

# Computer Engineering

## วิศวกรรมคอมพิวเตอร์



## บทที่ 5 การเขียนโปรแกรมแบบวนซ้ำ

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

# วัตถุประสงค์

01006012 Computer Programming

- นักศึกษาเข้าใจกระบวนการทำงานแบบวนซ้ำ
- นักศึกษาสามารถเปลี่ยนโปรแกรมภาษาซีโดยใช้คำสั่งเพื่อให้คอมพิวเตอร์ทำงานแบบวนซ้ำได้

# เนื้อหาหลัก

01006012 Computer Programming

- คำสั่งการวนซ้ำด้วย while
- คำสั่งการวนซ้ำด้วย do - while
- คำสั่งการวนซ้ำด้วย for

# 5.1 ทำไมต้องการเขียนโปรแกรมต้องมีการวนซ้ำ

01006012 Computer Programming

- 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 จนกว่าตัวแปร X จะมีค่ามากกว่า 30

# โปรแกรม 5.1 แสดงผลตัวเลข 0 – 10 | โดยไม่มีการวนซ้ำ

01006012 Computer Programming

จงเขียนผังงานและโปรแกรมเพื่อแสดงตัวเลข 0 - 10 ออกทางหน้าจอ

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

# โปรแกรม 5.1 แสดงผลตัวเลข 0 – 10 | โดยไม่มีการวนซ้ำ

01006012 Computer Programming

