

Control Statement

คำสั่งควบคุม



คำสั่งควบคุมการทำงาน

โดยทั่วไปแล้วโปรแกรมเกิดจากการนำคำสั่งมาต่อเรียงกันให้ทำงานต่อเนื่องกันไปเป็นโปรแกรม คำสั่งควบคุมหมายถึงคำสั่งที่ให้โปรแกรมทำงานข้ามขั้นตอนหรือกระโดดไป หรือทำซ้ำที่จะใดจุดหนึ่งตามข้อมูลที่ได้จากการประมวลผล



คำสั่งควบคุมในภาษาซี

- คำสั่งทดสอบเงื่อนไขในการตัดสินใจ (decision statements)
 - if, if else, nested if และ switch
- คำสั่งให้ทำงานตามจุดที่กำหนด (goto statements)
 - goto (ทางที่ดีไม่ควรใช้)
- คำสั่งวนลูปหรือทำงานซ้ำ ๆ แบบลูป (loop statements)
 - for , while , do while
 - break , continue



ตัวดำเนินการสัมพันธ์

ตัวดำเนินการ	ความหมาย	ตัวอย่าง
==	เท่ากับ	$x == y$
>	มากกว่า	$x > y$
<	น้อยกว่า	$x < y$
>=	มากกว่าหรือเท่ากับ	$x >= y$
<=	น้อยกว่าหรือเท่ากับ	$x <= y$
!=	ไม่เท่ากับ	$x != y$



ตัวอย่าง เมื่อให้ $x = 5$, $y = 10$

นิพจน์

ผลลัพธ์

การแปลความหมาย

$x == y$

$x > y$

$x >= y$

$x <= y$

$x != y$

$x * x < y * y$

$x + y >= x * y$



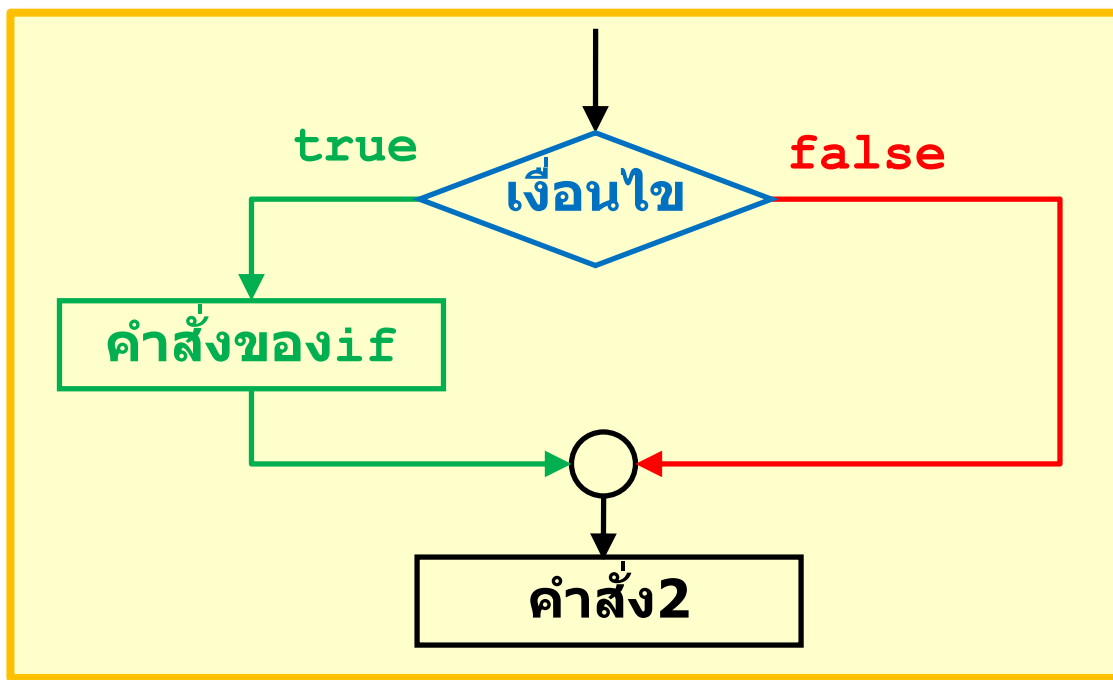
คำสั่งเงื่อนไข if

คำสั่งเงื่อนไข if เป็นคำสั่งให้ตรวจสอบผลลัพธ์ของเงื่อนไข (นิพจน์)

- ถ้าเป็น **จริง (True)** ให้ทำคำสั่งต่าง ๆ ใน if
- ถ้าเป็น **เท็จ (False)** จะไม่ทำคำสั่งต่าง ๆ ใน if

```
if (เงื่อนไข)  
    คำสั่งของif;  
คำสั่ง2;  
หรือ  
if (เงื่อนไข )  
{  
    คำสั่งของif;  
}  
คำสั่ง2;
```

- **True** จะทำคำสั่ง1
แล้วทำคำสั่ง2
- **False** จะข้ามไปทำคำสั่ง2 เลย





คำสั่งเงื่อนไข if

คำสั่งที่อยู่ใน if หากมีคำสั่งมากกว่า 1 คำสั่งสามารถทำได้โดยการใช้เครื่องหมายวงเล็บปีกกา { } เพื่อจัดคำสั่งเป็นกลุ่มเดียวกัน

```
if (เงื่อนไข)
{
    คำสั่ง1 ;
    คำสั่ง2 ;
    ...
    คำสั่งn ;
}
```

```
#include <stdio.h>
int main()
{
    int score;
    scanf("%d",&score);
    if (score >= 50)
    {
        printf("Score = %d \n",score);
        printf("You Passed \n");
    }
    printf("Good bye \n");
}
```

70
Score = 70
You Passed
Good bye

ตัวอย่าง 2

```
#include <stdio.h>
main()
{
    char ch1, ch2;
    ch1 = 'g';
    ch2 = 'k';
    printf("ch1 = %d\n", ch1);
    printf("ch2 = %d\n", ch2);
    if (ch2 > ch1)
        printf("ch2 is more than
ch1\n");
}
```

ข้อสังเกต

จะเห็นว่าการเปรียบเทียบตัวอักขระ

ระบบจะนำรหัส ASCII มาเปรียบเทียบกัน

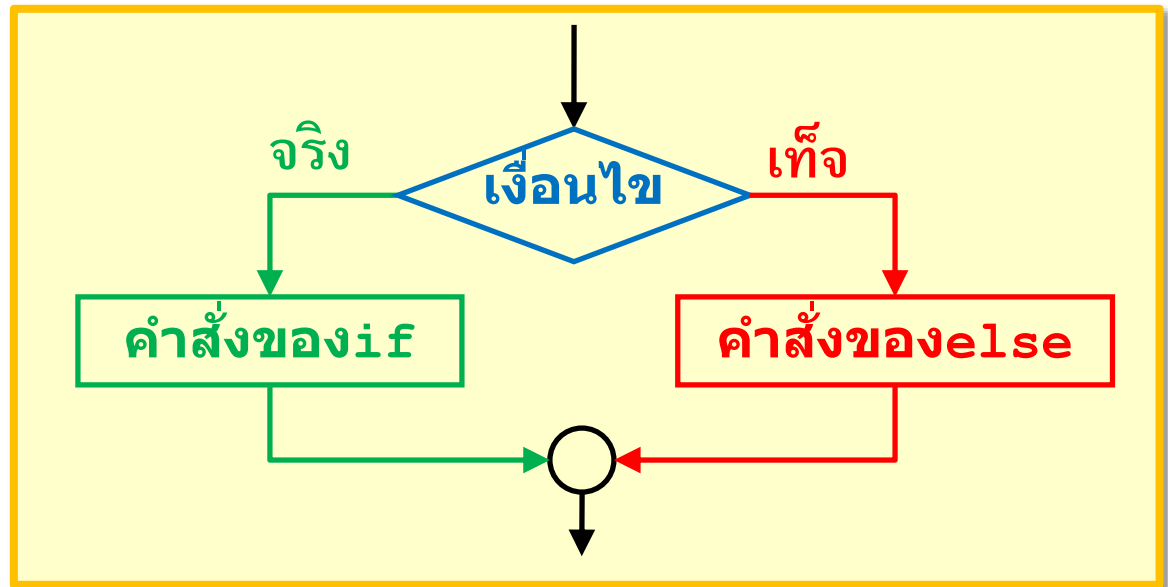


คำสั่งเงื่อนไข if - else

คำสั่งเงื่อนไข if เป็นคำสั่งให้ตรวจสอบผลลัพธ์ของเงื่อนไข (นิพจน์)

- ถ้าเป็น **จริง (true)** ให้ทำคำสั่งต่าง ๆ ใน if
- ถ้าเป็น **เท็จ (false)** ให้ทำคำสั่งต่าง ๆ ใน else

```
if (เงื่อนไข)  
    คำสั่งของif ;  
else  
    คำสั่งของelse ;
```



ตัวอย่าง 3

ให้เขียนโปรแกรมตรวจสอบค่าที่รับมา ถ้าค่ามากกว่า 10 ให้แสดงคำว่า Good ถ้าไม่ใช่ให้แสดงคำว่า Bad

```
#include <stdio.h>
main() {
    int a;
    printf("input number = ");
    scanf("%d", &a);
    if (a > 10) {
        printf("GOOD");
    }
    else{
        printf("BAD");
    }
}
```

