บทที่ 1 เริ่มต้นกับภาษา C (Introduction to C)

บทนี้จะกล่าวถึงการเริ่มต้นเขียนโปรแกรมด้วยภาษา C โดยจะกล่าวถึงโครงสร้างของภาษาที่จำเป็นสำหรับการเขียนโปรแกรมด้วยภาษา C โดยทั่วไป

ทั้งนี้จะนำเสนอด้วยโปรแกรมที่ทำงานในลักษณะต่าง ๆ กัน พร้อมคำอธิบายโปรแกรมโดยสังเขป และตามด้วยคำอธิบายวิธีเขียนคำสั่งที่เกี่ยวข้องของโปรแกรมต่าง ๆในตัวอย่าง

1.1 ตัวอย่างโปรแกรมภาษา C

ตัวอย่างที่ 1.1โปรแกรมพิมพ์ข้อความ

วัตถุประสงค์ เพื่อพิมพ์ข้อความที่จอแสดงผล

โปรแกรม

```
1.  /* Write string to output device */
2.  #include <stdio.h>
3.  main()
4.  {
5.          printf("Hello World");
6.          return 0;
7.  } /* End main  */
```

ผลลัพธ์ที่ได้

```
Hello World
```

บรรทัดที่	**คำอธิบายโดยสังเขป**
1	เป็นประโยคหมายเหตุ ระบุวัตถุประสงค์ของการทำโปรแกรม ตัวแปลภาษาจะไม่สนใจการทำงานของประโยคหมายเหตุ ลักษณะของประโยคหมายเหตุจะขึ้นต้น "/*" และจบลงด้วย "*/"
2	เป็นคำสั่งเพื่อนำฟังก์ชันต่างๆ ในแฟ้ม stdio.h เข้ามาในโปแรแกรม ก่อนที่ตัวแปลภาษา(compiler) จะดำเนินการ ซึ่งแฟ้ม stdio.h นี้เป็นแฟ้มที่บรรจุฟังก์ชันเกี่ยวกับการนำข้อมูลนำเข้าและการแสดงผลลัพธ์
3	ฟังก์ชัน main() เป็นฟังก์ชันหลักของโปรแกรม ซึ่งจำเป็นต้องมีในทุกโปรแกรม
4	เป็นเครื่องหมายแสดงจุดเริ่มต้นของฟังก์ชัน main()

5	ฟังก์ชัน printf() เป็นฟังก์ชันที่มีอยู่ในแฟ้ม stdio.h ที่สั่งให้พิมพ์ข้อความในเครื่องหมายคำพูดออกทางจอแสดงผล
6	ระบุการส่งค่าคืนไปยังโปรแกรม
7	เป็นเครื่องหมายแสดงจุดสิ้นสุดของฟังก์ชัน main()

ตัวอย่างที่ 1.2 โปรแกรมพิมพ์ข้อความหลายบรรทัด

วัตถุประสงค์ เพื่อพิมพ์ข้อความหลายบรรทัดบนจอแสดงผล โดยมีการจัดรูปแบบของการแสดงผลให้แต่ละบรรทัดเว้นจากริมขวา 1 แท็บ

โปรแกรม

```
1.  /* Write string to output device */
2.  #include <stdio.h>
3.  main()
4.  {
5.          printf("\tHello\n");
6.          printf("\tWorld");
7.          return 0;
8.  } /* End main  */
```

ผลลัพธ์ที่ได้

```
    Hello
    World
```

บรรทัดที่	**คำอธิบายโดยสังเขป**
1–4	เหมือนโปรแกรมในตัวอย่างที่ 1.1
5-6	การใช้ ฟังก์ชัน printf() จะไม่ขึ้นบรรทัดใหม่ให้โดยอัตโนมัติ ผู้ใช้ที่ประสงค์จะจัดตำแหน่งการพิมพ์ เช่นการตั้งแท็บต้องใช้ \t หรือ ขึ้นบรรทัดใหม่ต้องใช้ \n
7	ระบุการส่งค่าคืนไปยังโปรแกรม
8	เป็นเครื่องหมายแสดงจุดสิ้นสุดของฟังก์ชัน main()