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

# โปรแกรม 5.1 แสดงผลตัวเลข 0 – 10 | โดยไม่มีการวนซ้ำ

01006012 Computer Programming

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

# โปรแกรม 5.1 แสดงผลตัวเลข 0 – 10 โดยใช้คำสั่งวนซ้ำ

01006012 Computer Programming

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

int x = 0  
while (x ≤ 10)  
{ printf ("%d",x);  
 x++;  
}

# คำสั่งสำหรับการวนซ้ำ (ภาษา C++)

01006012 Computer Programming

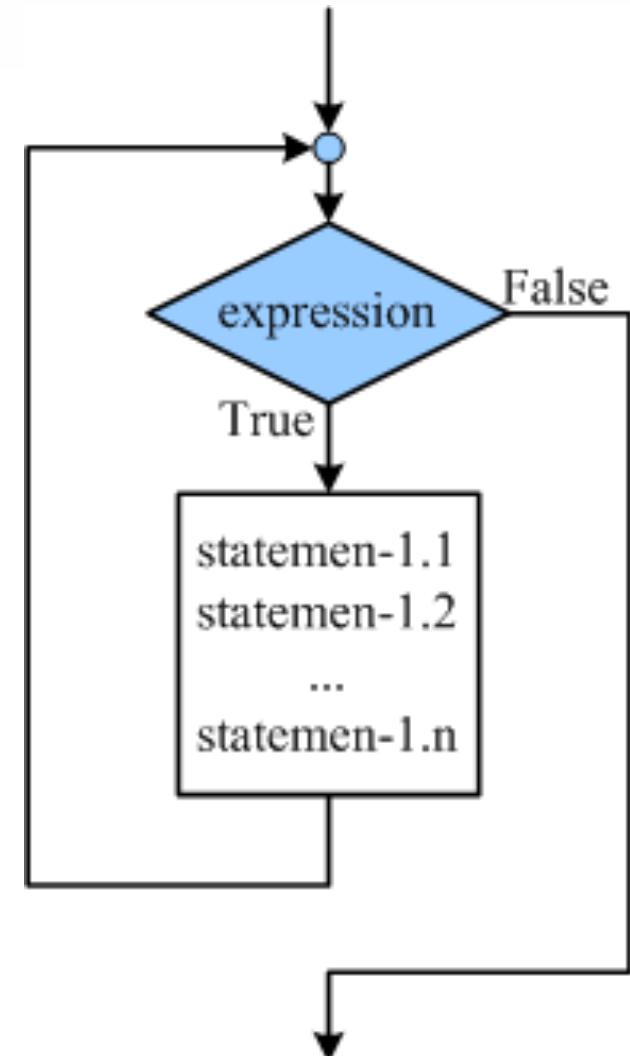
- **while**
- **do - while**
- **for**

## 5.2 คำสั่ง while

01001012 Principle of Computer Programming

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

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



# โปรแกรม 5.2 แสดงผลตัวเลข 0 – 10 | while

01006012 Computer Programming

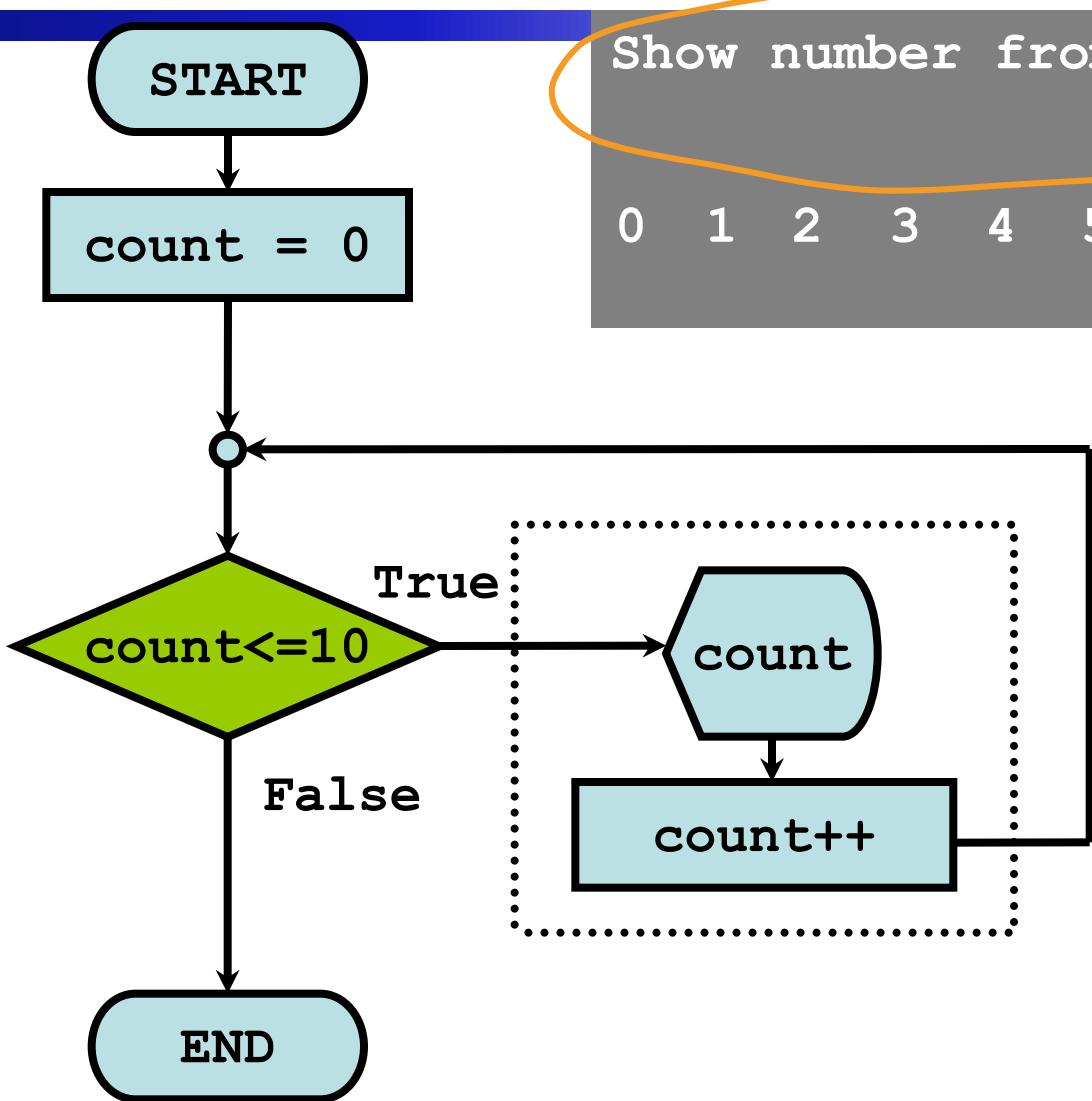
จงเขียนผังงานและ โปรแกรมที่มีการควบคุมทิศทางแบบวนรอบ โดยใช้คำสั่ง while เพื่อแสดงตัวเลข 0 - 10 ออกทางหน้าจอ