ผลการรันโปรแกรม

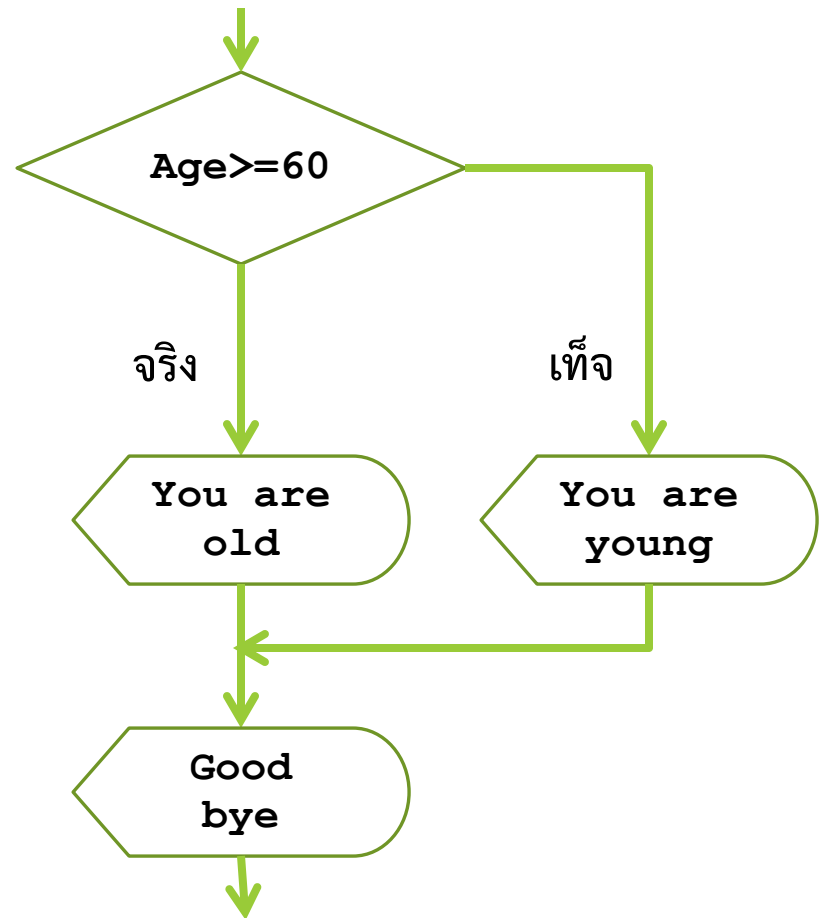
input number = 15
GOOD

ผลการรันโปรแกรม

input number = 3
BAD

โจทย์ if และ else โปรแกรมสอบถามอายุ

```
#include<stdio.h>
main()
{
    int age;
    printf("How old are you ?");
    scanf("%d",&age);
    if(age >= 60)
        printf("You are old \n");
    else
        printf("You are young\n");
    printf("Good bye!");
}
```

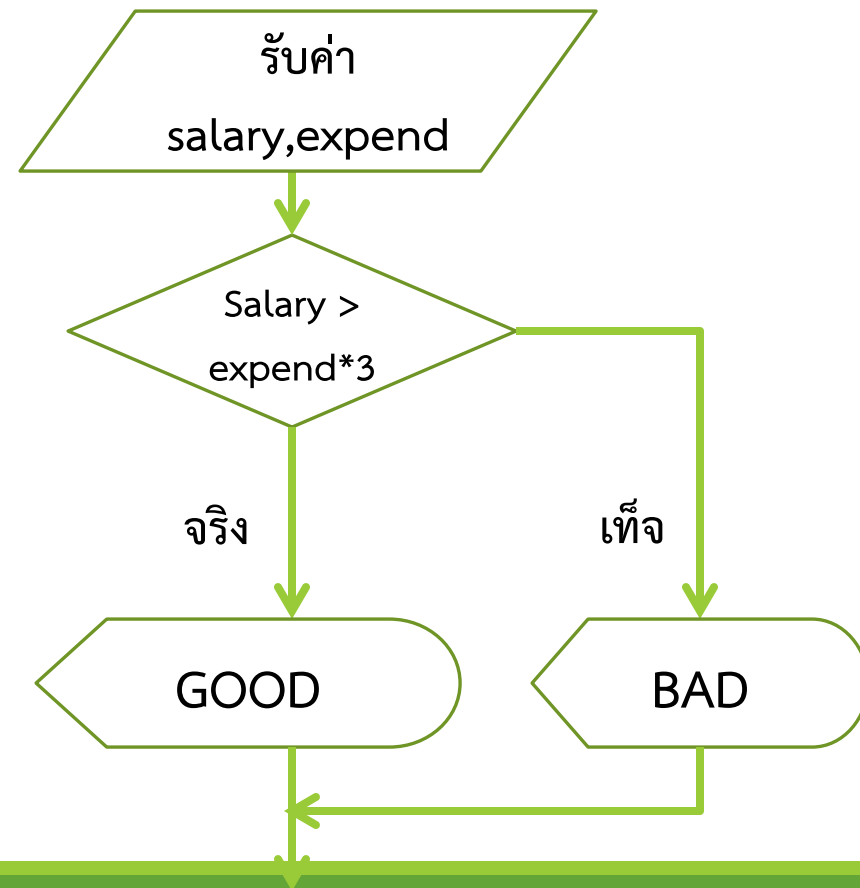


ตัวอย่าง 5

การสร้างเงื่อนไขแบบเปรียบเทียบค่าจากสูตรคำนวณ

สูตรคำนวณ ตัวดำเนินการ ค่าเปรียบเทียบ

เขียนโปรแกรมรับค่าเงินเดือน
กับรายจ่าย ถ้าเงินเดือนมากกว่า
3 เท่าของรายจ่าย ให้แสดงคำว่า
GOOD ถ้าไม่ใช่ให้แสดงคำว่า
BAD





การสร้างเงื่อนไขแบบเปรียบเทียบค่าจากสูตรคำนวณ (ต่อ)

```
#include <stdio.h>
main() {
    int a,b;
    printf("input salary = ");
    scanf("%d",&a);
    printf("input expend = ");
    scanf("%d",&b);
    if(a>b*3) {
        printf("GOOD");
    }
    else{
        printf("BAD");
    }
}
```

ผลการรันโปรแกรม

```
input salary = 10000
input expend = 2500
GOOD
```

ผลการรันโปรแกรม

```
input salary = 5000
input expend = 2500
BAD
```

**จากตัวอย่างนี้แสดงให้เห็นว่า เงื่อนไขสามารถใช้ตัวแปรมาเปรียบเทียบกัน หรือสูตรคำนวณมาเปรียบเทียบก็ได้



คำสั่งเงื่อนไข if – else แบบย่อ

ย่อคำสั่ง if-else ให้กระชับ ดูง่ายมากยิ่งขึ้น

เงื่อนไข ? คำสั่ง1 : คำสั่ง2

ตรวจสอบผลลัพธ์ของเงื่อนไข (นิพจน์)

- ถ้าเป็น **จริง (true)** ให้ทำคำสั่ง1
- ถ้าเป็น **เท็จ (false)** ให้ทำคำสั่ง2

```
int a, b, max;  
max = (a > b) ? a : b;  
ถ้า      a = 2 b = 3 เป็น เท็จ คำตอบคือ 3  
ถ้า      a = 4 b = 3 เป็น จริง คำตอบคือ 4
```

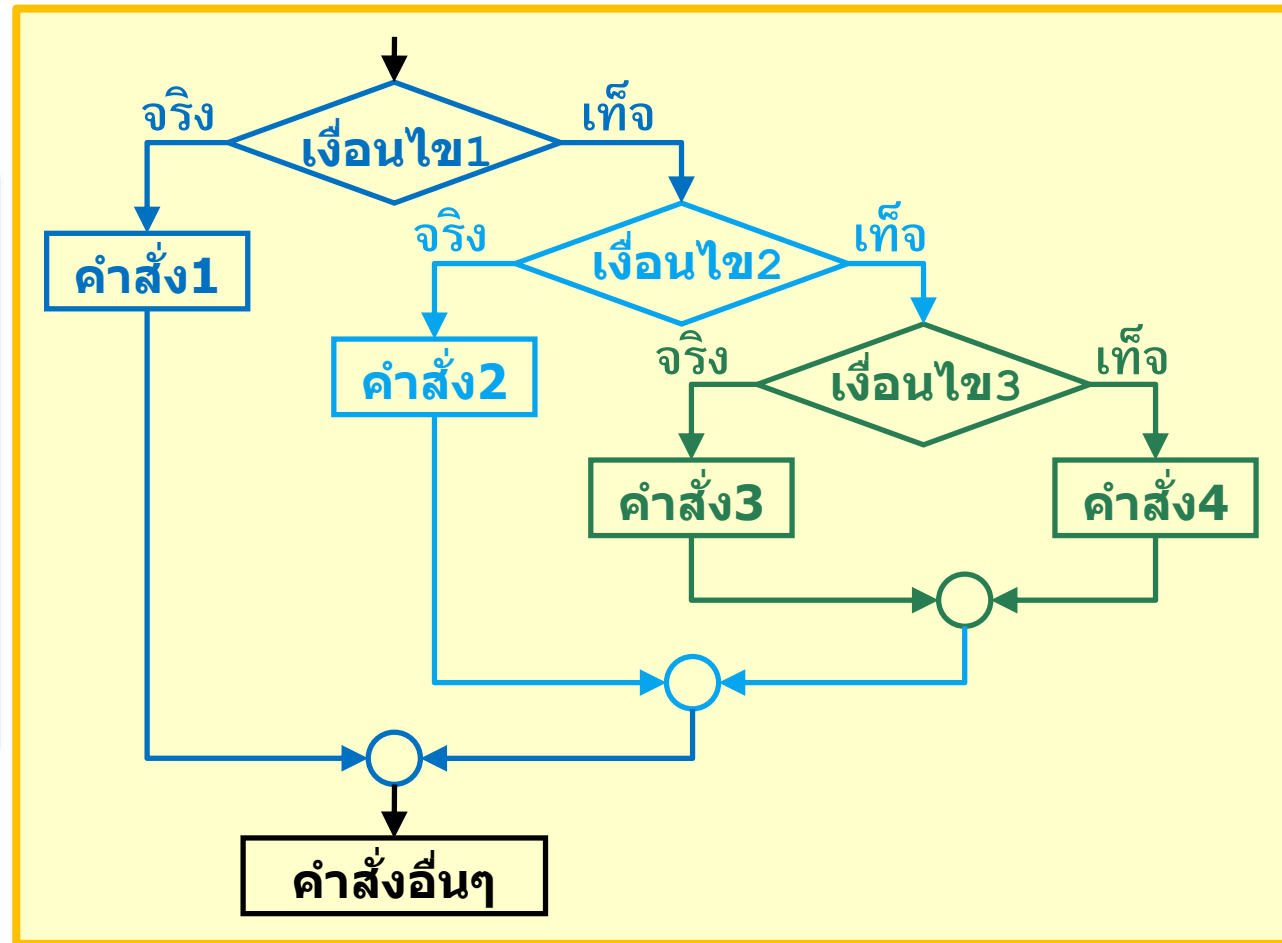
```
if (a > b)  
    max = a;  
else  
    max = b;
```



