

# บทที่ 5 การเขียนโปรแกรมแบบวนซ้ำ

คท211 วิทยาการคอมพิวเตอร์

โดย อาจารย์ภาณุวัฒน์ เมฆะ

สาขาวิชาวิทยาการคอมพิวเตอร์  
คณะวิทยาศาสตร์ มหาวิทยาลัยแม่โจ้

# วัตถุประสงค์

- นักศึกษาเข้าใจกระบวนการทำงานแบบวนซ้ำ
- นักศึกษาสามารถเขียน โปรแกรมภาษาซี โดยใช้คำสั่งเพื่อให้คอมพิวเตอร์ทำงานแบบวนซ้ำได้

# เนื้อหาหลัก

- คำสั่งการวนซ้ำด้วย while
- คำสั่งการวนซ้ำด้วย do - while
- คำสั่งการวนซ้ำด้วย for

## 5.1 ทำไมต้องการเขียนโปรแกรมต้องมีการวนซ้ำ

- เหตุการณ์ที่เกิดขึ้นหลายรอบ  
เช่น โปรแกรมแสดง ชื่อ 20 ครั้ง
- เหตุการณ์ที่เกิดขึ้นหลายรอบ โดยมีการเปลี่ยนแปลงค่า หรือมีเงื่อนไข

เช่น แสดงผลเลข 0, 1, 2, ..., 10

แสดงผลรวมของ 1,3,5,7, ..., 99

แสดง ชื่อ ไปเรื่อย ๆ จนกว่าค่า X จะมากกว่า 30

# โปรแกรม 5.1 แสดงผลตัวเลข 0 - 10 | โดยไม่มีการวนซ้ำ

จงเขียนผังงานและโปรแกรมเพื่อแสดงตัวเลข 0 - 10 ออกทางหน้าจอ

- Output Analysis
  - แสดงผลเลข 0, 1, 2, ..., 10
- Input Analysis
  - ไม่มี

## โปรแกรม 5.1 แสดงผลตัวเลข 0 - 10 | โดยไม่มีการวนซ้ำ

- Process Analysis
  - โปรแกรมแสดงผลเลข 0, 1, 2, ... ,10
- Variable Define
  - ไม่ใช่ (หรือใช้ count เพื่อเพิ่มค่า)

# โปรแกรม 5.1 แสดงผลตัวเลข 0 - 10 | โดยไม่มีการวนซ้ำ

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

    printf ("0\t");
    printf ("1\t");
    printf ("2\t");
    printf ("3\t");
    printf ("4\t");
    ...
    ...
    printf ("10\t");
    return 0;
}
```

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

    int    count = 0;
    printf ("%d\t",count++);
    printf ("%d\t",count++);
    printf ("%d\t",count++);
    printf ("%d\t",count++);
    printf ("%d\t",count++);
    ...
    ...
    printf ("%d\t",count++);
    return 0;
}
```

# โปรแกรม 5.1 แสดงผลตัวเลข 0 - 10 | โดยใช้คำสั่งวนซ้ำ

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int    count = 0;
    while (count <= 10)
    {
        printf ("%d\t",count++);
    }

    return 0;
}
```



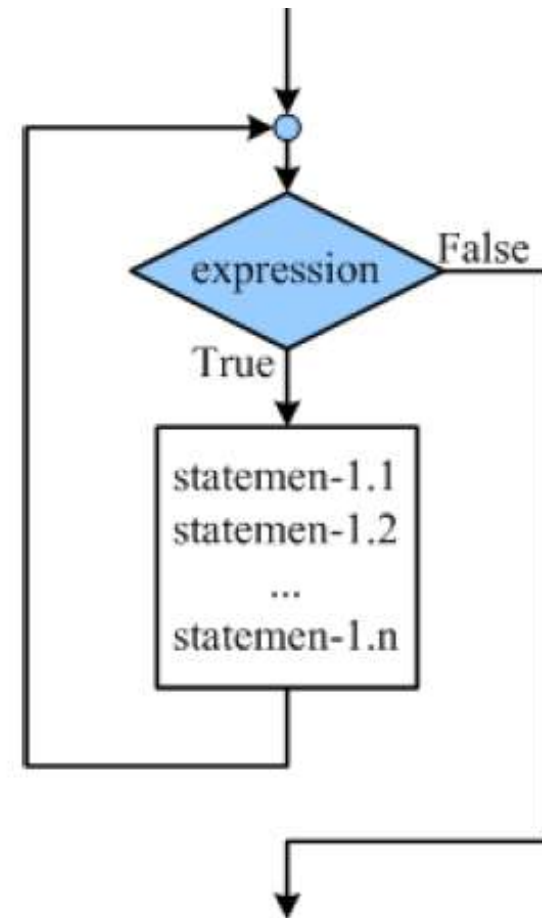
# คำสั่งสำหรับการวนซ้ำ (ภาษา ซี)

- **while**
- **do - while**
- **for**

## 5.2 คำสั่ง **while**