ตัวอย่างที่ 1.3 โปรแกรมพิมพ์จำนวนเต็มตั้งแต่ 1 ถึง 10 โดยพิมพ์บรรทัดละหนึ่งจำนวน

วัตถุประสงค์ เพื่อพิมพ์จำนวนเรียงลำดับจาก 1- 10 โดยมีรูปแบบการแสดงผล ให้เว้นจากริมขวา 1 แท็บ และพิมพ์บรรทัดละหนึ่งจำนวน

โปรแกรม

```
1  /* print running number */
2  #include <stdio.h>
3  main()
4  {
5          int i;
6          for (i=1; i<=10 ; i=i+1)
7          {
8                  printf("\t%d\n",i);
9          } /* End for  */
10         return 0;
11 } /* End main  */
```

ผลลัพธ์ที่ได้

```
1
2
3
4
5
6
7
8
9
10
```

บรรทัดที่ **คำอธิบายโดยสังเขป**

1-4 เหมือนโปรแกรมในตัวอย่างที่ 1.1

5 ประกาศตัวแปรตัวแปร i ให้มีประเภทข้อมูลเป็นจำนวนเต็ม int

6-9 เป็นการทำงานในวงวน for ซึ่งมีรายละเอียดดังนี้

บรรทัดที่ 6 เป็นการกำหนดนิพจน์เริ่มต้นเป็น i=1 ; นิพจน์ทดสอบเป็น i<=10; และนิพจน์เพิ่มค่าให้วงวนคือ i= i+1;

บรรทัดที่ 8 เป็นคำสั่งพิมพ์ค่า i ในแต่ละรอบ โดยระบุการพิมพ์ค่าจำนวนเต็มด้วย %d

10-11 เหมือนบรรทัด 6-7 ของโปรแกรมในตัวอย่างที่1.1

ตัวอย่างที่ 1.4 โปรแกรมสร้างตารางผลคูณ

วัตถุประสงค์ เพื่อพิมพ์ตารางผลคูณของจำนวน ตั้งแต่ 1 ถึง 12 ด้วย 2 3 4 และ 5 ตามลำดับ โดยมีรูปการแสดงผล ให้เว้นจากริมขวา 1 แท็บ ให้ผลคูณแต่ละค่าอยู่ห่างจากตำแหน่งทางซ้ายมือ 1 แท็บ และระยะห่างระหว่างสดมภ์เป็น 1 แท็บ

โปรแกรม

```
1.  /* arithmetic expression */
2.  #include <stdio.h>
3.  main()
4.  {
5.          int i;
6.          printf("\ti\ti*2\ti*3\ti*4\ti*5\n" );        /*column header*/
7.          for (i=1; i<=12 ; i=i+1)
8.          {
9.                  printf("\t%d\t%d\t%d\t%d\t%d\n",i,i*2,i*3,i*4,i*5 );
10.         }/* End for */
11.         return 0;
12. }/* End main */
```

ผลลัพธ์ที่ได้

```
i       i*2     i*3     i*4     i*5
1       2       3       4       5
2       4       6       8       10
3       6       9       12      15
4       8       12      16      20
5       10      15      20      25
6       12      18      24      30
7       14      21      28      35
8       16      24      32      40
9       18      27      36      45
10      20      30      40      50
11      22      33      44      55
12      24      36      48      60
```

บรรทัดที่ **คำอธิบายโดยสังเขป**

1-5 เหมือนโปรแกรมในตัวอย่างที่ผ่านมา

6 เป็นคำสั่งพิมพ์หัวตาราง

7-10 เป็นคำสั่งวงวน for เพื่อพิมพ์ผลคูณ 12 บรรทัดในแต่ละรอบ พิมพ์ 5 จำนวนในหนึ่งบรรทัด พิมพ์ค่า i และค่าผลคูณระหว่าง i กับ 2 3 4 และ 5 ให้สังเกตว่า รูปแบบของการพิมพ์จำนวนเต็มคือ %d จะต้องมีจำนวนเท่ากับจำนวนของค่าที่ต้องการพิมพ์ซึ่งในที่นี้คือ 5 จำนวน