คำสั่งเงื่อนไข if ซ้อน if (nested if)

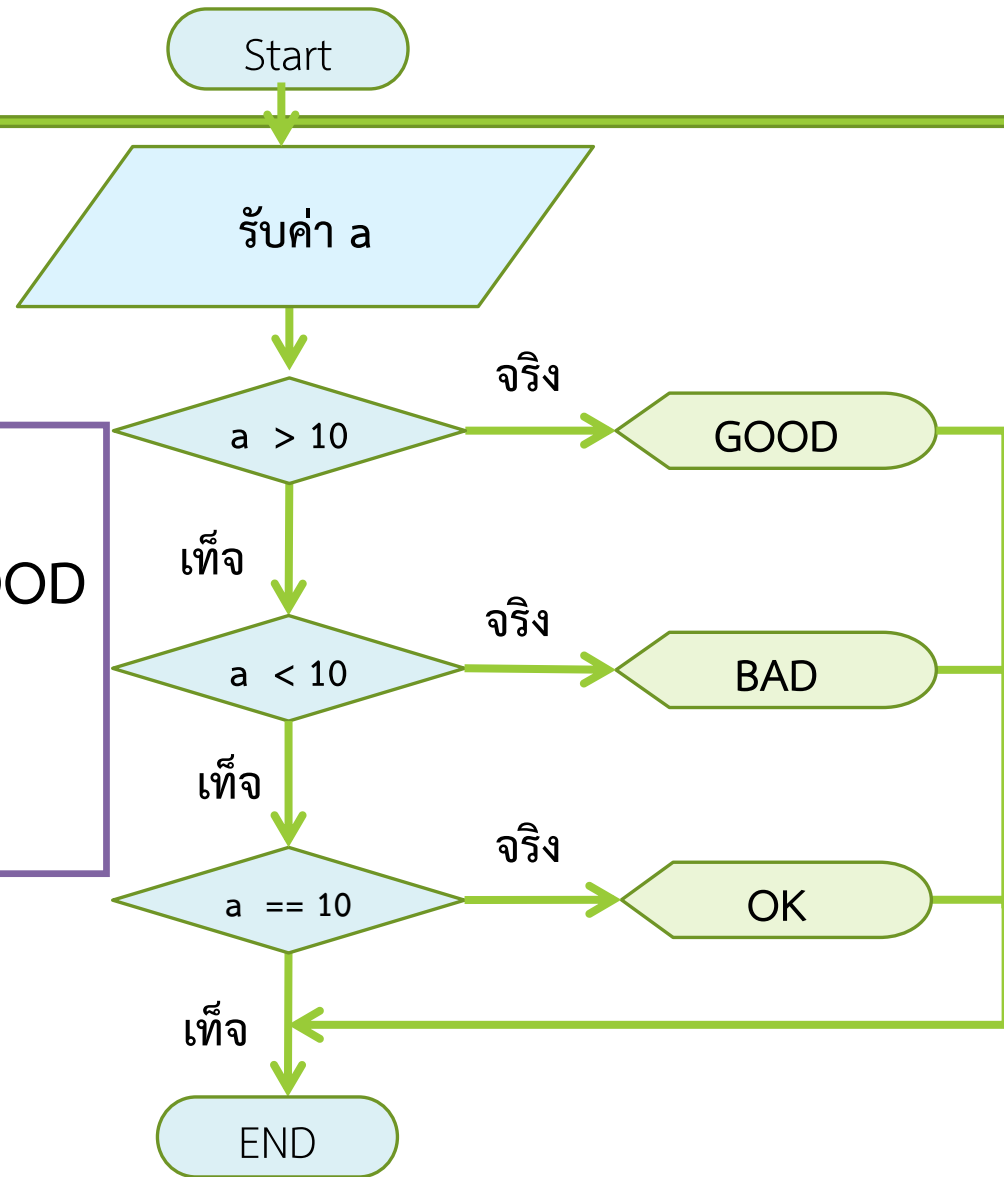
เป็นการนำ คำสั่ง if-else ไปใส่ไว้ภายใน if หรือ if-else อีกทีหนึ่ง ทำให้สามารถตรวจสอบเงื่อนไขได้มากขึ้น

```
if (เงื่อนไข1)
    คำสั่ง1 ;
else if (เงื่อนไข2)
    คำสั่ง2 ;
else if (เงื่อนไข3)
    คำสั่ง3 ;
else
    คำสั่ง4 ;
```



ตัวอย่าง 6

ถ้าค่ามากกว่า 10 ให้แสดงคำว่า GOOD
ถ้าน้อยกว่า 10 ให้แสดงคำว่า BAD
ถ้าเท่ากับ 10 ให้แสดงคำว่า OK





```
#include <stdio.h>
main() {
    int a;
    printf("input number = ");
    scanf("%d", &a);
    if(a>10){
        printf("GOOD");
    }
    else if(a<10){
        printf("BAD");
    }
    else if(a==10){
        printf("OK");
    }
}
```

ผลการรันโปรแกรม

input number = 15
GOOD

ผลการรันโปรแกรม

input number = 3
BAD

ผลการรันโปรแกรม

input number = 10
OK



ข้อแตกต่างระหว่างการใช้ if อย่างเดียวกับการใช้ else if

Tip

กรณีเงื่อนไขสุดท้ายคือเงื่อนไขที่ครอบคลุมถึงกรณีที่เหลือทั้งหมด
สามารถเขียนลดรูปโดยใช้ else แทนได้

```
if (a>10) {  
    printf("GOOD")  
}  
else  
if (a<10) {  
    printf("BAD")  
}  
else  
if (a==10) {  
    printf("OK")  
}
```

```
if (a>10) {  
    printf("GOOD")  
}  
else if (a<10) {  
    printf("BAD")  
}  
else {  
    printf("OK")  
}
```

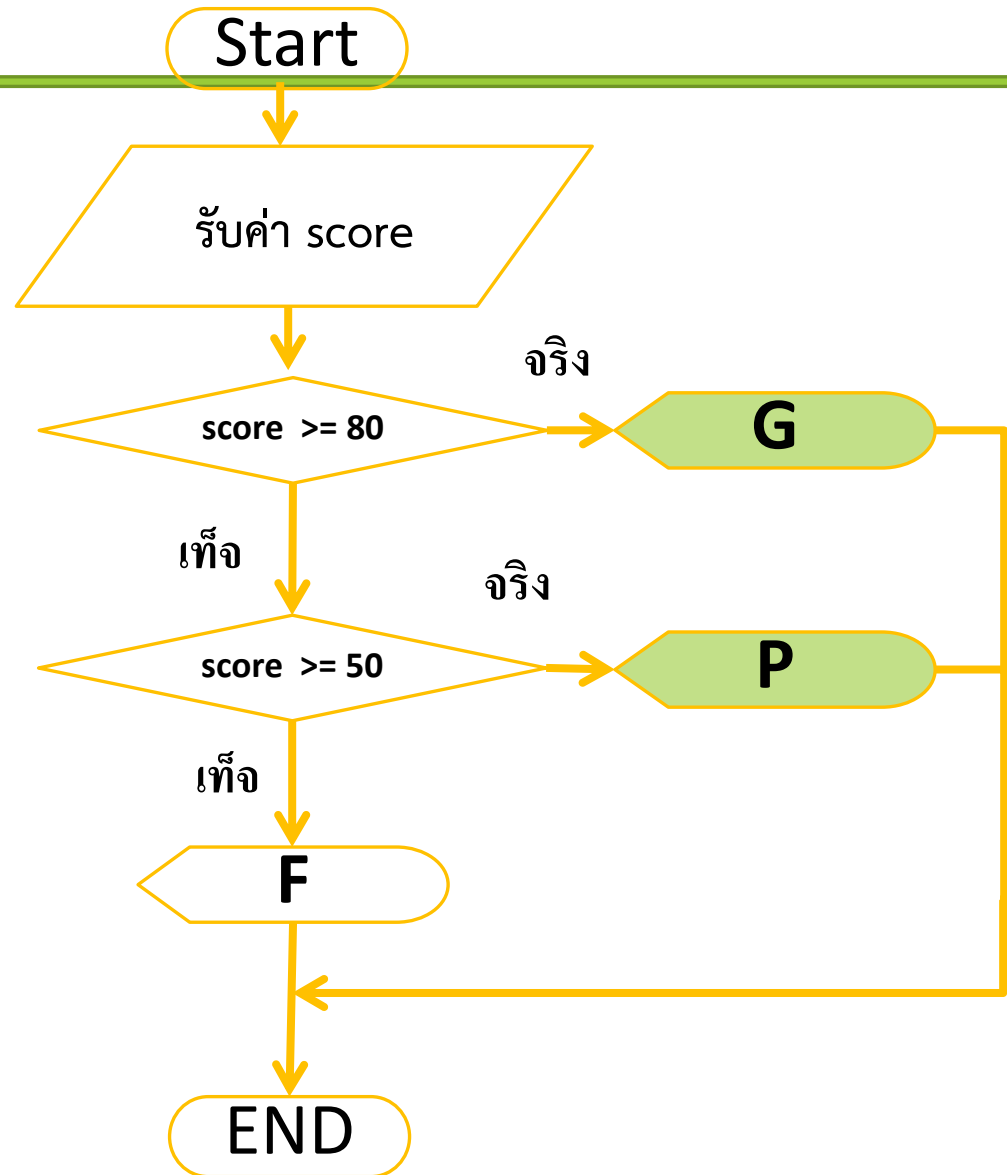
```
if (a>10) {  
    printf("GOOD")  
}  
if (a<10) {  
    printf("BAD")  
}  
if (a==10) {  
    printf("OK")  
}
```

