ชันที่มีตัวแปรภายในชื่อเดียวกับตัวแปรภายนอกและจะคงอยู่ตลอดการทำงานของโปรแกรม

7.6 สรุป

ในบทนี้ได้นำเสนอเนื้อหาเรื่องการใช้งานฟังก์ชันของโปรแกรม C51 โดยได้อธิบายฟังก์ชันเกี่ยวกับการจัดการกับข้อมูลนำเข้าและส่งออกที่ใช้ในการเขียนโปรแกรมไมโครคอนโทรลเลอร์ซึ่งอยู่ในไลบรารีมาตรฐานภาษาซี และฟังก์ชันที่สร้างขึ้นโดยนักพัฒนาโปรแกรมซึ่งมีรูปแบบการประกาศและเรียกใช้งานในลักษณะเดียวกัน อย่างไรก็ตาม ในการทดลองการทำงานโดยโปรแกรม Keil μ Vision สามารถแสดงผลการทำงานของฟังก์ชันการจัดการกับข้อมูลนำเข้าและส่งออกผ่านทาง Serial Windows ได้โดยการกำหนดค่าต่างๆ ในเรจิสเตอร์ที่เกี่ยวข้องกับการใช้งานพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 ซึ่งจะได้อธิบายรายละเอียดการทำงานในบทที่ 9 ต่อไป

7.7 แบบฝึกหัดท้ายบท

แบบฝึกหัดมีทั้งหมด 5 ข้อ ให้นักศึกษาทำแบบฝึกหัดทุกข้อ

1. จงแปลงองศาฟาเรนไฮต์ให้เป็นองศาเซลเซียส โดยใช้สูตร $(C/5) = (F-32)/9$

โดยที่ C คือ องศาเซลเซียส

F คือ องศาฟาเรนไฮต์

โดยกำหนดรูปแบบผลลัพธ์ดังนี้

Farenhite	Celcius
100	xx.xx
80	xx.xx
60	xx.xx
40	xx.xx
20	xx.xx
0	xx.xx

100 xx.xx

80 xx.xx

60 xx.xx

40 xx.xx

20 xx.xx

0 xx.xx

2. จงหาผลลัพธ์ของ $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$ โดยรับค่า n จากผู้ใช้

3. จงหาผลลัพธ์ของ $1 + 2 + 3 + \dots + n$ โดยรับค่า n จากผู้ใช้ โดยต้องการทราบเพิ่มเติมว่าพจน์ที่ทำให้ผลรวมดังกล่าวมีค่ามากกว่า 100 เป็นพจน์แรกคือค่าอะไร

4. จงเขียนโปรแกรมแสดงตารางผลคูณของตัวแปร i และ j โดยกำหนดให้ i และ j มีค่าตั้งแต่ 1 ถึง 12
5. จงเขียนโปรแกรมทดสอบจำนวนที่รับเข้ามาจากผู้ใช้งาน โดยกำหนดให้อยู่ในช่วง 0-99 และทำการทดสอบว่าจำนวนดังกล่าวมากกว่าหรือน้อยกว่า 50 หากจำนวนที่รับเข้ามามีค่ามากกว่า 50 ให้หาค่ากำลังสามของจำนวนนั้น หากน้อยกว่า 50 ให้หาค่ากำลังสองของจำนวนนั้น