

ครั้งที่ 4

คำสั่งการทำงานซ้ำและทำงานวนรอบ

โครงสร้างการปฏิบัติการ

- sequence
- selection
- repetition

คำสั่ง **While**

- counter-controlled while loop
- sentinel-controlled while loop
- flag-controlled while loop
- eof-controlled while loop

กรณีที่ **1** :

Counter-controlled while loop

รูปแบบทั่วไปดังนี้

set loop counter variable to an initial value

While loop control variable < final value

...

Increase loop control variable by 1

while Statement

รูปแบบ: while (loop repetition condition)
 statement ;

ตัวอย่าง: // display n asterisks

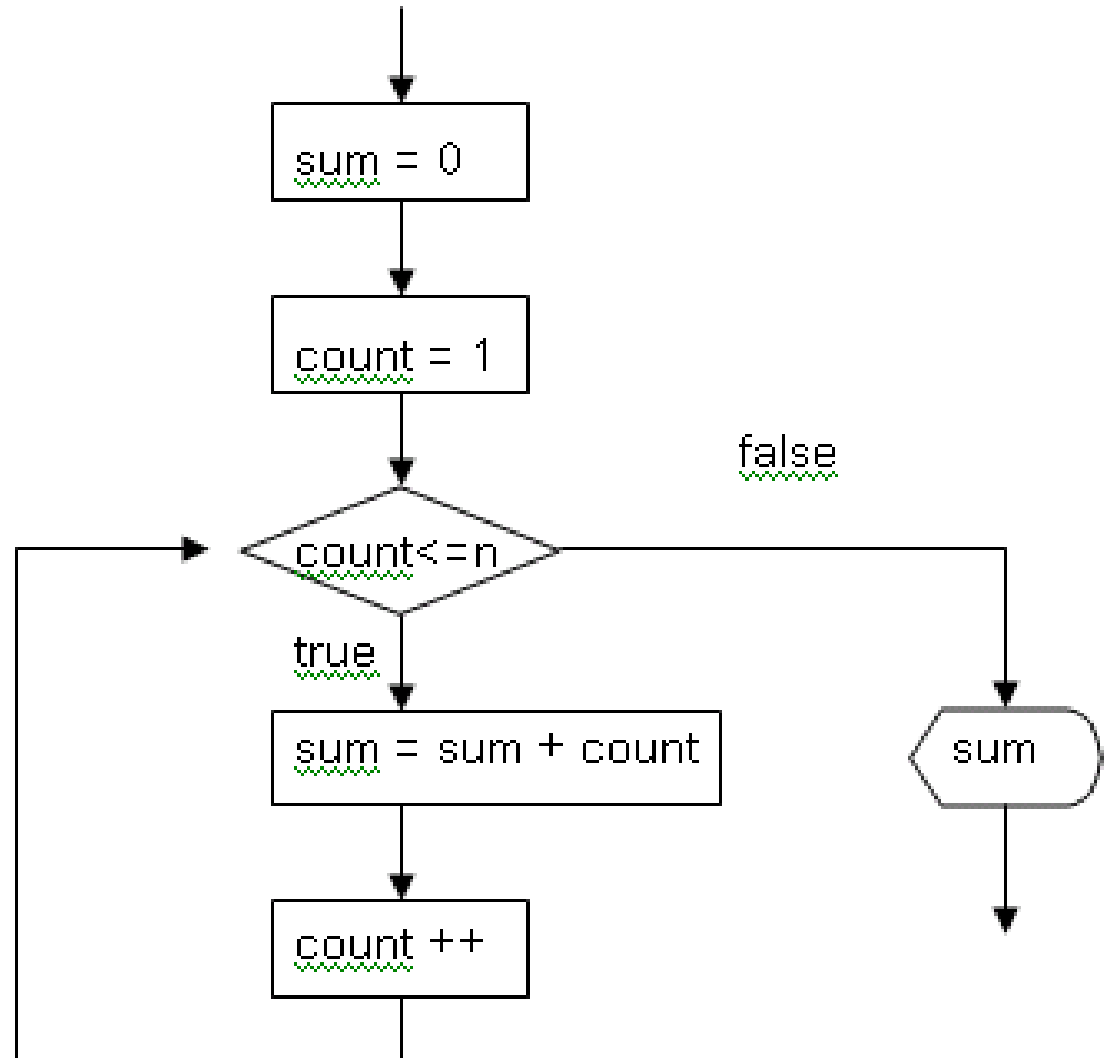
 countStar = 0 ;
 while (countStar < n)
 {
 cout << "*" ;
 countStar = countStar + 1 ;
 }