ตัวอย่าง 7

จงเขียนโปรแกรมตัดเกรดโดยรับค่าคะแนนจากคีย์บอร์ด ถ้าได้คะแนนตั้งแต่ 80 ขึ้นไปได้เกรด G ถ้าได้คะแนนตั้งแต่ 50 แต่ไม่ถึง 80 ได้เกรด P ถ้าได้คะแนนไม่ถึง 50 ได้เกรด F

FLOW CHART

กรณีเขียนแบบลดรูป



ตัวอย่าง 7

```
#include <stdio.h>
main() {
    int score;
    printf("input score = ");
    scanf("%d", &sc);
    if(score >=80) {
        printf("grade G");
    }
    else if(score >=50) {
        printf("grade P");
    }
    else{
        printf("grade F");
    }
}
```

ผลการรันโปรแกรม

input score = 90
grade G

ผลการรันโปรแกรม

input score = 75
grade P

ผลการรันโปรแกรม

input score = 40
grade F



คำสั่งเงื่อนไขตรวจสอบหลายเงื่อนไข

การเขียนคำสั่งเงื่อนไข ในบางครั้งจะมีกรณีที่ต้องตรวจสอบเงื่อนไขครั้งเดียวแต่มีเงื่อนไขมากกว่าหนึ่ง เงื่อนไขเพื่อให้ได้ผลลัพธ์เดียว เราสามารถใช้รูปแบบของ if ซ้อน if แบบนี้ได้

```
if(เงื่อนไข 1){  
    if(เงื่อนไข 2){  
        คำสั่งเมื่อเป็นจริง  
    }  
}
```



ตัวอย่าง หากต้องการเปรียบเทียบพร้อมๆ กัน และทำการเปรียบเทียบเป็นจริงทุกค่าเท่านั้น

เช่น ใช้ if ตรวจสอบว่าค่าตัวแปร a กับ b นั้นมากกว่า 0 ทั้งคู่หรือไม่ ถ้าเขียนโปรแกรมโดยใช้ if จะเขียนได้ดังนี้

```
int a=20;   int b=6;

if (a>0)
{
    if (b>0)
    {
        printf("Yes!!  a>0  and  b>0");
    }
}
```



การใช้คำสั่งเชื่อมเงื่อนไขใน IF เดียว

Logical Operator

&&

ตัวเชื่อม “และ (and)”

||

ตัวเชื่อม “หรือ (or)”

รูปแบบคำสั่ง

```
if(เงื่อนไข1 ตัวเชื่อม เงื่อนไข2 .. ตัวเชื่อม เงื่อนไขn){  
    คำสั่งเมื่อเงื่อนไขเป็นจริง;  
}
```

ตัวอย่าง 9

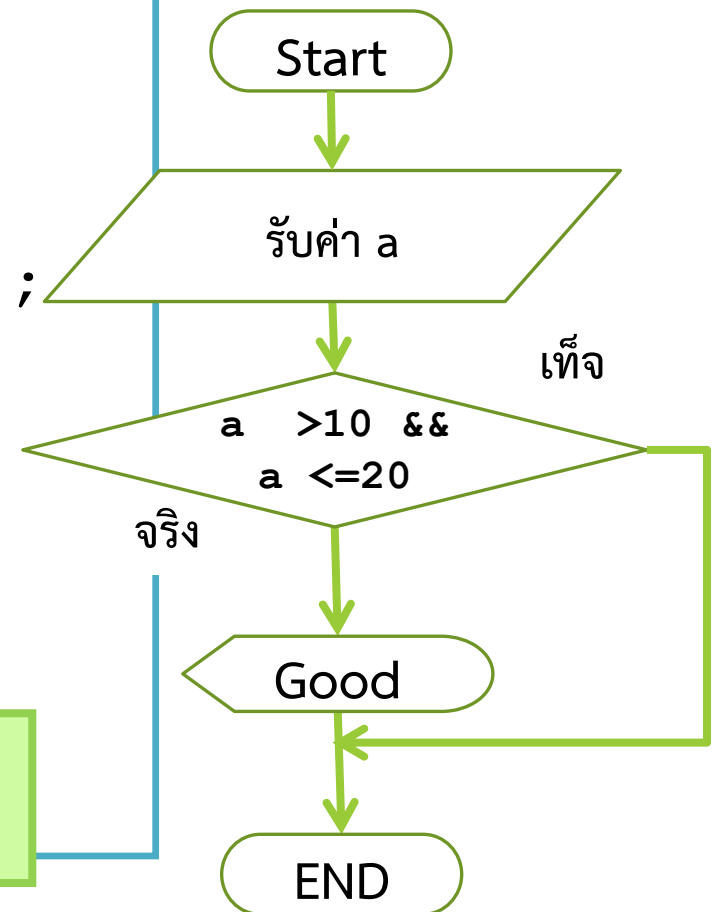
ถ้าค่ามากกว่า 10 แต่ไม่เกิน 20 ให้แสดงคำว่า GOOD

```
#include <stdio.h>
main() {
    int a;
    printf("input number = ");
    scanf("%d", &a);
    if( (a>10) && (a<=20) ) {
        printf("GOOD");
    }
}
```

input number = 15
GOOD

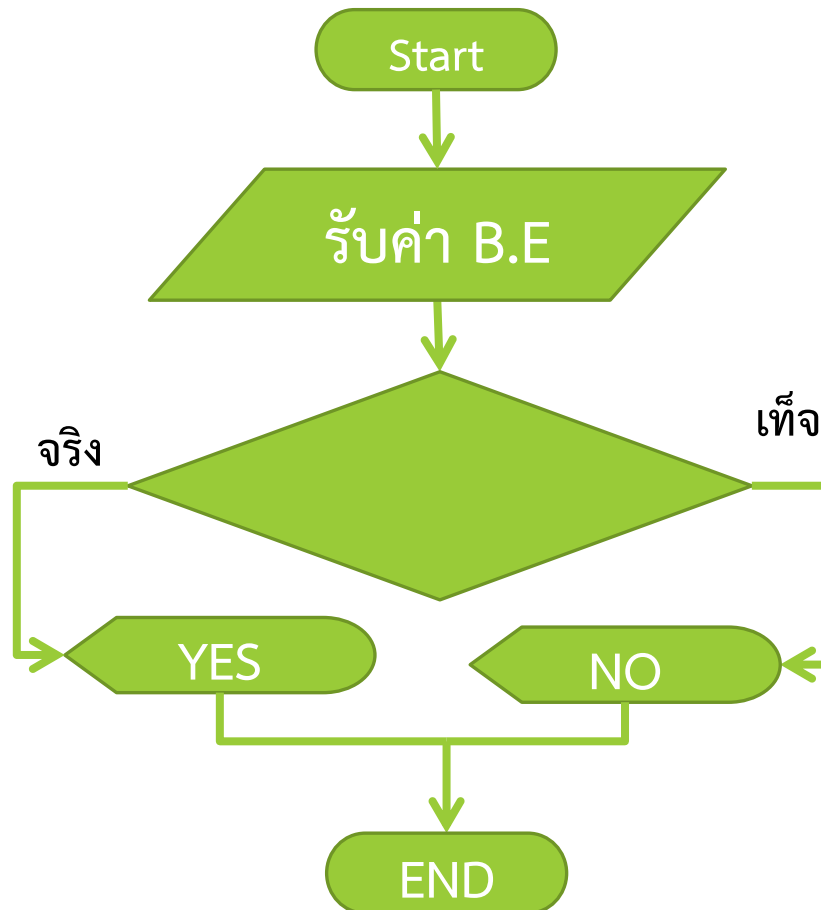
ผลการรันโปรแกรม

input number = 30



แบบฝึกหัด

จงเขียนโปรแกรมตรวจสอบว่า ปี พ.ศ. ที่สนใจเป็นปีอธิกสุรทินหรือไม่ ซึ่งปีอธิกสุรทิน คือ ปี พ.ศ. ที่หารด้วย 4 ลงตัว แต่หารด้วย 100 ไม่ลงตัว หรือหารด้วย 400 ลงตัว



ผลการรันโปรแกรม

input B.E. = 1800
NO

ผลการรันโปรแกรม

input B.E. = 2000
YES



สรุปคำสั่งเลือกทำ if

- คำสั่ง if
 - ใช้ตรวจสอบเงื่อนไข ถ้าผลลัพธ์เป็นจริงจะทำชุดคำสั่งที่ตามมา
- คำสั่ง if - else
 - ใช้สำหรับเลือกทำอย่างใดอย่างหนึ่ง
 - ถ้าเงื่อนไขเป็นจริงจะทำชุดคำสั่งที่ตามหลัง if ถ้าเป็นเท็จทำชุดคำสั่งที่ตามหลัง else
- คำสั่ง if ซ้อน
 - ใช้สำหรับเลือกทำหลาย ๆ อย่างโดยการนำ if หรือ if-else มาซ้อนกัน



คำสั่งเงื่อนไข switch – case

switch – case ใช้ในกรณีที่มีทางเลือกให้ทำงานหลายทาง โดยใช้เงื่อนไขร่วมกัน

โดยพิจารณาผลลัพธ์ของนิพจน์แล้วถ้าตรงกับ case ไต จะทำงานตามคำสั่งใต้ case นั้น

- กรณีมี **default** ถ้าไม่ตรงเงื่อนไขใด ๆ จะทำที่คำสั่ง default
- กรณีไม่มี default ถ้าไม่ตรงเงื่อนไขใด ๆ จะไม่ทำงานใน switch – case นั้นเลย
- default นี้จะมีหรือไม่มีก็ได้

นิพจน์ต้องได้ค่าเป็น

จำนวนเต็มหรือตัวอักษร

switch (นิพจน์,ตัวแปร)

```
{  
    case ค่าที่1 :  
        คำสั่ง1 ;  
        break;  
    case ค่าที่2 :  
        คำสั่ง2 ;  
        break;  
    ...  
    default ค่าที่n :  
        คำสั่งn ;  
}
```