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

# โปรแกรม5.2 แสดงผลตัวเลข 0 – 10 | while

01006012 Computer Programming

- 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

```

int x = 0
while (x <= 10)
{
    printf ("%d\n", x);
    x++;
}
return 0;
  
```

# โปรแกรม5.2 แสดงผลตัวเลข 0 – 10 | while

01006012 Computer Programming

```
#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

01006012 Computer Programming

จะเขียนผังงานและโปรแกรมที่มีการควบคุมทิศทางแบบวนรอบโดยใช้คำสั่ง while เพื่อให้โปรแกรมทำการบวกเลขจำนวนเต็มตั้งแต่ 1 จนถึงค่าที่ผู้ใช้งานกำหนด

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

# โปรแกรม5.3 แสดงผลตัวเลข 1 ถึงค่าที่กำหนด | while

01006012 Computer Programming

- Process Analysis

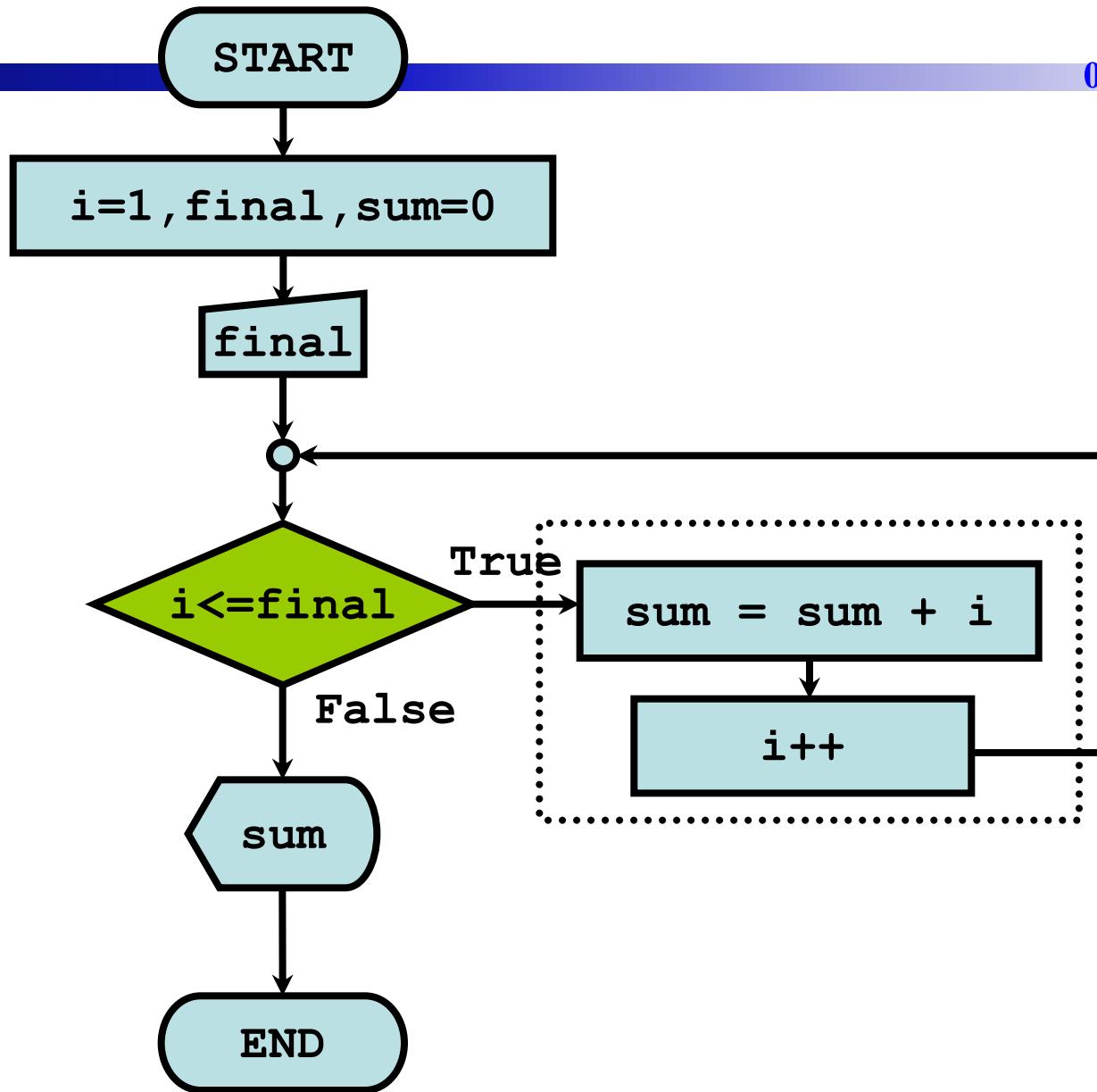
- โปรแกรมตามว่าผู้ใช้งานต้องการบวกเลขตั้งแต่ 1 ถึงเลขใด
