

344-111 ชุดวิชาการโปรแกรมและขั้นตอนวิธี

คำสั่งควบคุม



Outline

- คำสั่ง goto
- คำสั่งควบคุมแบบวนซ้ำ
 - คำสั่ง while
 - คำสั่ง do...while
 - คำสั่ง for
 - คำสั่ง break
 - คำสั่ง continue



คำสั่งควบคุม Control statements

- โปรแกรมภาษาซี จะเริ่มทำงานจากฟังก์ชัน main() โดยมีเครื่องหมาย { เป็นจุดเริ่มต้นการทำงานของฟังก์ชัน
- คำสั่งต่างๆ จะถูกประมวลผลเรียงตามลำดับคำสั่งก่อนหลังตามที่เราเขียนไปจนถึงจุดสิ้นสุดของการทำงานของโปรแกรมที่เราเขียนขึ้นมา ซึ่งจบด้วย เครื่องหมาย }
- หากเราต้องการเปลี่ยนแปลงขั้นตอนการทำงาน โดยอาจมีการกระโดดข้ามไปทำคำสั่งอื่นที่กำหนด หรือ ใหวนกลับมาทำคำสั่งเดิมๆ หลายๆ รอบ แต่ละรอบอาจให้ผลการทำงานที่ต่างกัน ซึ่งลักษณะการทำงานดังกล่าวนี้ สามารถกระทำได้ โดยจะต้องใช้คำสั่งควบคุมเพื่อเปลี่ยนแปลงลำดับขั้นตอนการทำงานแบ่งได้ เป็น 3 ประเภท
 - คำสั่งควบคุมให้ไปทำงานโดยไม่มีเงื่อนไข
 - goto
 - คำสั่งควบคุมให้ไปทำงานโดยมีเงื่อนไข
 - If
 - switch
 - คำสั่งควบคุมให้ไปทำงานโดยทำงานเป็นวงจร
 - While
 - do-while
 - for



คำสั่ง goto: ทำงานโดยไม่มีเงื่อนไข

- เป็นการสั่งให้ข้ามไปทำงานยังตำแหน่งที่มีชื่อ (label) ตามที่ระบุ โดยไม่มีเงื่อนไข
- รูปแบบ `goto statement-label;`
statement-label หมายถึง ชื่อของตำแหน่ง (label) ที่ต้องการให้ไปทำงาน

ตัวอย่าง:

```
int main ()
{
    float r = 1;
    float area;
    loop1: area = 22.0/7.0*r*r;
    printf(" R = %.2f Area = %.4f \n", r, area);
    r++;
    goto loop1;
    return (0);
}
```



คำสั่งควบคุมแบบวนซ้ำ (Loop)

- ถ้าเราต้องการวนซ้ำเพื่อคำนวณภาษีของลูกค้าหลายคน โดยไม่ต้องเปิดโปรแกรมขึ้นใหม่ จะทำอย่างไร ???

```
Enter name : Niwan  
Enter type : 2  
Enter value : 3500  
    Name = Niwan  
    TYPE = 2  
    VALUE = 3500.00  
    TAX = 525.00
```

```
Do you want to continue ? : y
```

```
Enter name : Mali  
Enter type : 1  
Enter value : 1500  
    TYPE = 1  
    VALUE = 1500.00  
    TAX = 150.00
```

```
Do you want to continue ? : N
```



คำสั่งควบคุมแบบวนซ้ำ (Loop)