```
while (expression)  
  statement-1;
```

```
while (expression)  
{  
  statement-1.1;  
  statement-1.2;  
  ...  
  statement-1.n;  
}
```



## โปรแกรม 5.2 แสดงผลตัวเลข 0 - 10 | **while**

จงเขียนผังงานและ โปรแกรมที่มีการควบคุมทิศทางแบบวนรอบ โดยใช้คำสั่ง `while` เพื่อแสดงตัวเลข 0 - 10 ออกทางหน้าจอ

- Output Analysis
  - แสดงผลเลข 0, 1, 2, ..., 10
- Input Analysis
  - ไม่มี

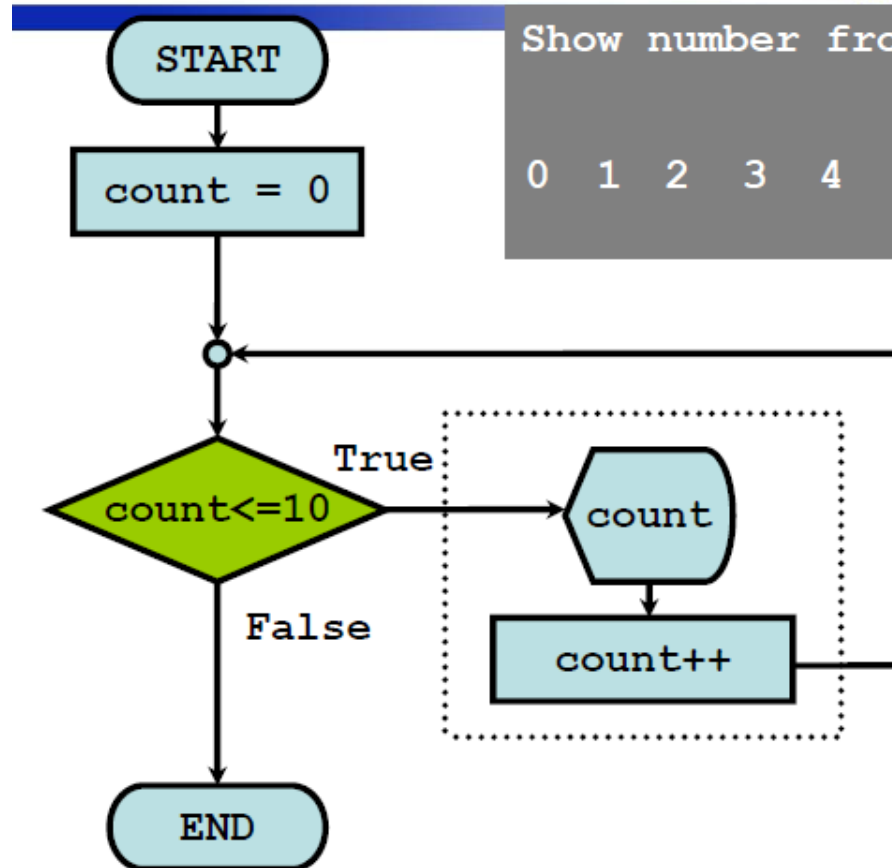
## โปรแกรม 5.2 แสดงผลตัวเลข 0 - 10 | **while**

- Process Analysis
  - โปรแกรมทำงานแบบวนรอบ เพื่อแสดงผลเลข 0, 1, 2, ... ,10
- Variable Define
  - count เป็นจำนวนเต็มเพื่อใช้นับจำนวนรอบ