- วนรอบแบบ while เพื่อบวกค่า
- แสดงผลลัพธ์ที่ได้

- Variable Define

sum = 0 ผลรวมของการบวก โดยเริ่มต้นมีค่าเท่ากับ 0

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

final เพื่อรับค่าจากผู้ใช้ และกำหนดจุดสิ้นสุดของค่า i



# โปรแกรมแสดงผลตัวเลข 1 ถึงค่าที่กำหนด | while

01006012 Computer Programming

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

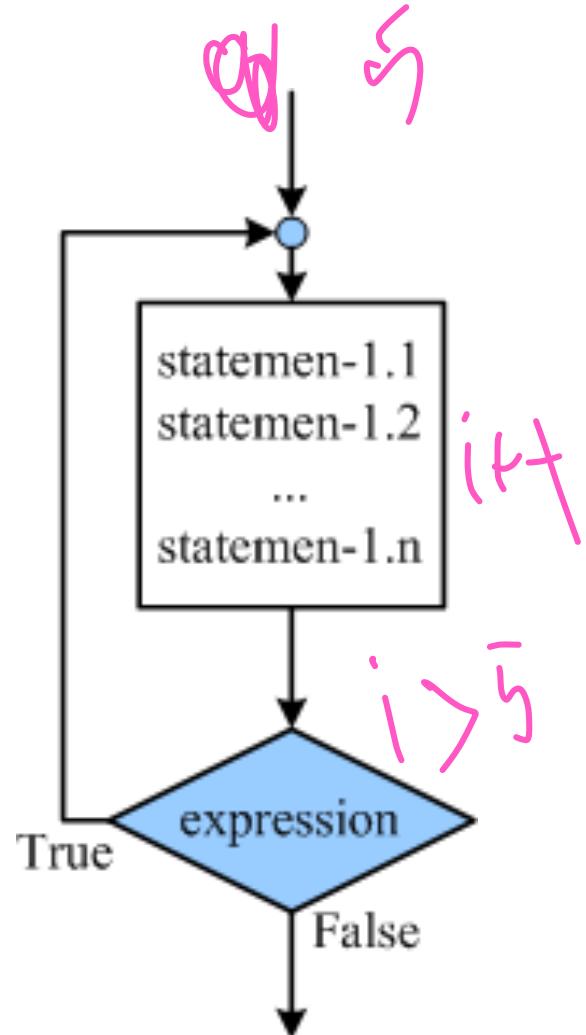
- \* **หมายเหตุ** เอกสารจากส่วนนี้เป็นต้นไป โปรแกรมตัวอย่างใดที่ ขึ้นต้นด้วย int main() เลย  
หมายความว่ามีพิธีประ塞สเซอร์ไดเร็คทีฟ แต่ไม่ได้เขียนแสดงไว้

## 5.3 คำสั่ง do-while

01006012 Computer Programming

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

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



## 5.3 คำสั่ง do-while

01006012 Computer Programming

เมื่อสั่งให้โปรแกรมด้านล่างทำงาน โปรแกรมจะแสดงค่าของ **i** ออกมากี่กันเท่าไร