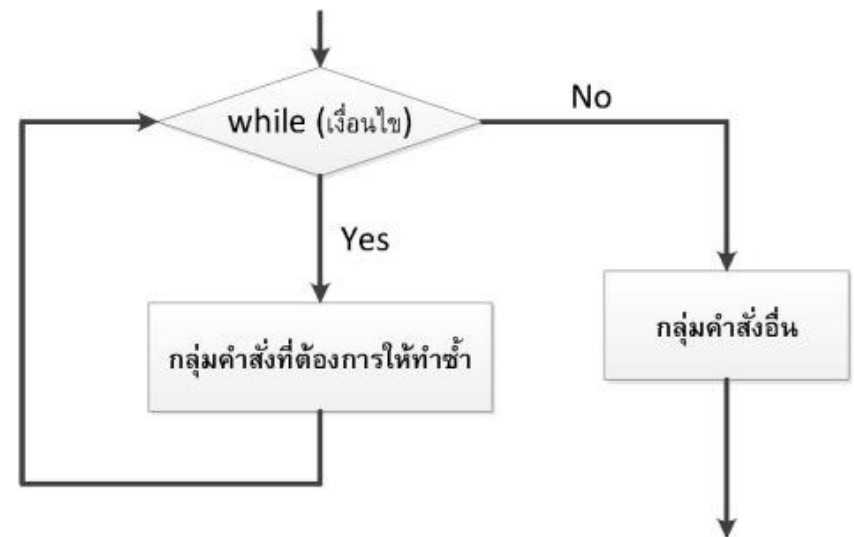
- ต้องมีความรู้เกี่ยวกับคำสั่งควบคุมแบบวนซ้ำ (loop) มี 3 ลักษณะ
 - แบบที่ 1 ตรวจสอบเงื่อนไขก่อนการวนซ้ำ (while)
 - แบบที่ 2 ทำงาน 1 รอบ แล้วจึงตรวจสอบเงื่อนไขเพื่อวนซ้ำ (do ... while)
 - แบบที่ 3 ทำงานวนซ้ำโดยการกำหนดจำนวนรอบของการทำงาน (for)



while: ตรวจสอบเงื่อนไขก่อนการวนซ้ำ

รูปแบบ

```
while ( เงื่อนไข )  
{  
    statement (s) ;  
}
```





ตัวอย่าง

ตัวอย่าง

```
x = 1;  
while (x <= 10)  
{  
    printf("%d\n", x);  
    x++;  
}
```

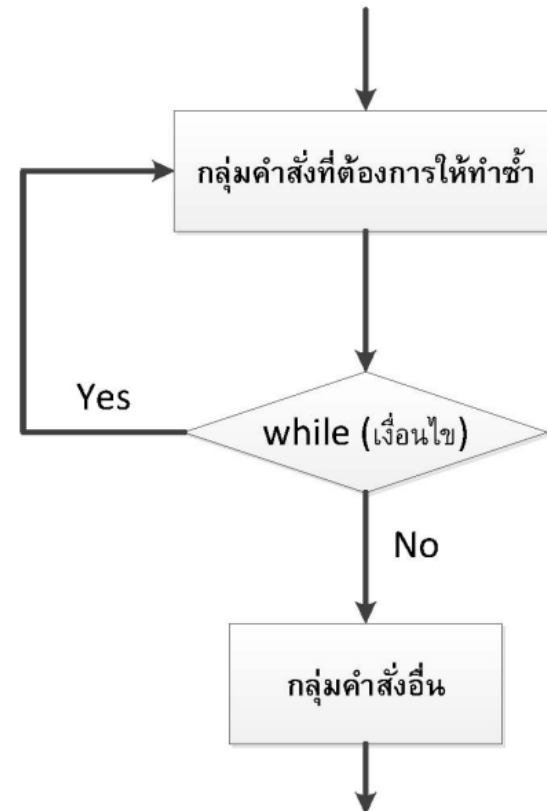
ผลลัพธ์ ????

do..while

- ทำงาน 1 รอบ แล้วจึงตรวจสอบเงื่อนไขเพื่อวนซ้ำ

รูปแบบ

```
do  
{  
    statement (s) ;  
} while ( เงื่อนไข );
```





while vs. do...while

รูปแบบ :

```
while ( เงื่อนไข )  
{  
    statement (s) ;  
}
```

ตัวอย่าง :

```
x = 1;  
while (x <= 10)  
{  
    printf("%d\n", x);  
    x++;  
}
```