11-12 คล้ายตัวอย่างข้างต้น

ตัวอย่างที่ 1.5 โปรแกรมพิมพ์ผลคูณโดยมีการรับข้อมูลนำเข้าจากผู้ใช้

วัตถุประสงค์ เพื่อพิมพ์ตารางผลคูณของจำนวนเต็มที่เริ่มจาก 10 ถึงจำนวนที่รับเข้าจากผู้ใช้และให้แสดงผลเช่นเดียวกับรูปแบบของโปรแกรมในตัวอย่างที่ 1.4

โปรแกรม

```
1.  /* arithmetic expression  */
2.  #include <stdio.h>
3.  main()
4.  {
5.          int i , max;
6.          printf ("Please Enter the value greater than 10 : ");
7.          scanf ("%d",&max);
8.          printf("\ti\ti*2\ti*3\ti*4\ti*5\n" );
9.          for (i=10; i<=max ; i=i+1)
10.         {
11.                 printf("\t%d\t%d\t%d\t%d\t%d\n",i,i*2,i*3,i*4,i*5 );
12.         }/* End for  */
13.        return 0;
14. }/* End main  */
```

ผลลัพธ์ที่ได้

```
Please Enter the value greater than 10 : 25
        i       i*2     i*3     i*4     i*5
        10      20      30      40      50
        11      22      33      44      55
        12      24      36      48      60
        13      26      39      52      65
        14      28      42      56      70
        15      30      45      60      75
        16      32      48      64      80
        17      34      51      68      85
        18      36      54      72      90
        19      38      57      76      95
        20      40      60      80      100
        21      42      63      84      105
        22      44      66      88      110
        23      46      69      92      115
        24      48      72      96      120
        25      50      75      100     125
```

บรรทัดที่	**คำอธิบายโดยสังเขป**
7	ฟังก์ชัน scanf() เป็นฟังก์ชันหนึ่งในแฟ้ม stdio.h สำหรับรับค่าจากแป้นพิมพ์เพื่อไปเก็บไว้ในตัวแปร max ซึ่งต้องใส่เครื่องหมาย & นำหน้า
บรรทัดอื่น ๆ	คล้ายตัวอย่างข้างต้น

ตัวอย่างที่ 1.6 โปรแกรมคำนวณหาเงินรวมที่ได้รับในแต่ละปี จากการฝากเงิน

❇ ข้อมูลเบื้องต้นในการเขียนโปรแกรม

สถาบันการเงินแห่งหนึ่ง ได้จัดทำโปรแกรมเพื่อเอื้ออำนวยกับลูกค้าในการขอดูเงินรวมในแต่ละปี ซึ่งธนาคารให้ดอกเบี้ย 8.75% โดยให้ลูกค้าสามารถกรอกยอดฝากเริ่มต้นได้

โปรแกรมที่ใช้ในการจัดการรับข้อมูลนำเข้า และแสดงผลลัพธ์ที่เกี่ยวข้องกับโจทย์ เจ้าหน้าที่ของสถาบันการเงินแห่งนั้น เขียนโปรแกรมด้วยภาษา C

ทั้งนี้การจัดทำโปรแกรมดังกล่าวมีขั้นตอนการทำงานดังนี้

❇ ขั้นตอนที่ 1 กำหนดรูปแบบผลลัพธ์

รูปแบบผลลัพธ์	**คำอธิบาย**
Capital : XXXXXX.XX bahts	หัวรายงานบรรทัดที่1
Interest Rate : 8.75%	หัวรายงานบรรทัดที่2
YEAR : GRAND TOTAL (bahts)	หัวสดมภ์
1 : XXXXXX.XX	รายละเอียดบรรทัดที่ 1
2 : XXXXXX.XX	รายละเอียดบรรทัดที่ 2
3 : XXXXXX.XX	รายละเอียดบรรทัดที่ 3
4 : XXXXXX.XX	รายละเอียดบรรทัดที่ 4
5 : XXXXXX.XX	รายละเอียดบรรทัดที่ 5

❇ ขั้นตอนที่ 2 ข้อมูลที่ใช้

ชื่อข้อมูล	**ประเภทข้อมูล**	**คำอธิบาย**
grand_total	float	จำนวนเงินฝาก
INTEREST	float	ดอกเบี้ยมีค่า = 8.75 เป็นค่าคงที่
i	int	จำนวนปีที่ฝากเริ่มจาก 1