```
count = 11
```

```
Show number from zero to ten
```

```
0 1 2 3 4 5 6 7 8 9 10
```



# โปรแกรม 5.2 แสดงผลตัวเลข 0 - 10 | while

```
#include<stdio.h>
int main()
{
    int    count = 0;
    printf ("Show number from zero to ten\n\n");
    while (count<=10)
    {
        printf ("%d\t",count);
        count++;
    }
    return 0;
}
```

Show number from zero to ten

0      1      2      3      4      5      6      7      8      9      10

## โปรแกรม5.3 แสดงผลตัวเลข 1 ถึงค่าที่กำหนด | **while**

จงเขียนผังงานและ โปรแกรมที่มีการควบคุมทิศทางแบบวนรอบ โดยใช้คำสั่ง `while` เพื่อให้โปรแกรมทำการบวกเลขจำนวนเต็ม ตั้งแต่ 1 จนถึงค่าที่ผู้ใช้งานกำหนด

- Output Analysis
  - ผลลัพธ์การบวกเลขจำนวนเต็ม ตั้งแต่ 1 ถึงค่าที่ผู้ใช้กำหนด
- Input Analysis
  - ค่าที่ผู้ใช้งานป้อนเข้ามา

# โปรแกรม 5.3 แสดงผลตัวเลข 1 ถึงค่าที่กำหนด | **while**

- Process Analysis