รูปแบบ :

```
do  
{  
    statement (s) ;  
} while ( เงื่อนไข );
```

ตัวอย่าง :

```
x = 1;  
do  
{  
    printf("%d\n", x);  
    x++;  
} while (x <= 10);
```



for: กำหนดจำนวนรอบการวนซ้ำได้

รูปแบบ

```
for (ตัวแปร = ค่าเริ่มต้น ; นิพจน์ตรรกะ ; ค่าเพิ่มหรือค่าลด )  
{  
    statement (s) ;  
}
```

ตัวอย่าง 1

```
for (x = 1 ; x <= 5 ; x++)  
    printf("%d\t", x);
```

ตัวอย่าง 2

```
for (x = 5 ; x > 1 ; x--)  
    printf("%d\t", x);
```



- ตัวแปรควบคุมอาจเป็นชนิด character ได้

```
for(ch='a'; ch<= 'z';ch++)
```

```
    printf("ASCII = %c , Decimal = %d\n",ch, ch);
```



- ตัวแปรควบคุมมีได้มากกว่า 1 ตัวแปร

```
for(x=0,y=0 ; x+y<10 ; ++x,++y)
```

```
printf(“%d\n”,x+y);
```

break;

- เป็นคำสั่งที่ให้จบออกจากการทำงานของคำสั่ง switch .. Case
- หรือออกจากการทำงานแบบวนซ้ำ while, do..while, for

ตัวอย่าง break; กับ while loop

```
double x, sqrt_x;
printf("Enter number : "); scanf("%lf", &x);
while (x <= 100) {
    if (x < 0) {
        printf("ERROR - NEGATIVE VALUE FOR X");
        break;
    } // end of if
    sqrt_x = sqrt(x);
    printf("Square root of %lf is %lf \n", x, sqrt_x);
    printf("Enter number : "); scanf("%lf", &x);
} // end of while
printf("END OF JOB");
```

continue;

- เป็นคำสั่งที่ให้กลับไปทำยังคำสั่งแรกของคำสั่งควบคุม for, while, do..while ใหม่ โดยไม่ต้องทำคำสั่งส่วนที่เหลือภายใต้คำสั่งควบคุม

ตัวอย่าง continue; กับ for loop

```
#include <stdio.h>
void main ()
{
    int x;
    for (x=0; x<=10; x++)
    {
        if (x%2==0)
            continue;
        printf("%d \n",x);
    }
}
```



Classwork 1



Lab5_1

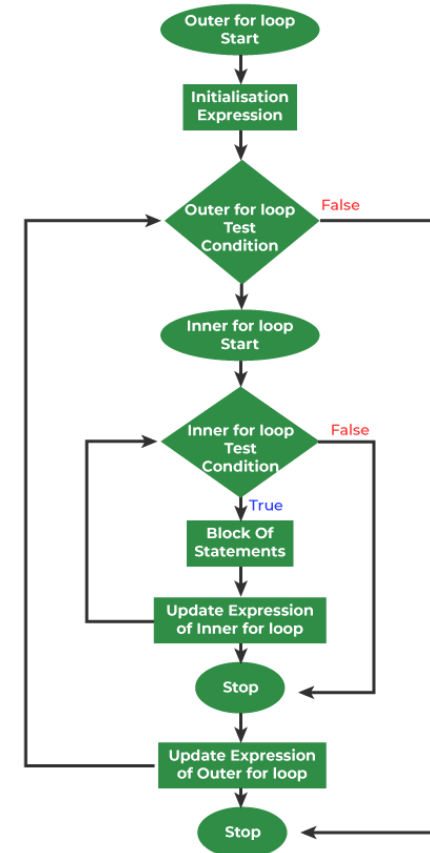
- เขียนโปรแกรมเพื่อรับตัวเลขเข้ามา 5 จำนวนเพื่อคำนวณหาค่าเฉลี่ย

```
Enter number 1:1  
Enter number 2:2  
Enter number 3:3  
Enter number 4:4  
Enter number 5:5  
Average is 3.000000
```

Nested Loop

- Loop inside loops
- อ้างถึงทุกรูปแบบของ loop ได้ โดยใช้ flowchart
- ตัวอย่างรูปแบบของ for

```
for ( initialization; condition; increment ) {  
  
    for ( initialization; condition; increment ) {  
  
        // statement of inside loop  
    }  
  
    // statement of outer loop  
}
```