ความหมาย : countStar เป็นตัวนับการทำงาน โดยถ้า n มีค่าเท่ากับ 5
การทำงานจะวนรอบทั้งหมด 5 รอบโดยพิมพ์ "*****"
ออกทางจอภาพ และจะหยุดการทำงานเมื่อเงื่อนไขเป็น false

ตัวอย่าง การหาผลรวมของ $1 + 2 + \dots + n$

```
sum = 0 ;   count = 1 ;  
while (count <= n)  
    {  
        sum = sum + count ;  
        count ++ ;  
    }  
cout << sum << endl
```

แสดงการทำงานเป็นผังโปรแกรม ดังนี้



ตัวอย่าง โปรแกรมการหาค่าเฉลี่ย

```
sum = 0 ;
counter = 0 ;
while (counter < limit)
{
    cin >> number ;
    sum = sum + number ;
    counter++ ;
}
cout << " sum of the " << limit << "numbers = " << sum << endl
;
if (counter != 0)
    cout << "The average = " << sum / counter << endl ;
else
    cout << "No input." << endl ;
```


กรณีที่ 2 : sentinel-controlled while loop

รูปแบบทั่วไป

```
cin >> variable ;  
    while (variable != sentinel)  
{  
    ...  
    cin >> variable ;  
}
```

ตัวอย่าง โปรแกรมการหาค่าเฉลี่ย

```
const int SENTINEL = -999 ;
int main()
{
    int number ;           // variable to store the number
    int sum = 0;           // variable to store the sum
    int count = 0;         // variable to store the total
    cout << "Enter numbers ending with " << sentinel << endl ;
    cin >> number ;
    while ( number != SENTINEL)
    {
        sum = sum + number ;
        count ++ ;
        cin >> number ;
    }
    cout << "sum of the " << count << "numbers is " << sum << endl;
    if (count != 0 )      cout << "The average is " << sum/count << endl ;
    else                  cout << "No input" << endl ;
    return 0;
}
```

กรณีที่ 3 :

flag-controlled while loop

รูปแบบการทำงานดังนี้

```
found = false ; // initialize the loop control variable
while (!found) // test the loop control variable
{
    ...
    if (expression)
        found = true ;    // re-initialize the loop control variable
    ...
}
```

ตัวอย่าง ฟังก์ชันในการค้นหาอักขระที่เป็นตัวเลข

```
char getDigit()
{
    char nextChar ;           // user input – next data character
    bool digitRead ;          // status flag – set true
                                // when digit character is read
    digitRead = false;        // no digit character read yet
    while (!digitRead)
    {
        cin >> nextChar ;
        digitRead = (('0' <= nextChar) && (nextChar <= '9'));
    }
    return nextChar ;
}
```

ตัวอย่าง FIBONACCI NUMBER

การหาค่าจำนวน fibonacci นั้นมีลำดับของจำนวนดังนี้

1 , 1 , 2 , 3 , 5 , 8 , 13 , 21 , 34 , ...

ถ้าให้จำนวน 2 ตัวแรกของลำดับมีค่าเท่ากับ a_1 และ a_2 ค่าของลำดับต่อมาจะมีค่าเท่ากับ

$$a_3 = a_1 + a_2$$

ดังนั้น ถ้าให้ a_n โดย $n \geq 3$ แล้ว

$$a_n = a_{n-1} + a_{n-2}$$

$$a_3 = a_2 + a_1$$

$$= 1 + 1$$

$$= 2$$

$$a_4 = a_3 + a_2$$

$$= 2 + 1$$

$$= 3$$

คำสั่ง **for**

for Statement

รูปแบบ: for (initialization expression ;
 loop repetition condition ;
 update expression)
 statement ;

ตัวอย่าง การพิมพ์ '*' จำนวน **n** ครั้ง

```
// Display N asterisks.  
for (countStar = 0 ; countStar < n ; countStar ++)  
    cout << '*' ;
```

ตัวอย่าง

การหาค่าที่มากที่สุดของกลุ่มตัวเลข **10** จำนวน

```
double larger(double x , double y )  
{  
    if ( x >= y )  
        return x ;  
    else  
        return y ;  
}
```



```
# include <iostream>
using namespace std;
double larger(double x , double y) ;
int main()
{
    double num ; // variable to hold the current number
    double max ; // variable to hold the larger number
    int count ; // loop control variable

    cout << "Enter 10 numbers. " << endl ;
    cin >> num ;
    max = num ;
    for (count = 1 ; count < 10 ; count++)
    {
        cin >> num ;
        max = larger(max , num ) ;
    }
    cout << "The larger number is " << max << endl ;
    return 0 ;
} // end main
```