```
#include <stdio.h>
int main()
{
    int i ;
    i = 0 ;
    do{
        printf("\nKMITL") ;
        i = i+1;
    } while (i > 5) ;
    printf("%d",i ) ;
    return 0 ;
}
```

1

# โปรแกรม 5.4 หาผลรวม 1 ถึง 100 | do-while

01006012 Computer Programming

จงเขียนผังงานและโปรแกรมสำหรับรวมเลขจำนวนเต็ม

ตั้งแต่ 1 – 100 โดยใช้คำสั่ง do-while

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

# โปรแกรม 5.4 หาผลรวม 1 ถึง 100 | do-while

01006012 Computer Programming

- Process Analysis

- โปรแกรมทำการบวกค่าเก็บไว้ในตัวแปรผลลัพธ์ แล้วเพิ่มค่าจนถึง 100

- Variable Define

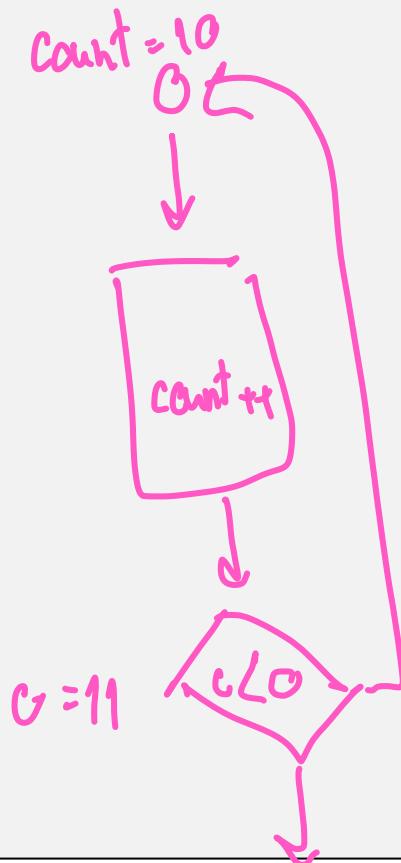
count เป็นตัวแปรชนิดจำนวนเต็มเพื่อนับจำนวน

sum เป็นจำนวนเต็มเพื่อเก็บค่าผลรวม

# ตัวอย่าง do-while

01006012 Computer Programming