❇ ขั้นตอนที่ 3 การทำงานของโปรแกรม

❇ ประกาศตัวแปรที่ใช้งาน

❇ รับข้อมูลนำเข้า

-จำนวนเงินฝากเก็บไว้ในตัวแปรชื่อ grand_total

❇ พิมพ์หัวรายงาน

-พิมพ์หัวรายงาน (heading) 2 บรรทัด

-พิมพ์หัวสดมภ์ (column heading)

✽ พิมพ์รายละเอียด

◆ ทำงานต่อไปนี้ทำการวนซ้ำ i รอบ ในที่นี้ เท่ากับ 5

-เงินสุทธิเมื่อสิ้นปี = ((เงินฝากต้นปี(จากรอบก่อน) *ดอกเบี้ย)/100) + เงินฝากต้นปี(จากรอบก่อน)

```
grand_total=(((grand_total *INTEREST)/100)+grand_total);
```

-พิมพ์ i และ grand_total

✽ จบงาน

✽ ขั้นตอนที่ 4 ลงรหัส รันโปรแกรม และทดสอบผล

```
1  /* Show capital and iterest after each year */
2  #include <stdio.h>
3  #define INTEREST 8.75
4
5  main()
6  {
7          float grand_total;
8          int i;
9          printf("Enter your first amount ");
10         scanf("%f",&grand_total);
11         printf("Capital \t:\t%9.2f baht\n",grand_total);     /* \t = tab , \n = new line*/
12         printf("Interest Rate \t:\t%4.2f\%\n",INTEREST);
13         printf("\tYEAR\t:\tGRAND TOTAL \(bahts\)\n");
14         for (i=1; i<=5 ; i=i+1)
15         {
16                 grand_total=(((grand_total *INTEREST)/100)+grand_total);
17                 printf("\t%d\t:\t %9.2f\n",i,grand_total);
18         }/* End for */
19         return 0;
20 }/* End main  */
```

ผลลัพธ์ที่ได้

```
Enter your first amount : 100000
Capital          :       100000.00 baht
Interest Rate    :       8.75%
        YEAR     :       GRAND TOTAL (bahts)
        1        :        108750.00
        2        :        118265.62
        3        :        128613.87
        4        :        139867.58
        5        :        152105.98
```

1.2 การเขียนประโยคหมายเหตุ

ประโยคหมายเหตุ เป็นประโยคที่ใช้อธิบายความ หรือขยายความ ที่ผู้เขียนโปรแกรมประสงค์ที่จะสร้างประโยคดังกล่าวให้ผู้อื่นเข้าใจเนื้อหาของโปรแกรมหรือเขียนไว้เพื่อเตือนความจำเพื่อการแก้ไขโปรแกรมในอนาคต

การเขียนประโยคหมายเหตุ เป็นการเขียนเพิ่มเติมเข้าในโปรแกรม โดยประโยคหมายเหตุนี้มิได้มีบทบาทใด ๆในการทำงานของโปรแกรม ตัวแปลภาษา (compiler) จะไม่สนใจประโยคที่เห็นว่าเป็นประโยคหมายเหตุ

รูปแบบการเขียนประโยคหมายเหตุ จะเริ่มต้นด้วยเครื่องหมาย "/*" (slash และ asterisk) และจบลงด้วยเครื่องหมาย "*/" (asterisk และ slash)

ผู้เขียนโปรแกรมสามารถวางประโยคหมายเหตุไว้ ณ ตำแหน่งใด ๆ ในโปรแกรมก็ได้ เช่น

จากตัวอย่างที่ 1.6 เราจะพบ ประโยคหมายเหตุในบรรทัดที่ 1 วางไว้ที่ต้นโปรแกรมเพื่ออธิบายวัตถุประสงค์ของโปรแกรม ในบรรทัดที่ 11 วางไว้ที่ท้ายประโยคคำสั่งเป็นการขยายความของประโยคคำสั่งทั้งคำสั่ง หรืออธิบายถึงบางส่วนในประโยคคำสั่งที่อยู่ข้างหน้า