ตัวอย่าง ฟังก์ชันหาค่าแฟกทอเรียล

```
// Computes factorial (n!)
// Pre: n is greater than or equal to zero
int factorial(int n)
{
    int product ;           // accumulator for product computation
    product = 1;
    // Computes the product n x (n-1) x (n-2) x ... x 2
    for (int i = n ; i > 1 ; i - - )
        product = product * i ;
    // Returns function result
    return product ;
}
```

แบบฝึกหัด 1

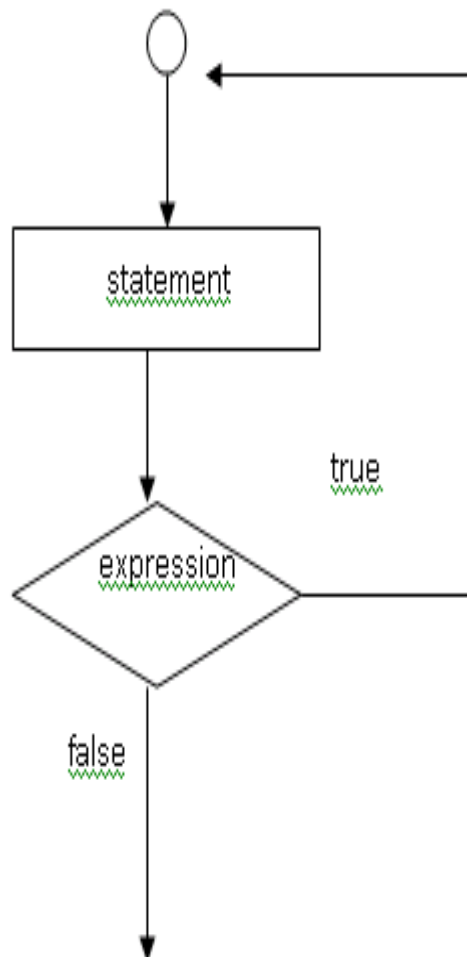
- จงเขียนโปรแกรมหาค่า **Permutation (n, r)** และ **Combination (n, r)**

ตัวอย่าง การแบ่งกลุ่มเลขจำนวนเต็ม

- โปรแกรมนี้จะให้ผู้เข้าป้อนกลุ่มของเลขจำนวนเต็มใดๆ 20 จำนวนทางแป้นพิมพ์ นำข้อมูลต่างๆเหล่านั้นมาแบ่งกลุ่มเพื่อนับจำนวนว่ามีเลขจำนวนเต็มคู่(even) เลขจำนวนเต็มคี่(odd) และเลขศูนย์(zero) อยู่อย่างละกี่จำนวน

คำสั่ง **do..while**

- เป็นคำสั่งทำงานซ้ำโดยมีการตรวจสอบเงื่อนไขการทำงานหลังจากปฏิบัติงาน ซึ่งการทำงานในลักษณะนี้จะมีการทำงานอย่างน้อยก่อน **1** รอบ
- เหมาะสำหรับใช้แก้ปัญหาที่ต้องการตรวจสอบค่าของข้อมูลเข้า **(input)** ว่าอยู่ในขอบเขตที่ต้องการหรือไม่ ถ้าผู้ใช้ป้อนข้อมูลผิด โปรแกรมจะย้อนกลับให้ป้อนข้อมูลใหม่ที่ถูกต้อง หรืออาจเป็นเมนูเลือกการทำงาน หรือสารสนเทศเพื่อถามผู้ใช้ว่าต้องการทำซ้ำหรือไม่เป็นต้น



do-while Statement

รูปแบบ: do
statement ;
while (expression) ;

ตัวอย่าง ฟังก์ชันในการตรวจสอบค่า

```
int getIntRange(int min , int max)    // range boundaries
{
    int nextInt ;                      // next number read
    // enter data until a number between min and max is read
    do
    {
        cout << "Enter a number between " << min << " and " << max << " : " ;
        cin >> nextInt ;
    } while ((nextInt < min) || (nextInt > max)) ;
    return nextInt ;
}
```

การเรียกใช้ Value = getIntRange(1 , 100) ;