- โปรแกรมถามว่าผู้ใช้งานต้องการบวกเลขตั้งแต่ 1 ถึงเลขใด
- วนรอบแบบ while เพื่อบวกค่า
- แสดงผลลัพธ์ที่ได้

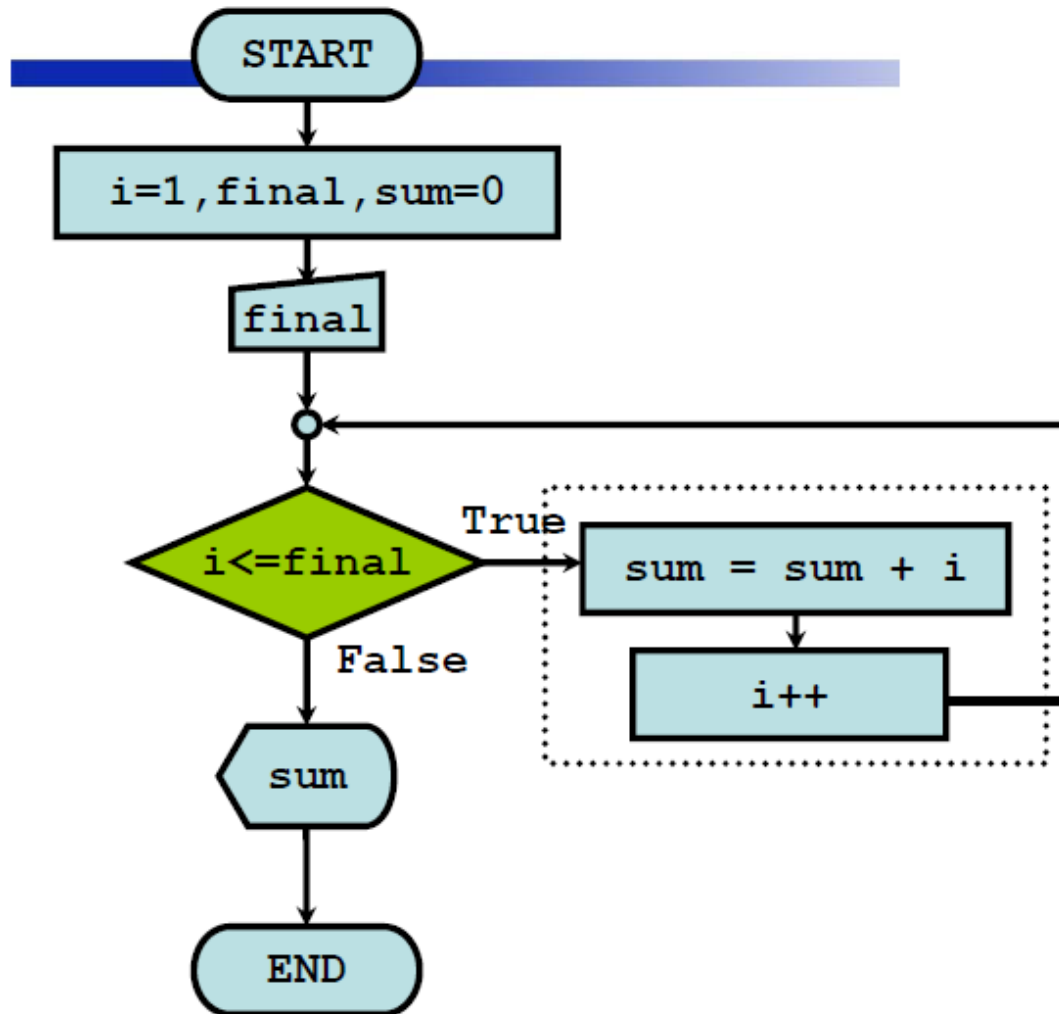
- Variable Define

sum = 0    ผลรวมของการบวก โดยเริ่มต้นมีค่าเท่ากับ 0

i = 1    ค่าที่นำเข้าไปบวกกับ sum ในแต่ละรอบ โดยรอบแรกค่า i  
มีค่าเท่ากับ 1 และมีค่าเพิ่มขึ้นรอบละ 1

final    เพื่อรับค่าจากผู้ใช้งาน และกำหนดจุดสิ้นสุดของค่า i





# โปรแกรมแสดงผลตัวเลข 1 ถึงค่าที่กำหนด | **while**

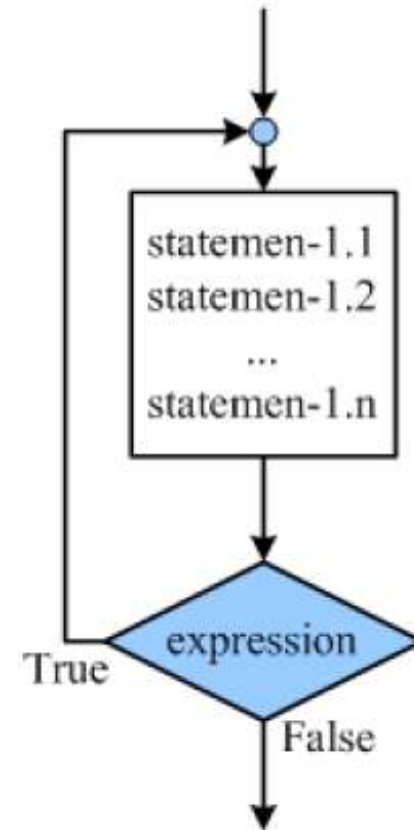
```
int main()
{
    int    i = 1, final, sum=0;
    printf ("Enter final number : ");
    scanf  ("%d",&final);
    while (i<=final)
    {
        sum = sum + i;
        i++;
    }
    printf  ("Sum = %d",sum);
    return 0;
}
```

- \* **หมายเหตุ** เอกสารจากส่วนนี้เป็นต้นไป โปรแกรมตัวอย่างใดที่ ขึ้นต้นด้วย `int main()` เลย หมายความว่าไม่มีพร็โพรเซสเซอร์ใดเรียกทีฟ แต่ไม่ได้เขียนแสดงไว้

## 5.3 คำสั่ง do-while