1.3 ตัวประมวลผลก่อน (preprocessor) ตัวแปลภาษา

โปรแกรมในกลุ่มนี้จัดเป็นโปรแกรมพิเศษที่ภาษา C จะทำงานกับคำสั่งของโปรแกรมส่วนนี้ก่อนที่จะส่งคำสั่งที่เหลือไปให้ตัวแปลภาษา (Compiler) จัดการ เรียกโปรแกรมในกลุ่มนี้ว่า preprocessor ลักษณะของคำสั่งในกลุ่มโปรแกรมนี้ จะขึ้นต้นประโยคด้วยเครื่องหมาย # (sharp) ซึ่งมีรายละเอียดการเรียกใช้แตกต่างกันไป สำหรับในบทนี้จะกล่าวถึงเฉพาะส่วนที่มีการเรียกใช้ในตัวอย่างโปรแกรมข้างต้น

1.3.1 คำสั่ง # include <stdio.h>

คำสั่งนี้จะพิมพ์ # ที่สดมภ์ที่ 1 มีหน้าที่ในการระบุให้รวมเนื้อหาทั้งหมดในแฟ้ม stdio.h เข้าไปในโปรแกรม หลังจากนั้นจึงส่งโปรแกรมทั้งหมดไปให้ตัวแปลภาษา นั่นหมายความว่าตัวแปลภาษาจะมองไม่เห็นคำสั่ง #include <stdio.h> แต่จะมองเห็นข้อมูลภายในแฟ้ม stdio.h แทน ซึ่งข้อมูลในแฟ้ม stdio.h เป็นฟังก์ชันสำหรับรับข้อมูลเข้าและแสดงผล คำสั่งนี้มักพบเป็นบรรทัดแรกของโปรแกรมเสมอ

จากตัวอย่างโปรแกรมที่ 1.6 มีการกำหนดในบรรทัดที่ 2

1.3.2 คำสั่ง #define NAME VALUE

คำสั่งนี้เป็นอีกคำสั่งหนึ่งในกลุ่มของ preprocessor มีหน้าที่ในการกำหนดค่าคงตัว (constant) โดยมีรูปแบบการเขียนเพื่อกำหนดค่าคงที่ ดังนี้

#define NAME VALUE

ซึ่งมีความหมายว่าให้กำหนดค่าให้กับตัวระบุ NAME ทั้งนี้ #define ต้องเริ่มเขียนที่สดมภ์ ที่ การกำหนดค่าคงตัวลักษณะนี้สามารถเขียนไว้ในตำแหน่งใด ๆ ในโปรแกรมก็ได้ แต่โดยทั่วไปแล้วคำสั่งในกลุ่ม preprocessor นี้นิยมเขียนไว้ที่ส่วนต้นของโปรแกรม และในการกำหนดค่าคงที่นี้ ชื่อตัวระบุสำหรับการเก็บค่าคงที่มักนิยมเขียนด้วยตัวพิมพ์ใหญ่ทั้งหมด เพื่อบอกข้อแตกต่างจากตัวแปรลักษณะอื่น ๆ

จากตัวอย่างโปรแกรมที่ 1.6 มีการกำหนดในบรรทัดที่ 3 เป็น

#define INTEREST 8.75

เมื่อ preprocessor พบคำสั่ง #define จะกำหนดค่า 8.75 ให้กับตัวระบุ ที่ปรากฏอยู่ทุกๆ ที่ในโปรแกรม แล้วส่งโปรแกรมทั้งหมดไปให้ตัวแปลภาษา นั่นหมายความว่าตัวแปลภาษาจะมองไม่เห็นตัวแปลที่ชื่อ INTEREST หากแต่จะเห็นค่า 8.75 แทน

ข้อดีของการกำหนดค่าคงตัวในลักษณะนี้ คือ ทำให้โปรแกรมอ่านแล้วเข้าใจได้ง่าย และสามารถเปลี่ยนแปลงค่าได้ง่าย เพราะการเปลี่ยนค่าคงตัวลักษณะนี้จะทำการเปลี่ยนค่าที่ส่วนประกาศเท่านั้น เมื่อใส่ค่าใหม่ในครั้งต่อไป ค่าใหม่จะเข้าไปแทนที่ในตัวระบุตามต้องการ