ตัวอย่าง โปรแกรมเลือกการทำงาน

do

Display the menu

Read the user's choice

Perform the user's choice

Display activity

while the choice is not Quit

งาน

- จงเขียนโปรแกรมในการหาค่าของพื้นที่ตามเมนูดังนี้

Menu

1. Triangle
 2. Ractangular
 3. Circle
- select 1/2/3 ?

Nested Loops

- การแก้ปัญหาในบางกรณีสามารถใช้ คำสั่งกระทำซ้ำซ้อนๆกันได้ โดยการทำงานนั้นจะกระทำเริ่มจาก ลูปนอกก่อน และ จะกระทำลูปใน

ตัวอย่าง โปรแกรมพิมพ์ตารางผลคูณ

```
// Display table , row-by-row
for (int rowVal = 0 ; rowVal < 10 ; rowVal++)
{
    cout << setw(3) << rowVal << '|';
    for ( int colVal = 0 ; colVal < 10 ; colVal++)
        cout << setw(3) << rowVal * colVal ;
    cout << endl ;
}
```

0	0	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>
1	0	1	2	3	4	5	6	7	8	9
2	0	2	4	6	<u>8</u>	<u>10</u>	12	14	16	18
3	0	3	6	9	<u>12</u>	<u>15</u>	18	21	24	27
4	0	4	8	<u>12</u>	<u>16</u>	20	24	28	32	36
5	0	5	<u>10</u>	<u>15</u>	20	25	30	35	40	45
6	0	6	<u>12</u>	<u>18</u>	24	30	36	42	48	54
7	0	7	<u>14</u>	<u>21</u>	28	35	42	49	56	63
8	0	8	<u>16</u>	<u>24</u>	32	40	48	56	64	72
9	0	9	<u>18</u>	<u>27</u>	36	45	54	63	72	81

ตัวอย่าง การพิมพ์รูปที่กำหนด

ต้องการพิมพ์รูปดังนี้

ทางจอภาพ

*

* *

* * *

* * * *

* * * * *

```
int main ()
{
    int i , j ;
    for (i = 1 ; i<= 5 ; i++)
    {
        for (j=1 ; j <= i ; j++)
            cout << "*" ;
        cout << endl ;
    }

    return 0 ;
}
```

แบบฝึกหัด 1

- จงเขียนโปรแกรมรับ ความยาวฐาน และส่วนสูง ของสามเหลี่ยมหนึ่ง จงพิมพ์ ความยาวฐาน ส่วนสูง และพื้นที่ของสามเหลี่ยม โดยเพิ่มความยาวฐานทีละ 1 และเพิ่มส่วนสูงครั้งละ 0.5 จนกระทั่งพื้นที่ของสามเหลี่ยมรูปใหม่มีพื้นที่มากกว่าพื้นที่ของสามเหลี่ยมรูปแรกเป็น 3 เท่า

แบบฝึกหัดที่ 2

- จงเขียนโปรแกรมโดยใช้คำสั่งวนรอบเพื่อพิมพ์ดังต่อไปนี้

1

2 3

4 5 6

7 8 9 10

11 12 13 15 16

แบบฝึกหัดที่ 3

- ร้านขายส่งผลไม้แห่งหนึ่งมีผลไม้ขาย 3 ชนิดคือ ส้ม มะม่วง ทุเรียน โดย ขายส้มโลละ 10 บาท มะม่วงโลละ 8 บาท ทุเรียนโลละ 30 บาท

ถ้าลูกค้าซื้อสินค้าทั้งหมดเกิน 5000-20000 บาทได้ส่วนลด 5 % แต่ถ้าซื้อสินค้าเกิน 20000 บาทได้ส่วนลด 10%

จงเขียนโปรแกรมคิดราคาสินค้า เมื่อ **run** โปรแกรม ผลการทำงานเป็นดังนี้

INPUT : Customer 1

Orange = ? 2

Manglo = ? 5

Durain = ? 1

OUTPUT :

amount = 90

discount = 0

net = 90

โปรแกรมจะถามผู้ใช่ว่ามีลูกค้าคนต่อไปหรือไม่

RUN AGAIN? Y/N

ถ้าผู้ใช้ตอบ Y จะกระทำซ้ำให้ป้อนลูกค้าคนต่อไป

INPUT : Customer 2

Orange = ? 200

Manglo = ? 500

Durain = ? 100

OUTPUT :

amount = 9000

discount = 450

net = 8550

โปรแกรมจะถามผู้ใช่ว่ามีลูกค้าคนต่อไปหรือไม่

RUN AGAIN? Y/N

ถ้าผู้ใช้ตอบ **N** จะหยุดการทำงานและสรุปรายได้ประจำวันดังนี้

Total amount = 9090

Total discount = 450

Total net = 8640