```
do  
    statement-1.1;  
while (expression);
```

```
do  
{  
    statement-1.1;  
    statement-1.2;  
    ...  
    statement-1.n;  
}  
while (expression);
```



## โปรแกรม 5.4 หาผลรวม 1 ถึง 100 | do-while

จงเขียนผังงานและ โปรแกรมสำหรับรวมเลขจำนวนเต็ม  
ตั้งแต่ 1 – 100 โดยใช้คำสั่ง do-while

- Output Analysis
  - ผลรวมของเลขจำนวนเต็ม ตั้งแต่ 1 - 100
- Input Analysis
  - ไม่มี

# โปรแกรม 5.4 หาผลรวม 1 ถึง 100 | do-while

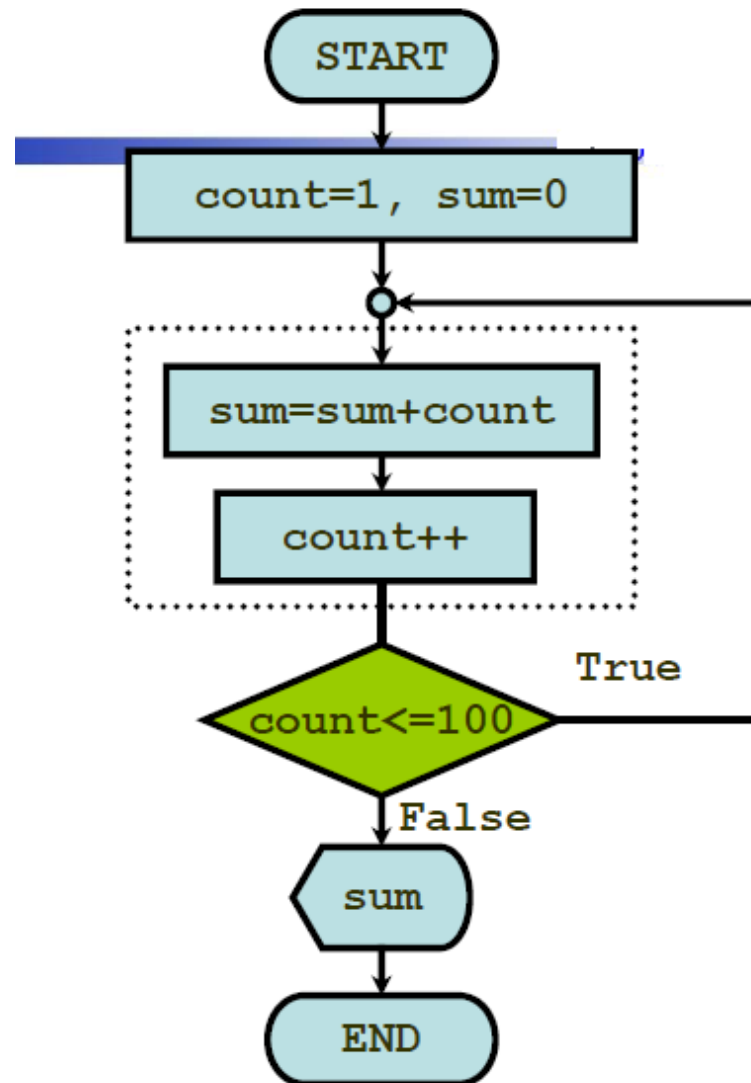
- Process Analysis

- โปรแกรมทำการบวกค่าเก็บไว้ในตัวแปรผลลัพธ์ แล้วเพิ่มค่าจนถึง 100

- Variable Define

count เป็นตัวแปรชนิดจำนวนเต็มเพื่อนับจำนวน

sum เป็นจำนวนเต็มเพื่อเก็บค่าผลรวม



## โปรแกรม 5.4 หาผลรวม 1 ถึง 100 | do-while