```
int main()
{
    int count=10;
    do
    {
        printf("%d",count);
        count++;
    }
    while(count<0);
    return 0;
}
```

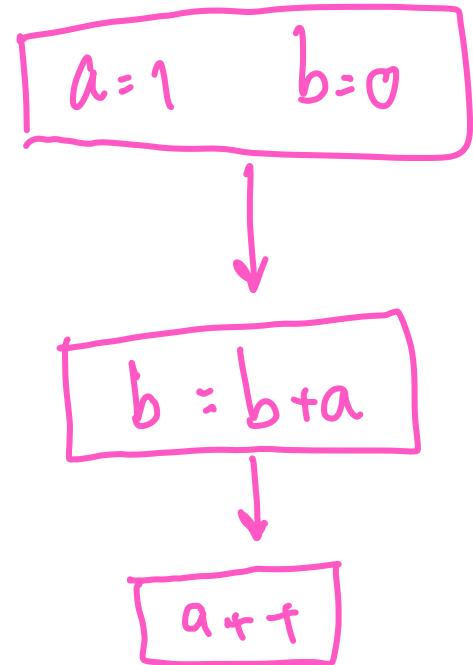
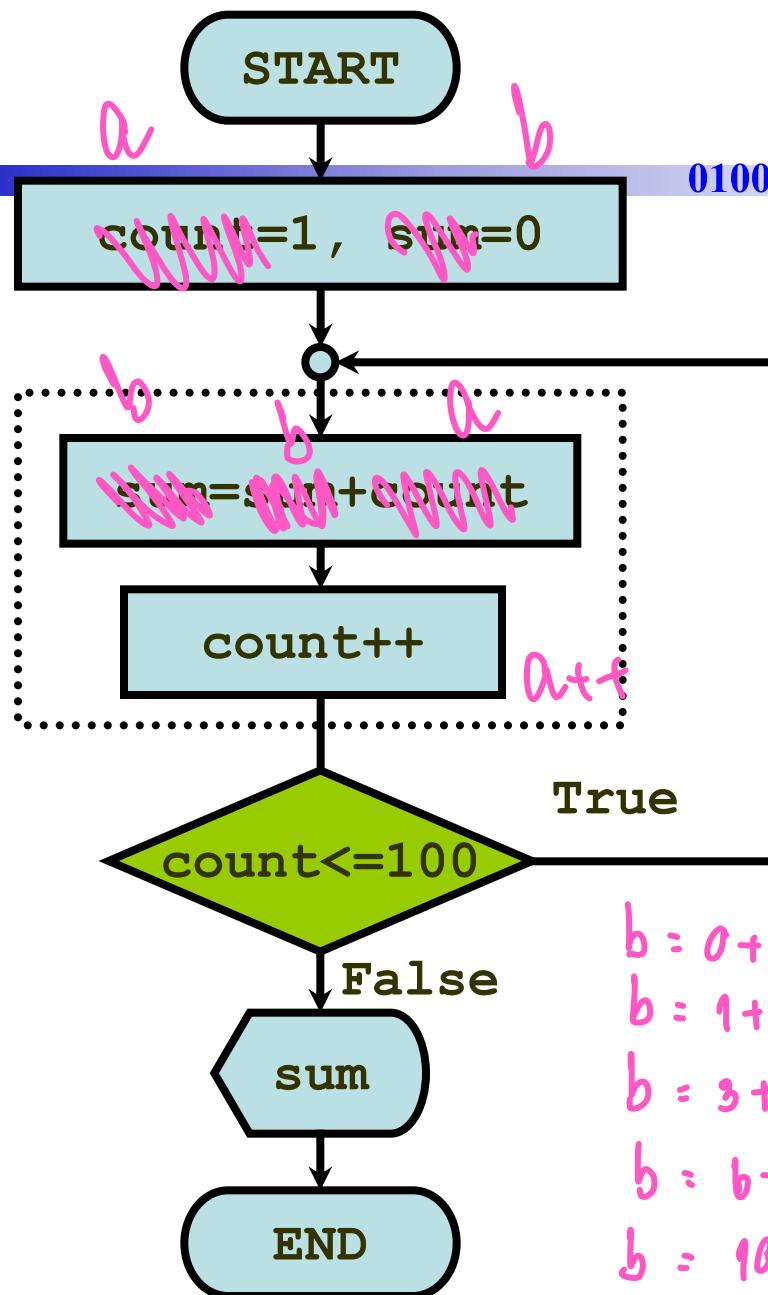


**Output**  
10



23

1  
2  
+  
3



$$\begin{aligned}
 b &= 0+1 = 1 & n &= (a_1+a_n) \\
 b &= 1+2 = 3 & & \frac{n}{2} \\
 b &= 3+3 = 6 & & \frac{5}{2}(1+3) = 15 \\
 b &= 6+4 = 10 & & \\
 b &= 10+5 = 15 & &
 \end{aligned}$$

# โปรแกรม5.4 หาผลรวม 1 ถึง 100 | do-while

01006012 Computer Programming