คำสั่งเงื่อนไข switch – case

ในทุกๆ case ควรมีคำสั่ง break; เพื่อปิดท้าย case นั้นๆ ยกเว้น default เนื่องจากเป็นกรณีสุดท้าย จึงไม่ต้องมี ในแต่ละ case สามารถมีคำสั่งมากกว่าหนึ่งคำสั่ง หรือจะไม่มีเลยก็ได้

ถ้าในแต่ละ case มี statement มากกว่า 1 statement สามารถใส่ต่อกันได้เลยโดยไม่ต้องใส่ { }

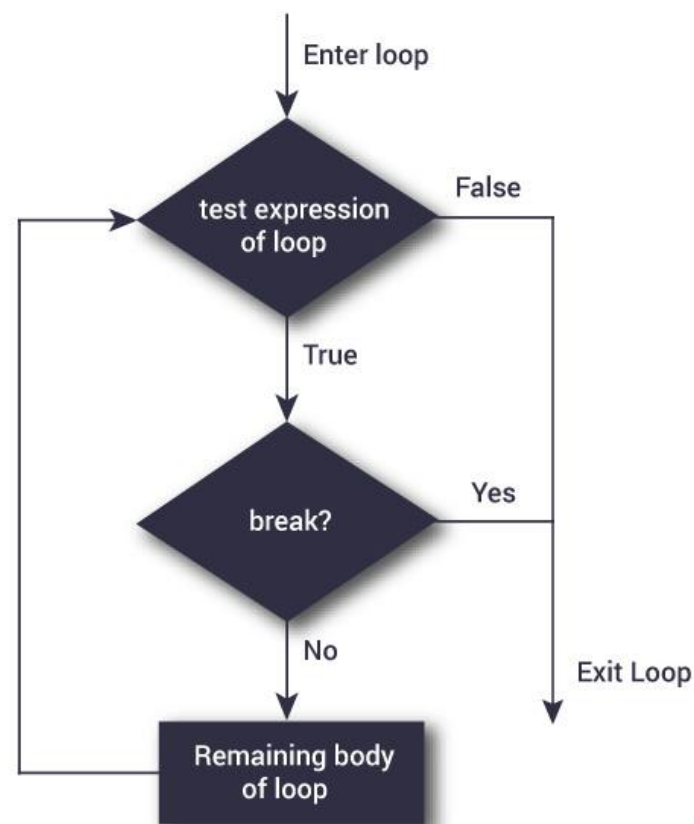


คำสั่ง break

คำสั่ง break คือคำสั่งที่ใช้ยกเลิกคำสั่งที่เหลืออยู่ภายในกลุ่มคำสั่ง { } ถ้าโปรแกรมทำงานมาจนคำสั่ง break ก็จะออกจากกลุ่มคำสั่งปัจจุบันทันที

```
break ;
```

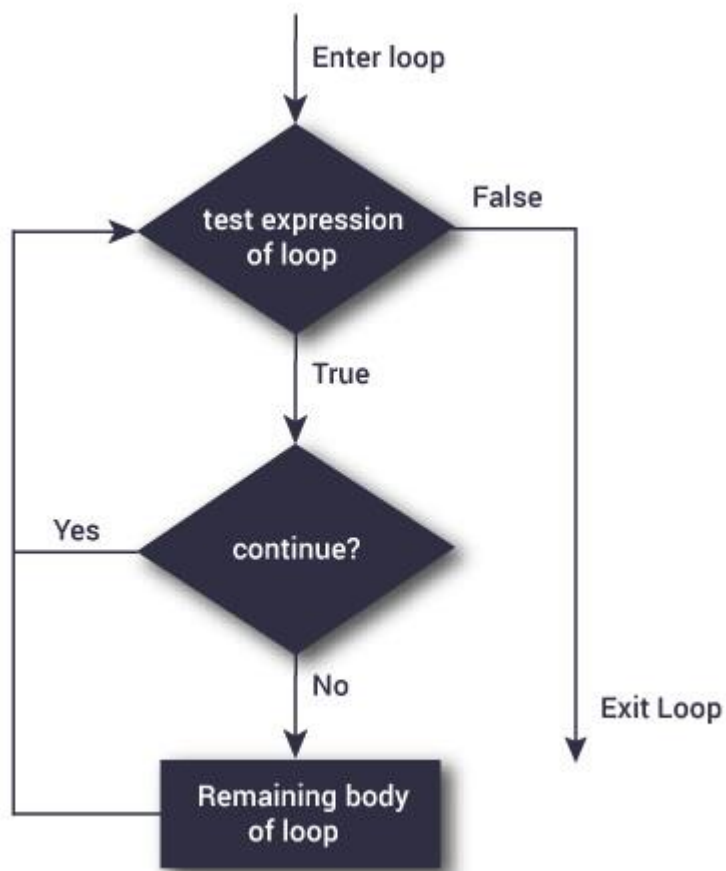
ใช้ร่วมกับคำสั่ง switch โดยใช้เป็นคำสั่งสุดท้ายของแต่ละ case เพื่อไม่ให้ทำงาน case ถัดไปใน switch แล้วออกจาก switch นั้นทันที





คำสั่ง continue

continue ;



ตัวอย่าง 11

```
#include <stdio.h>
main()
{
    int value;
    scanf("%d", &value );
    switch (value % 2)
    {
        case 0 :    printf("Even integer \n");
        case 1 :    printf("Odd integer\n");
    }
}
```

ถ้าค่า $value \% 2$ เท่ากับ 0

จะแสดง

Even integer

ถ้าค่า $value \% 2$ เท่ากับ 1

จะแสดง

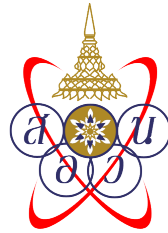
Odd integer

ตัวอย่าง 11

```
scanf ("%d", &value );  
switch (value % 2)  
{  
case 0 :  
    printf("Even integer \n");  
    break;  
case 1 :  
    printf("Odd integer\n");  
    break;  
}  
}
```

แก้ไขโดยใช้คำสั่ง
break เพื่อ
ให้หยุดและออกจาก
switch


```
#include <stdio.h>
main()
{
    char sex;
    printf("are you m(male) of f(female)?");
    scanf("%c",&sex);
    switch (sex)
    {
        case 'm':printf("You are male\n");
        break;
        case 'f':printf("You are female\n");
        break;
        default :printf("Try again !!!\n");
    }
}
```



Loop

คำสั่งทำงานซ้ำ ๆ



คำสั่งทำงานซ้ำ ๆ ในลูป

- คำสั่ง for
 - ใช้กับลูปที่ทราบรอบการทำงานแน่นอน
 - ทำซ้ำจนกว่าเงื่อนไขที่กำหนดเป็นเท็จแล้วจึงออกนอกลูป
- คำสั่ง while
 - ใช้กับลูปที่ไม่ทราบจำนวนทำซ้ำที่แน่นอน
 - ทำซ้ำโดยตรวจสอบเงื่อนไขก่อนเข้าลูป
- คำสั่ง do while
 - ทำซ้ำโดยตรวจสอบเงื่อนไขหลังจากทำงานในลูปไปแล้วหนึ่งครั้ง

การตรวจสอบเงื่อนไข
จะใช้ตัวดำเนินการทางตรรก



คำสั่งวนซ้ำ for

เป็นคำสั่งให้วนรอบ (loop) โดยมีจำนวนรอบในการวนซ้ำที่แน่นอน

for (ค่าเริ่มต้น ; เงื่อนไข ; เพิ่มลดค่า)
คำสั่งของ**for** ;

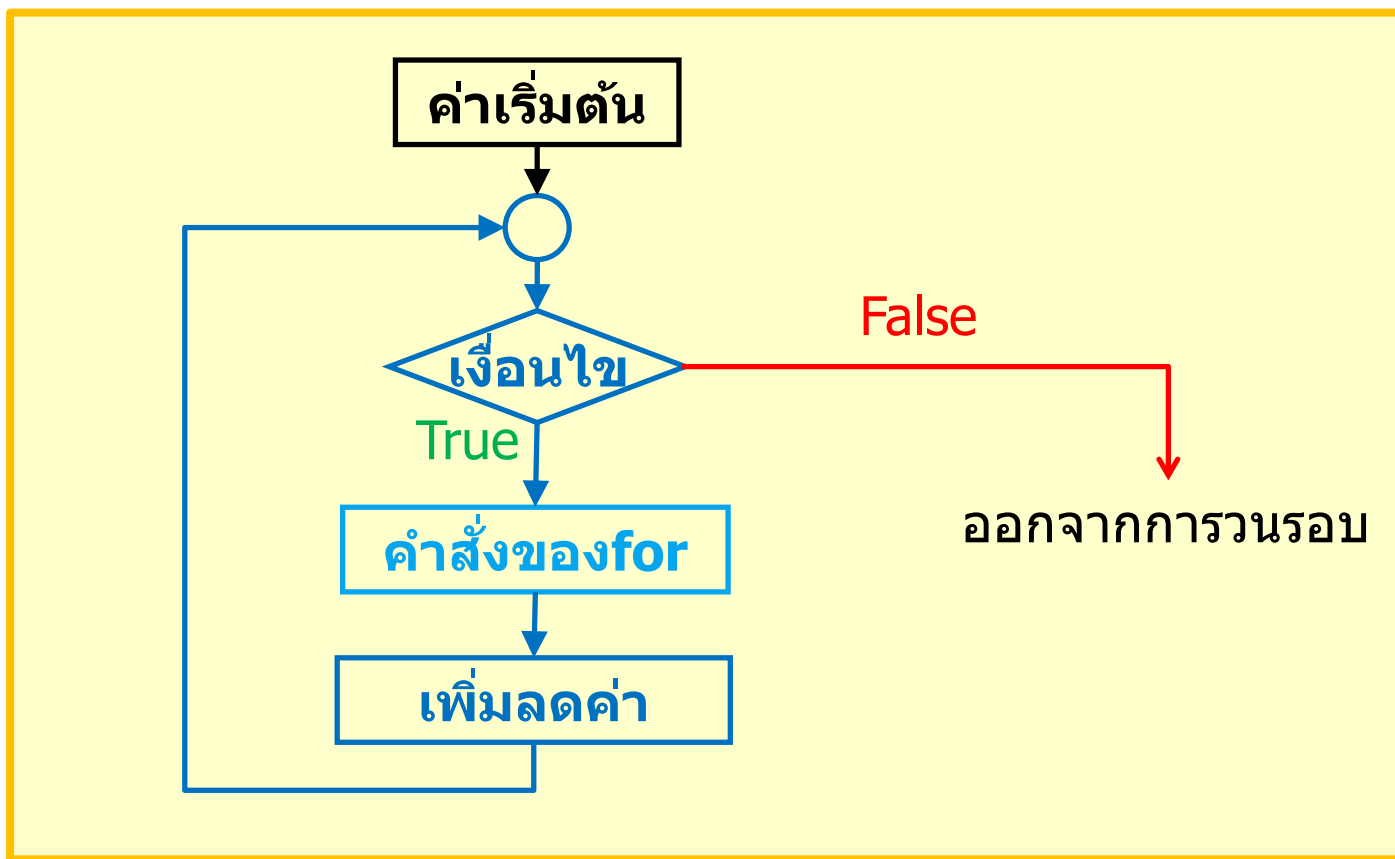
1. **ค่าเริ่มต้น** – ส่วนที่กำหนดค่าเริ่มต้นให้กับตัวแปรนับรอบ
2. **เงื่อนไข** – ใช้ตรวจสอบการทำงานของตัวแปรว่า จริง หรือ เท็จ
 - ถ้าเงื่อนไขเป็น **จริง** ก็ทำงานในลูป for ต่อไป
 - ถ้าเงื่อนไขเป็น **เท็จ** จะออกจากการทำงานของลูป for
3. **เพิ่มลดค่า** – ใช้เพิ่มค่าหรือลดค่าให้กับตัวแปรนับรอบ