<https://www.geeksforgeeks.org/nested-loops-in-c-with-examples/>



Nested Loop

```
#include<stdio.h>
main()
{
    int end = 5;

    printf("Pattern Printing using Nested for loop");

    int i = 1,j=1;

    for (i =1;i<=end;i++) {
        printf("\n");
        for (j=1;j <= i;j++) {
            printf("%d ", j);
        }
    }
}
```

```
1  #include<stdio.h>
2  main()
3  {
4      int end = 5;
5
6      printf("Pattern Printing using Nested for loop");
7
8      int i = 1,j=1;
9
10     for (i =1;i<=end;i++) {
11         printf("\n");
12         for (j=1;j <= i;j++) {
13             printf("%d ", j);
14         }
15     }
16 }
17
```

```
Pattern Printi
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
Process return
Press any key
```



Nested loop

- โปรแกรมพิมพ์คำว่า Hi ออกมากี่ตัว

```
1  #include<stdio.h>
2  main()
3  {
4      int a =1, b=1, count = 0;
5      while(a<10)
6      {
7          while(b<10)
8          {
9              printf("Hi ");
10             a++;
11             b++;
12         }
13     }
14 }
```

Nested loop

■ ตัวอย่างรูปของ while

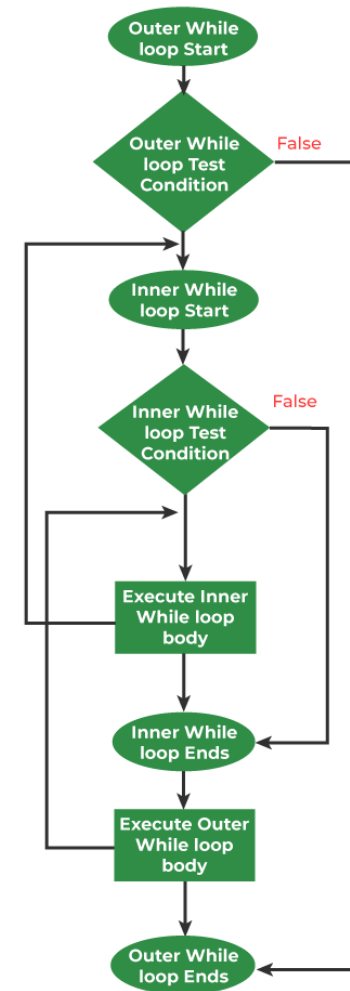
Syntax:

```
while(condition) {
```

```
    while(condition) {
```

```
        // statement of inside loop
    }
```

```
    // statement of outer loop
}
```





```
#include <stdio.h>

int main()
{
    int end = 5;

    printf("Pattern Printing using Nested While loop");

    int i = 1;

    while(i <= end) {
        printf("\n");
        int j = 1;
        while(j <= i) {
            printf("%d ", j);
            j = j + 1;
        }
        i = i + 1;
    }
    return 0;
}
```

Lab5_2

- เขียนโปรแกรมเพื่อวาดรูปสี่เหลี่ยมตามที่ผู้ใช้ต้องการ โดยให้ความกว้างและยาวเป็นจำนวนเต็มจากผู้ใช้

```
Enter width:2
Enter height:3
**
**
**
```



Lab5_3

- เขียนโปรแกรมคิดราคาดอกไม้ พร้อมคำนวณภาษี 10% จากราคาดอกไม้ทั้งหมด โดยร้านขายดอกไม้มีรายการขายดอกไม้พร้อมราคาดังนี้

Menu

1. Rose 19 baht
2. Lily 25 baht
3. Daisy 22 baht
4. Lotus 10 baht
5. Orchid 15 baht
6. Tulip 30 baht
7. End

โปรแกรมจะให้ผู้ใช้ป้อนรายการดอกไม้เข้ามา จนกว่าผู้ใช้จะป้อนค่า 7 แล้วออกจากโปรแกรม