```
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

คำเริ่มต้น

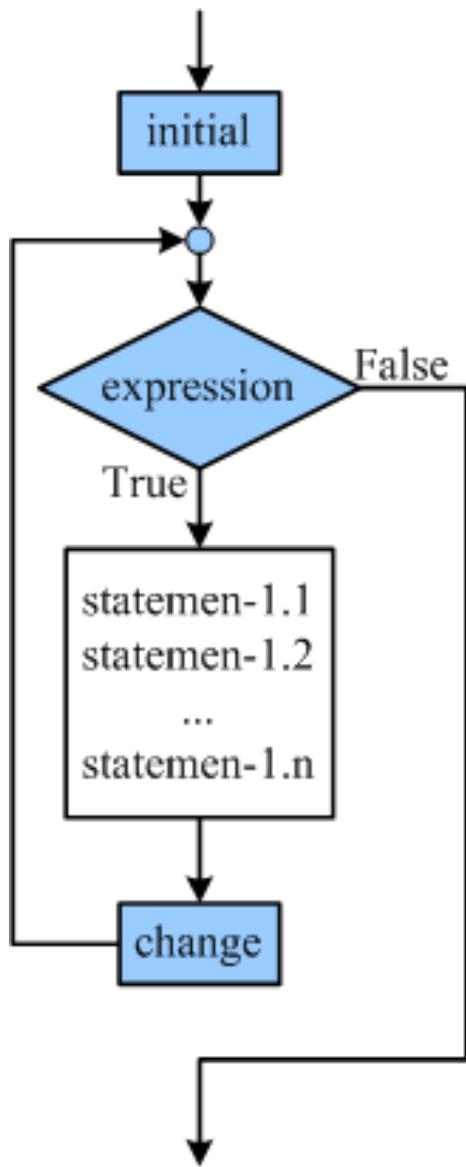
เงื่อนไข

```
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

01006012 Computer Programming



# โปรแกรม 5.5 หาผลรวม 1 ถึง 100 | for

01006012 Computer Programming

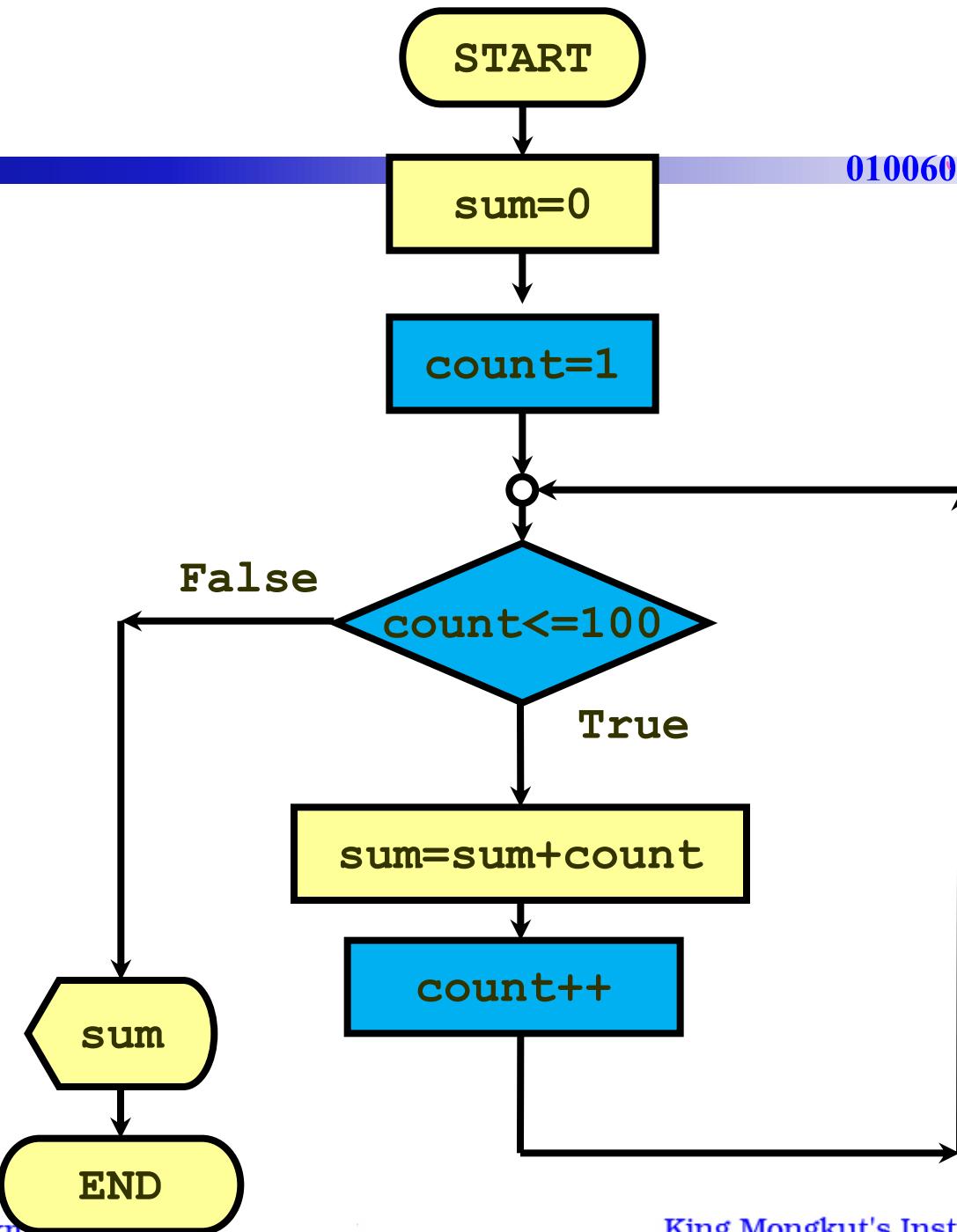
- Process Analysis

- โปรแกรมทำการบวกค่าเก็บไว้ในตัวแปรผลลัพธ์ แล้วเพิ่มค่าจนถึง 100

- Variable Define

count เป็นตัวแปรชนิดจำนวนเต็มเพื่อนับจำนวน

sum เป็นจำนวนเต็มเพื่อเก็บค่าผลรวม



# โปรแกรม5.5 หาผลรวม 1 ถึง 100 | for

01006012 Computer Programming

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

# โปรแกรม5.6 แสดงผล a – z | for

01006012 Computer Programming

จงเขียนผังงานและโปรแกรมสำหรับแสดงผลอักษร a – z ออกทางจอภาพ โดยใช้คำสั่ง for

- Output Analysis
  - แสดงผล a – z ทางจอภาพ