หาก **คำสั่งของfor** มีหลายคำสั่ง ให้ใส่ { } คุม



คำสั่งวนซ้ำ for

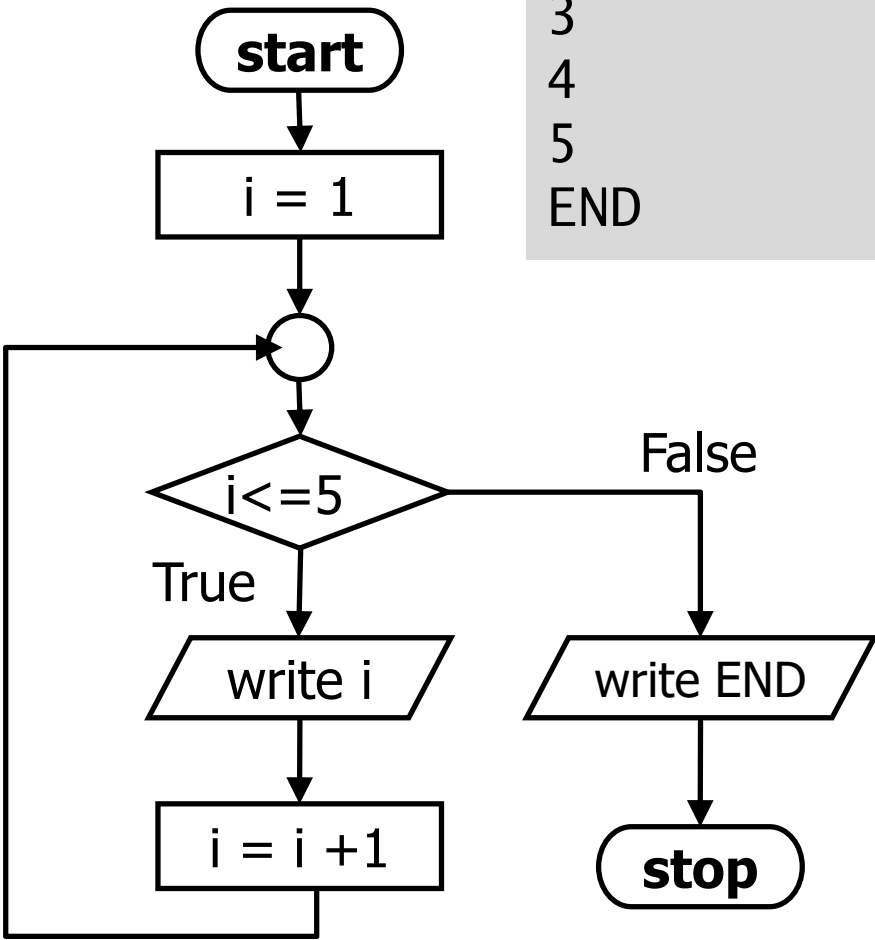
for (ค่าเริ่มต้น ; เงื่อนไข ; เพิ่มลดค่า)
คำสั่งของ**for** ;



ตัวอย่าง 14

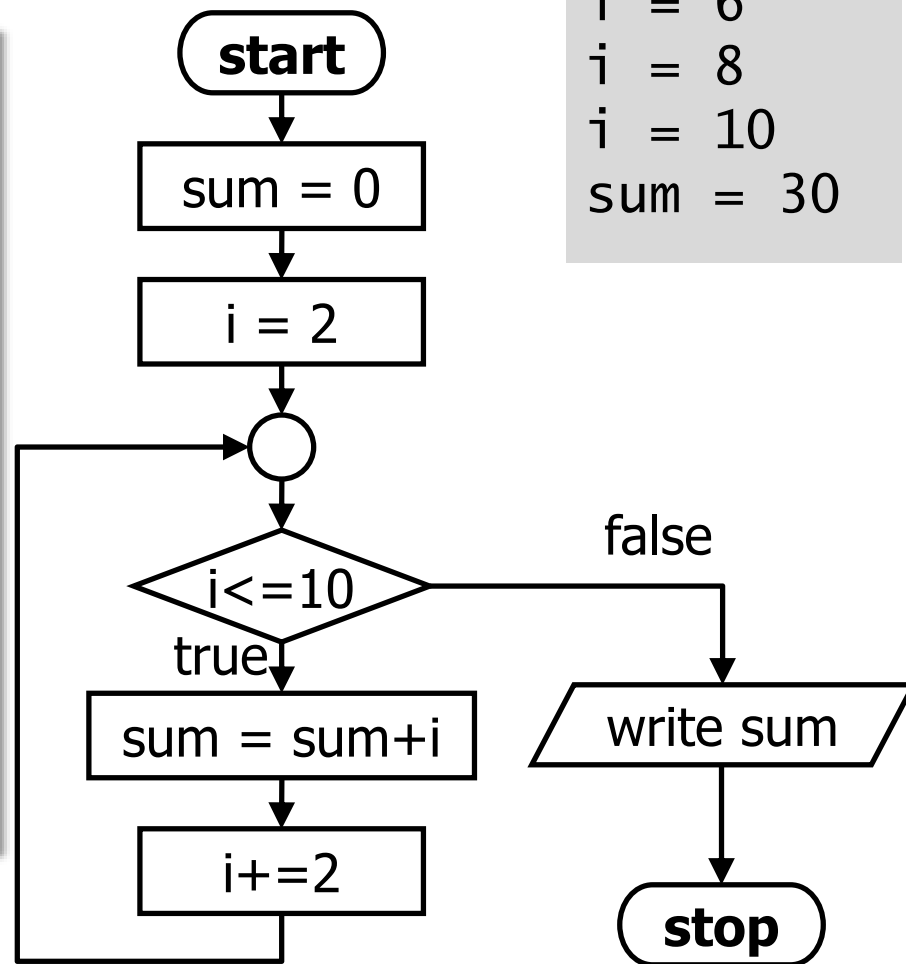
```
#include <stdio.h>
main()
{
    int i;
    for(i = 1; i <= 5; i++)
        printf("%d \n ",i);
    printf("END");
}
```

1
2
3
4
5
END



ตัวอย่าง 16

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



i = 2
i = 4
i = 6
i = 8
i = 10
sum = 30



คำสั่งวนซ้ำ while

while เป็นคำสั่งวนซ้ำ สั่งให้ทำคำสั่งใน while ซ้ำๆ จนกระทั่งเงื่อนไขเป็นเท็จ จึงจะจบการวนซ้ำ

while (เงื่อนไข)
คำสั่งของwhile ;