1.4 ฟังก์ชัน main( )

ฟังก์ชันในภาษา C จะประกอบด้วยสองส่วน คือ

1.4.1 ส่วนหัวของฟังก์ชัน ประกอบด้วย ชื่อฟังก์ชันและรายการพารามิเตอร์ โดยปกติแล้วรายการพารามิเตอร์จะปรากฏในวงเล็บที่ตามหลังชื่อฟังก์ชัน สำหรับตัวอย่างในโปรแกรมที่ 1.6 ฟังก์ชันในบรรทัดที่ 5 มีชื่อว่า main และไม่มีพารามิเตอร์ จึงแสดงด้วยรายการว่าง ( )

1.4.2 รายละเอียดของฟังก์ชัน

ประกอบด้วยการประกาศค่าตัวแปรและคำสั่งต่าง ๆ ที่จะทำงานเมื่อฟังก์ชันถูกเรียก คำสั่งต่าง ๆ ในส่วนนี้จะบรรจุไว้ในเครื่องหมายวงเล็บปีกการเปิด "{" และปิดด้วยเครื่องหมายวงเล็บปีกกาปิด "}" จากตัวอย่างโปรแกรมที่ 1.6 พบว่า

บรรทัดที่	คำสั่ง	คำอธิบาย
5	main( )	ชื่อฟังก์ชัน ***(หัวของฟังก์ชัน)***
6	{	***ประกาศตัวแปรและคำสั่งต่าง ๆ (ลำตัวของฟังก์ชัน)***
19	}	

สำหรับฟังก์ชัน main เป็นฟังก์ชันหลักของโปรแกรมทุกโปรแกรม การทำงานทั้งหมดของโปรแกรมนี้มีความหมายพิเศษ โดยจะเป็นตัวระบุจุดเริ่มต้นการทำงานของโปรแกรมทุกโปรแกรม นั่นแปลว่าโปรแกรมทุกโปรแกรมต้องมีฟังก์ชัน main อยู่ด้วยเสมอ

1.5 การประกาศตัวแปร

จะต้องประกาศประเภทข้อมูลของตัวแปรทุกตัวที่เรียกใช้ จากตัวอย่างโปรแกรมที่ 1.6 บรรทัดที่ 7 และ 8 มีการประกาศตัวแปรดังนี้

```
float grand_total ;
int i ;
```

โดยระบุว่าตัวแปรชื่อ grand_total มีประเภทข้อมูลเป็น float และตัวแปร i มีประเภทข้อมูลเป็นจำนวนเต็ม โดยระบุเป็น int

การประกาศตัวแปรนี้เป็นการจองเนื้อที่ในหน่วยความจำ (storage) เพื่อจัดเก็บข้อมูล

การระบุประเภทข้อมูลเป็น int หมายความว่าค่าของข้อมูลที่จะจัดเก็บในตัวแปรจะต้องเป็นจำนวนเต็ม

1.6 คำสั่งการกำหนดค่า

คำสั่งการกำหนดค่า เป็นการกำหนดค่าซึ่งเป็นค่าของนิพจน์ที่อยู่ทางด้านงขวามือของเครื่องหมายเท่ากับให้กับตัวแปรทางซ้ายมือของเครื่องหมายเท่ากับ เช่น

i=i+1; ในบรรทัดที่ 14 ของโปรแกรมในตัวอย่างที่ 1.6 เป็นการกำหนดค่าของ i+1 ให้กับตัวแปร i

1.7 ฟังก์ชัน printf

printf เป็นฟังก์ชันที่มีอยู่ในคลังแฟ้มคำสั่งมาตราฐาน stdio.h ไว้เรียบร้อยแล้วที่ผู้ใช้สามารถเรียกใช้ได้ โดยต้องใช้คำสั่ง include ไว้ก่อน คือ #include <stdio.h> ไว้ที่ส่วนต้นของโปรแกรม ซึ่งมีหน้าที่ในการแสดงผลลัพธ์บนอุปกรณ์ส่งออกมาตรฐาน เช่น จอภาพ