```
int a = 97  
for (a=97; a <= 122, a++)  
{ printf("%c ", a); }
```

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

01006012 Computer Programming

- Process Analysis

- โปรแกรมทำการวนรอบเพื่อแสดงผลอักษรตั้งแต่ a – z โดยการเพิ่มค่าตัวแปรขึ้นครั้งละ 1 (ดูตาราง ASCII Code)

- Variable Define

letter เป็นตัวแปรชนิดอักขระ

## 5.4 คำถ้ามท้ายบท

01006012 Computer Programming

1. จงเขียนโปรแกรมแสดงรหัสแอสกี ตั้งแต่ 33 ถึง 55

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

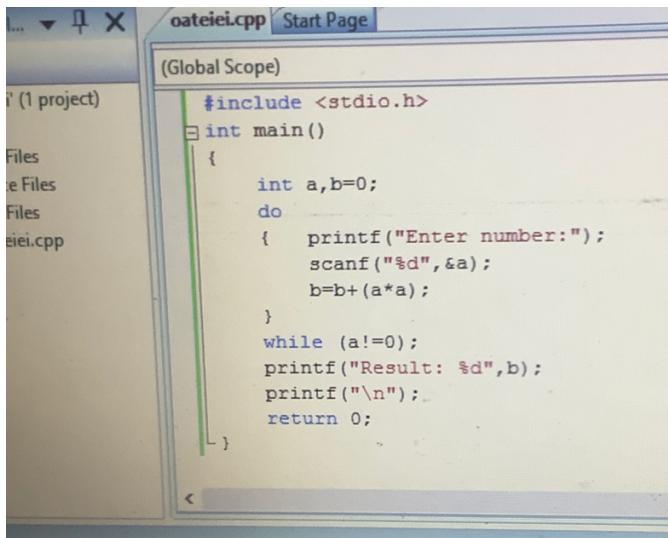
2. จงเขียนโปรแกรมรับตัวเลขเพื่อมาคำนวณหาผลบวกกำลังสอง  
จนกระทั่งตัวเลขที่รับเข้ามามีค่าเป็น 0

Enter a number : 2

Enter a number : -5

Enter a number : 0

Result : 29



```
#include <stdio.h>
int main()
{
    int a,b=0;
    do
    {
        printf("Enter number:");
        scanf("%d",&a);
        b=b+(a*a);
    }
    while (a!=0);
    printf("Result: %d",b);
    printf("\n");
    return 0;
}
```

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 {**  
*0+3 = 3+3 = 6+3*  
**i+=3;**

**}** **while (i%3==0) ;**