ตรวจสอบเงื่อนไขก่อนเข้าทำงานใน while ทุกครั้ง

ถ้าเงื่อนไขเป็นจริง จะทำงานคำสั่งใน while แล้วจึงกลับไปตรวจสอบเงื่อนไขอีกครั้ง

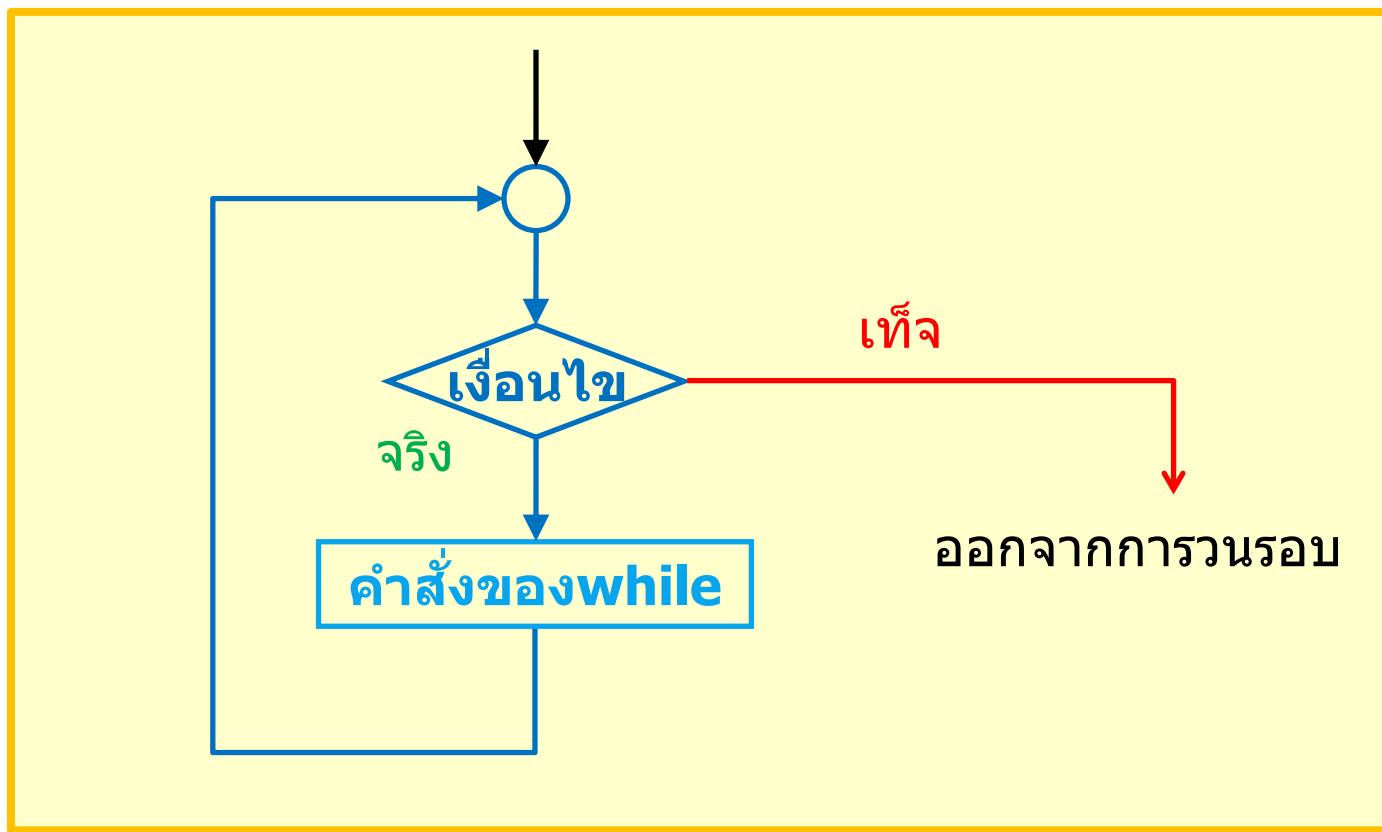
วนซ้ำจนกว่าเงื่อนไขจะเป็นเท็จ จึงออกจาก loop while

หาก **คำสั่งของ while** มีหลายคำสั่ง ให้ใส่ { } คุ่ม



คำสั่งวนซ้ำ while

while (เงื่อนไข)
คำสั่งของwhile ;



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

1
2
3
4
5

ตัวอย่าง 19

```
#include <stdio.h>
main()
{
    int num, sum, n;
    sum = n = 0;
    printf("Enter number \n(0 = End) :");
    scanf("%d", &num);
    while (num != 0)
    {
        printf("num = %d \n", num);
        sum = sum + num;
        n++;
        printf("Enter number:");
        scanf("%d", &num);
    }
    printf("AVG.    = %d ", sum/n);
}
```

```
Enter number
(0 = End) :10
num = 10
Enter number:20
num = 20
Enter number:30
num = 30
Enter number:0
AVG.    = 20
```



คำสั่งวนซ้ำ do – while

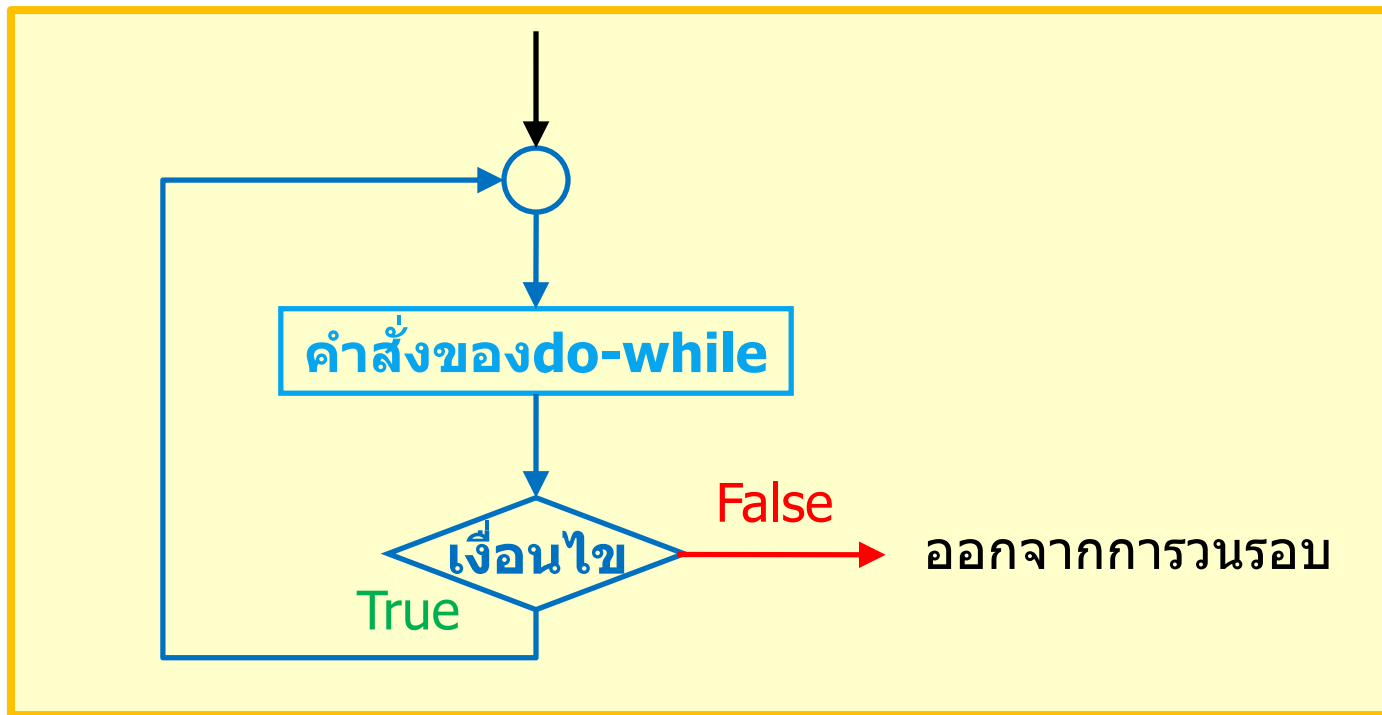
do – while เป็นคำสั่งวนซ้ำ สั่งให้ทำคำสั่งใน do-while หนึ่งรอบ แล้วจึงตรวจสอบเงื่อนไข แล้วทำซ้ำจนกระทั่งเงื่อนไขเป็นเท็จ จึงจะจบการวนซ้ำ

```
do {  
    คำสั่งของdo-while ;  
}  
while (เงื่อนไข);
```



คำสั่งวนซ้ำ do – while

```
do {  
    คำสั่งของdo-while ;  
}  
while (เงื่อนไข);
```



ตัวอย่าง 20

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

```
n = 0
n = 1
n = 2
n = 3
n = 4
```

ตัวอย่าง 21

```
#include <stdio.h>
main()
{
    int num, sum, n;
    sum = n = 0;
    printf("Enter number \n(0 = End) :");
    scanf("%d", &num);
    do
    {
        printf("num = %d \n", num);
        sum = sum + num;
        n++;
        printf("Enter number:");
        scanf("%d", &num);
    } while (num != 0)
    printf("AVG. = %d ", sum/n);
}
```

```
Enter number
(0 = End) :10
num = 10
Enter number:20
num = 20
Enter number:30
num = 30
Enter number:0
AVG. = 20
```

do-while จะคล้ายกับ while
แต่

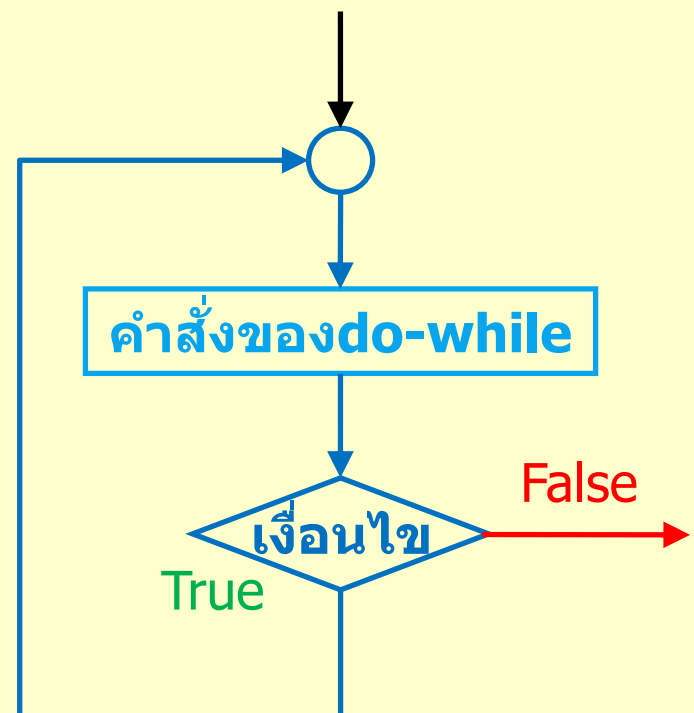
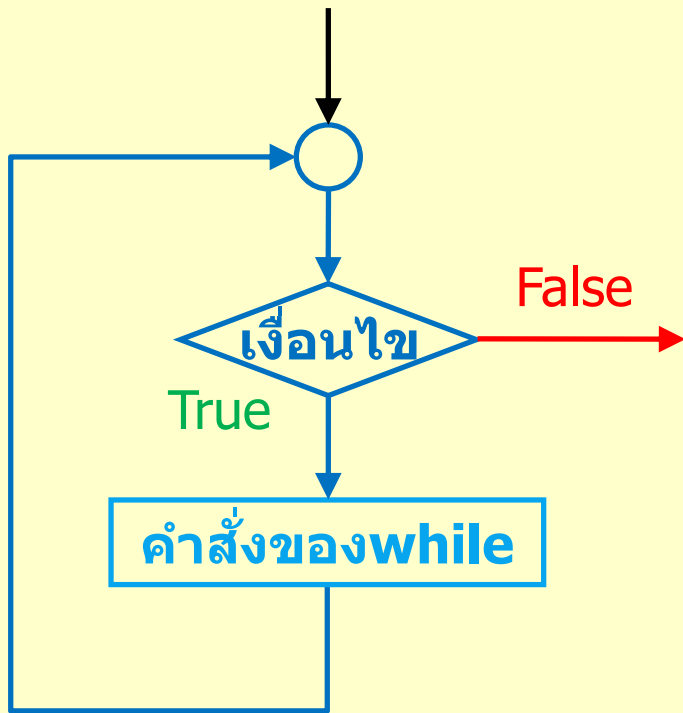
do-while จะต้องทำอย่างน้อย
1 รอบก่อน