รูปแบบคำสั่ง

printf("control string", arg_1, arg_2, arg_3,…, arg_n);

โดยทั่วไป printf จะแสดงทุกสิ่งทุกอย่างที่ระบุไว้ใน control string ที่อยู่ภายใต้เครื่องหมายคำพูด แต่มีข้อยกเว้นสำหรับอักขระที่ตามหลังเครื่องหมาย % เพราะอักขระที่ตามหลังเครื่องหมาย % จะเป็นการระบุประเภทของข้อมูล เช่น

%d หมายถึงข้อมูลประเภทจำนวนเต็ม

%f หมายถึงข้อมูลประเภทจำนวนจริง

control string ประกอบด้วยสายอักขระและรูปภาพการแสดงผลของอากิวเมนท์ เช่น

```
printf("Enter the value greater than 10 ");
```

หรือ

```
printf("\t %d \t: \t %9.2t \n",i,grand_total);
```

ในที่นี้อากิวเมนต์คือ i และ grand_total โดยทั่วๆ ไป อากิวเมนต์อาจเป็นตัวแปรหรือนิพจน์ก็ได้

ให้สังเกตว่า การระบุรูปแบบให้คำสั่ง printf จำนวนรูปแบบอากิวเมนท์และจำนวนอากิวเมนท์จะต้องเท่ากันเสมอ จะมีอย่างใดอย่างหนึ่งมากกว่าหรือน้อยกว่ากันไม่ได้ เพราะจะทำให้ทำงานผิดจากความเป็นจริง

1.8 คำสั่ง scanf

scanf เป็นฟังก์ชันที่มีอยู่ในคลังแฟ้มคำสั่งมาตราฐาน stdio.h ไว้เรียบร้อยแล้ว ที่ผู้ใช้สามารถเรียกใช้ได้ โดยต้องใช้คำสั่ง include ไว้ก่อน คือ #include <stdio.h> ไว้ที่ส่วนต้นของโปรแกรม ซึ่งมีหน้าที่ในการแสดงผลลัพธ์บนอุปกรณ์ส่งออกมาตรฐาน เช่น จอภาพ

รูปแบบคำสั่ง

scanf ("control string", &arg_1, &arg_2, &arg_3,…, &arg_n);

control string จะมีรูปแบบเช่นเดียวกับฟังก์ชัน printf ให้สังเกตว่า สำหรับฟังก์ชัน scanf การเขียนอากิวเมนท์จะต้องมีเครื่องหมาย & นำหน้า

ตัวอย่างที่ 1.7 โปรแกรมแสดงการรับข้อมูลหลาย ๆ จำนวนพร้อมกันจากแป้นพิมพ์

```
1.  /* Receive more than one input from keyboard */
2.  #include <stdio.h>
3.  main()
4.  {
5.          float num1,num2,num3;
6.          printf ("Enter 3 numbers ");
7.          scanf ("%f %f %f",&num1,&num2,&num3);
8.          printf("The average = \t%7.2f\n", ((num1+num2+num3)/3));
9.          return 0;
10. } /* End main  */
```

ผลลัพธ์ที่ได้

```
Enter 3 numbers 15.8 25.3 19.00
The average =     20.03
```

หมายเหตุ การป้อนข้อมูลนำเข้าแต่ละจำนวน ให้เว้นช่องว่างระหว่างจำนวนข้อมูลแต่ละตัว

บรรทัดที่	**คำอธิบายโดยสังเขป**
7	แสดงการรับข้อมูลประเภทจำนวนจริง 3 จำนวน ให้สังเกตว่า รูปแบบของการรับจำนวนจริง คือ %f
บรรทัดอื่น ๆ	คล้ายตัวอย่างข้างต้น

1.9 คำสั่ง return

ปกติแล้วการทำงานของฟังก์ชันจะต้องส่งค่าคืนกลับไปยังโปรแกรมหรือฟังก์ชันที่เรียกใช้คำสั่งสำหรับการคืนค่า คือคำสั่ง return เช่น

```
return 0;
```

หมายถึงโปรแกรมจะส่งค่า 0 คืนไปยังฟังก์ชันที่เรียกใช้เมื่อทำงานเสร็จ