```
int main()
{
    int    count=1,sum=0;
    do
    {
        sum = sum + count;
        count++;
    }
    while(count<=100);
    printf ("Summation of 1 to 100 = %d",sum);
    return 0;
}
```

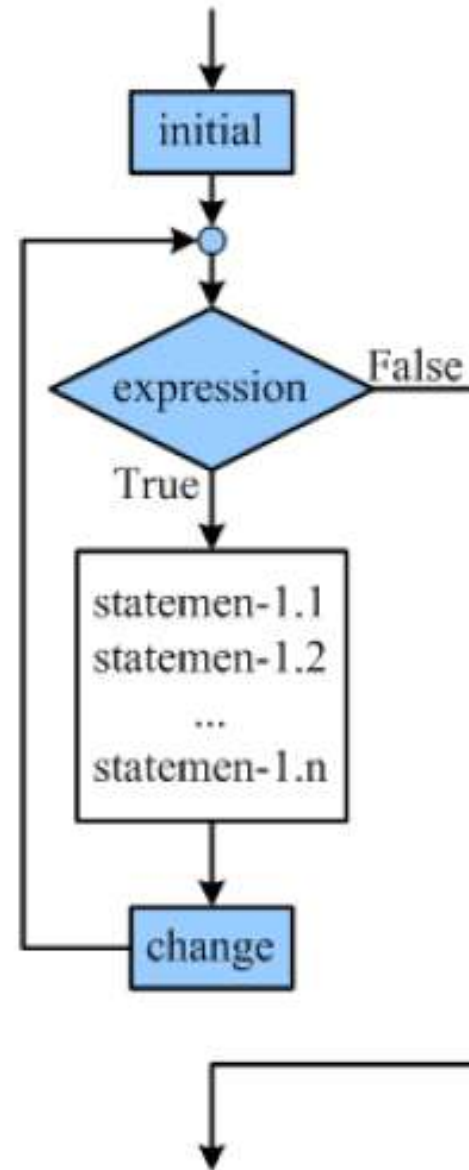
## 5.4 คำสั่ง **for**

```
for (initial; expression; change)  
{  
    statement-1.1;  
    statement-1.2;  
    ...  
    statement-1.n;  
}
```

- *initial* เป็นส่วนที่ใช้กำหนดค่าเริ่มต้นให้กับตัวแปร
- *condition* เป็นเงื่อนไขเพื่อพิจารณา
- *change* เป็นส่วนที่เปลี่ยนแปลงค่าตัวแปร
- *statement-1, 2, ... , n* เป็นคำสั่งที่จะทำงานเมื่อเงื่อนไขเป็นจริง

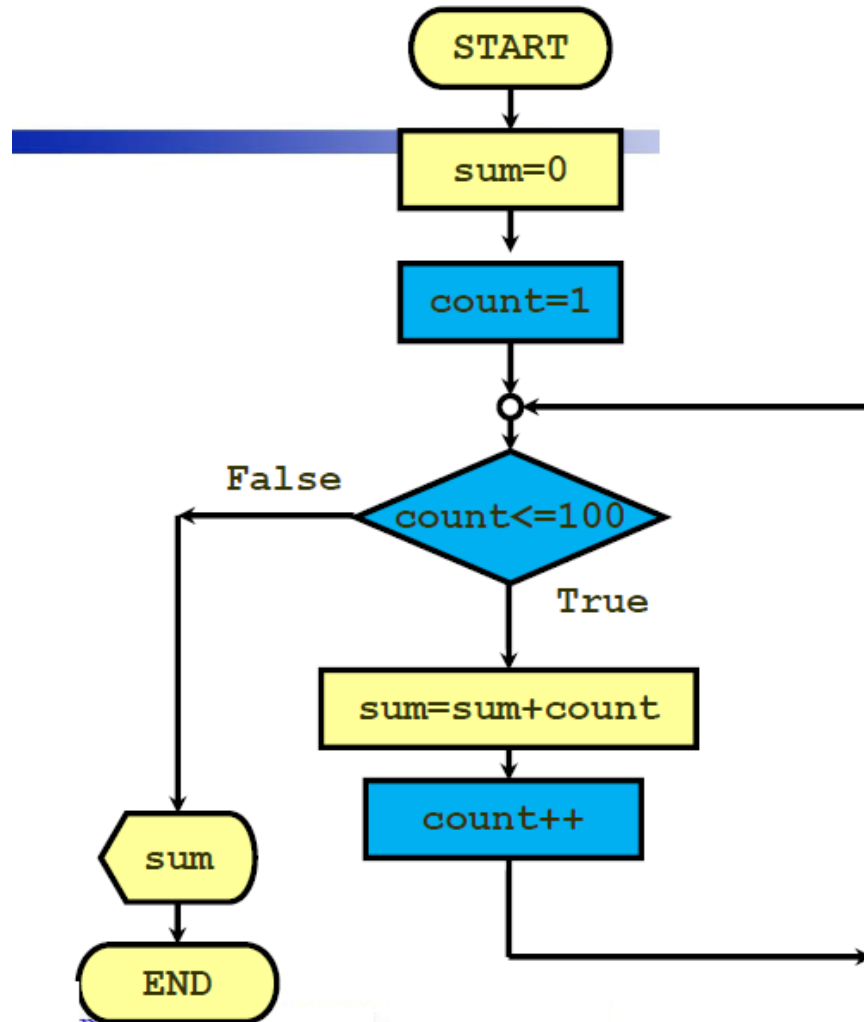


# รูปแบบการใช้คำสั่ง **for**



## โปรแกรม5.5 หาผลรวม 1 ถึง 100 | **for**

- Process Analysis
  - โปรแกรมทำการบวกค่าเก็บไว้ในตัวแปรผลลัพธ์ แล้วเพิ่มค่าจนถึง 100
- Variable Define
  - count เป็นตัวแปรชนิดจำนวนเต็มเพื่อบันทึกจำนวน
  - sum เป็นจำนวนเต็มเพื่อเก็บค่าผลรวม



## โปรแกรม5.5 หาผลรวม 1 ถึง 100 | for