เปรียบเทียบ while กับ do while

คำสั่งวนซ้ำ while

คำสั่งวนซ้ำ do-while





Loop อนันต์

โดยปกติแล้วการใช้คำสั่งวนซ้ำจะต้องทำการสร้างเงื่อนไขโดยการใช้ตัวดำเนินการทางคณิตศาสตร์ที่ใช้ในการเปรียบเทียบมาเป็นตัวกำหนด แต่จะมีการใช้คำสั่งลูปในอีกรูปแบบหนึ่งที่ไม่ต้องสร้างเงื่อนไขโดยใส่ 1 เข้าไปในวงเล็บในส่วนของเงื่อนไข ซึ่งจะใช้ในกรณีที่ต้องการใช้งานลูปไปเรื่อยๆ หรือไม่ทราบว่าจะต้องทำงานลูปกี่รอบ

```
while(1){  
    คำสั่ง;  
}
```



จงคำนวณหาพื้นที่วงกลม โดยรับค่ารัศมีจากคีย์บอร์ด โดยให้คำนวณได้หลายรอบ

ตัวอย่าง

input radius = 3

area = 28.26

input radius = 7

area = 153.86

input radius = _

ผลการรันโปรแกรม



Loop เชิงซ้อน

การเขียนโปรแกรมแก้ปัญหาในเรื่องที่มีความซับซ้อนสูง
อาจจะต้องมีการคิดในรูปแบบ 2 มิติ คือ **มีทั้งแถวบนและแถวตั้ง**
(ให้นึกถึงการแสดงผลแบบตาราง EXCEL)

ทำให้จำเป็นต้องใช้ลูปมากกว่า 1 ลูปในการเขียนโปรแกรม
ครั้งเดียว ซึ่ง **ผู้เขียนโปรแกรมจำเป็นต้องอาศัยจินตนาการควบคู่กัน**
ไปด้วย

การทำงานจะเริ่มจากลูป for ที่อยู่ข้างนอกไปสู่การทำงานของ
ลูป for ที่อยู่ข้างใน

ลูป for ที่อยู่ข้างในจะทำจนเสร็จสมบูรณ์จึงจะกลับออกไปทำ
ลูปนอกอีกครั้ง



Looping เชิงซ้อน

```
for(ค่าเริ่มต้น;เงื่อนไข;เปลี่ยนค่าตัวแปรนับรอบ){  
    for(ค่าเริ่มต้น;เงื่อนไข;เปลี่ยนค่าตัวแปรนับรอบ){  
        คำสั่ง;  
    }  
}
```



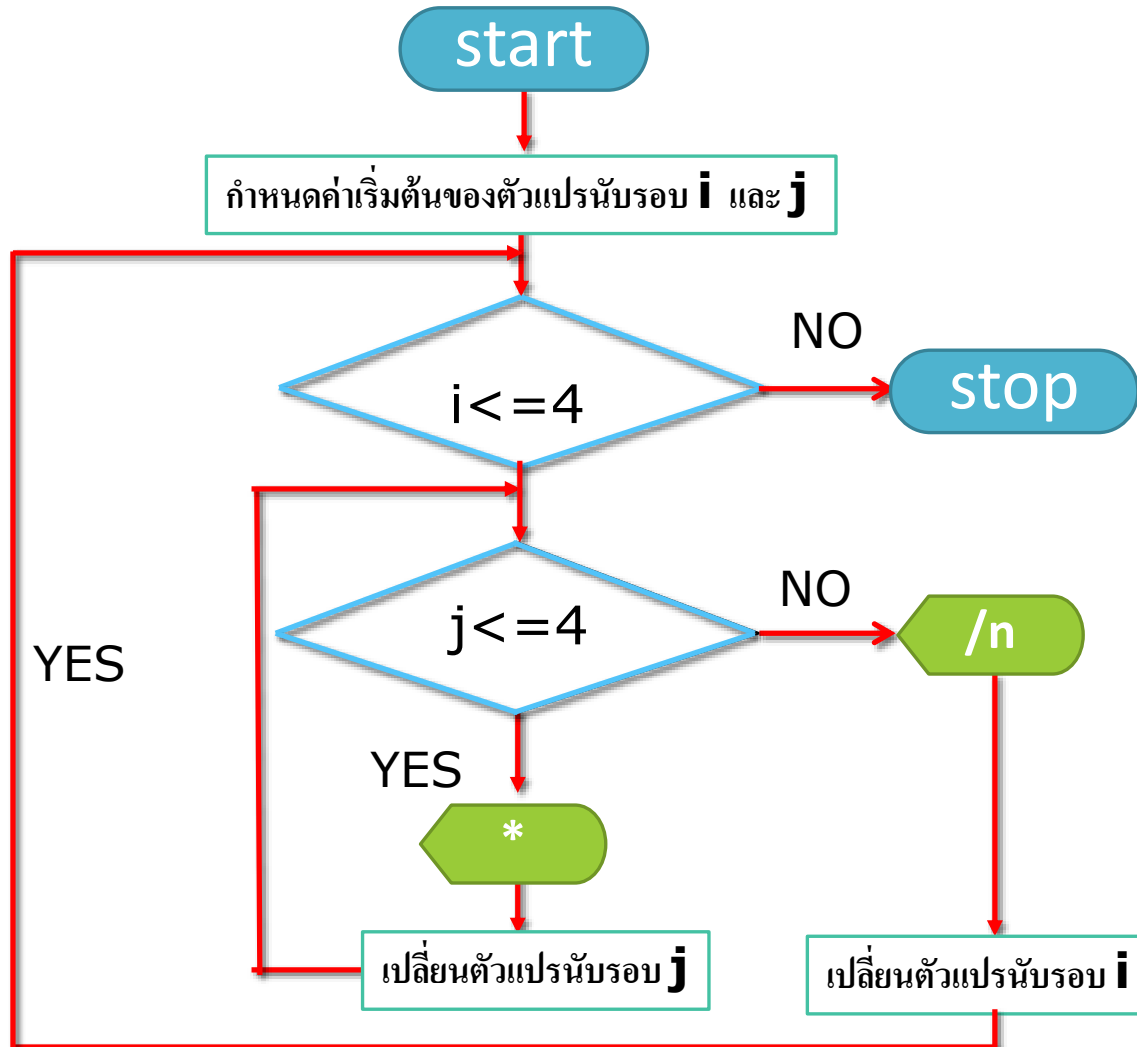
โจทย์ ให้แสดงผลเป็นรูปดาว 4 ดวง จำนวน 4 แถว โดยให้
ใช้ได้แค่ `printf(“*”)` และ `printf(“\n”)` เท่านั้น

```
* * * *  
* * * *  
* * * *  
* * * *
```

แนวคิด หลังจากที `printf(“*”)` ไป 4 ครั้ง ก็จะต้อง
`printf(“\n”)` เพื่อขึ้นบรรทัดใหม่ แล้วทำแบบเดิม
ซ้ำจนครบ 4 บรรทัด

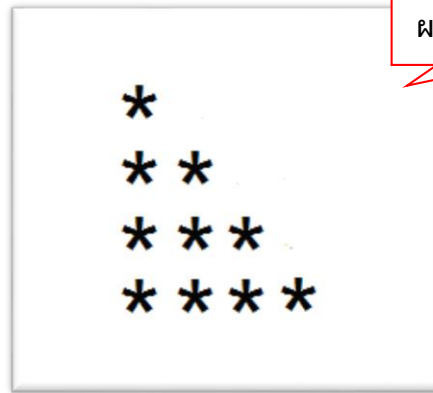


โจทย์ ให้แสดงผลเป็นรูปดาว 4 ดวง จำนวน 4 แถว โดยให้ใช้ได้แค่
`printf(“*”)` และ `printf(“\n”)` เท่านั้น





Lab - ชั้นบันได



ผลการรันโปรแกรม

แนวคิด `printf("*")` แต่ละแถวจะไม่เท่ากัน โดย...

แถวที่ 1 พิมพ์ `* = 1` แล้วขึ้นบรรทัดใหม่

แถวที่ 2 พิมพ์ พิมพ์ `* = 2` แล้วขึ้นบรรทัดใหม่

แถวที่ 3 พิมพ์ `* = 3` แล้วขึ้นบรรทัดใหม่

แถวที่ 4 พิมพ์ `* = 4` แล้วขึ้นบรรทัดใหม่

```
#include <stdio.h>
int main() {
    int n, i, j;
    scanf("%d",&n);
    for(i = 1; i <= n ; i++)
    {
        for(j = 1; j <= i ; j++)
            if(j != i)
                printf("0");
            else
                printf("0\n");
    }
}
```

4


```
#include <stdio.h>
int main()
{
    int i, j ;
    for (i=1;i<=3;i++)
    {
        printf("i = %d\n", i);
        for ( j=1 ; j<=3 ; j++)
        {
            printf("j = %d\n", j);
        }
        printf("\n");
    }
}
```



อ้างอิง

Control Statement , ยนต์ชนก เขาแก้ว, ภาควิชาวิทยาการคอมพิวเตอร์
และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
คำสั่งควบคุมการทำงาน, เอกสารประกอบการอบรม สอวน.สาขาคอมพิวเตอร์
ศูนย์โรงเรียนสามเสนวิทยาลัย
ภาษาซีฉบับภาษาชาวบ้าน , กวินวิชญ์ พุ่มสาขา , ศูนย์เทคโนโลยีเพื่อการเรียน
การสอน โรงเรียนสตรีอ่างทอง