```
#include<stdio.h>
int main()
{
    int  sum=0, count;
    for (count=1; count<=100; count++)
    {
        sum = sum + count ;
    }
    printf ("Summation of 1 to 100 = %d",sum) ;
    return 0;
}
```

## โปรแกรม 5.6 แสดงผล a - z | for

จงเขียนผังงานและ โปรแกรมสำหรับแสดงผลอักษร a - z ออกทางจอภาพ โดยใช้คำสั่ง for

- Output Analysis
  - แสดงผล a - z ทางจอภาพ

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
```

- Input Analysis
  - ไม่มี

# โปรแกรม 5.5 แสดงผล a - z | for

- Process Analysis
  - โปรแกรมทำการวนรอบเพื่อแสดงผลอักษรตั้งแต่ a – z โดยการเพิ่มค่าตัวแปรขึ้นครั้งละ 1 (ดูตาราง ASCII Code)
- Variable Define
  - letter เป็นตัวแปรชนิดอักขระ

## คำสั่งสำหรับการวนซ้ำ (ภาษา ซี)

- **while** ← **เช็คก่อน แล้วค่อยทำ**
- **do - while** ← **ทำก่อน แล้วค่อยเช็ค**
- **for** ← **มีการนับขึ้นหรือลง**

## 5.4 คำถามท้ายบท

1. จงเขียนโปรแกรมแสดงรหัสแอสกี ตั้งแต่ 33 ถึง 55

Decimal	ASCII
33	!
34	"
35	#
	. . .
	. . .
55	7



## คำถามท้ายบท

2. จงเขียนโปรแกรมรับตัวเลขเพื่อมาคำนวณหาผลบวกกำลังสอง  
จนกระทั่งตัวเลขที่รับเข้ามามีค่าเป็น 0

```
Enter a number : 2
```

```
Enter a number : -5
```

```
Enter a number : 0
```

```
Result : 29
```

### 3. ข้อใดเป็นโปรแกรมที่รันไม่รู้จบ (Infinite loop)

เมื่อกำหนด `int i=0;`

3.1 `for(i=0; i>0; i++) printf("%d",i);`

3.2 `for(i=0; i%2!=0; i += 2) puts("a");`

3.3 `while(i<7) printf("%d",i--);`

3.4 `do {  
    i+=3;  
} while(i%3==0);`



# ซาลามัด เชียง หมายถึง สวัสดิ์ในภาษาอินโดนีเซีย



Factorial,x-mas,is prime,reverse,palindrome