

if & switch & for & while

CH02 流程控制

主講人：資訊二甲 林柏丞

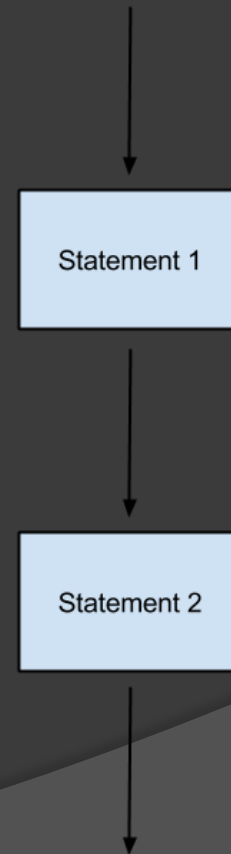
日期：2014/10/15

程式結構

- ◎ 順序性結構 Sequential
- ◎ 選擇性結構 Selection
- ◎ 重複性結構 Iteration

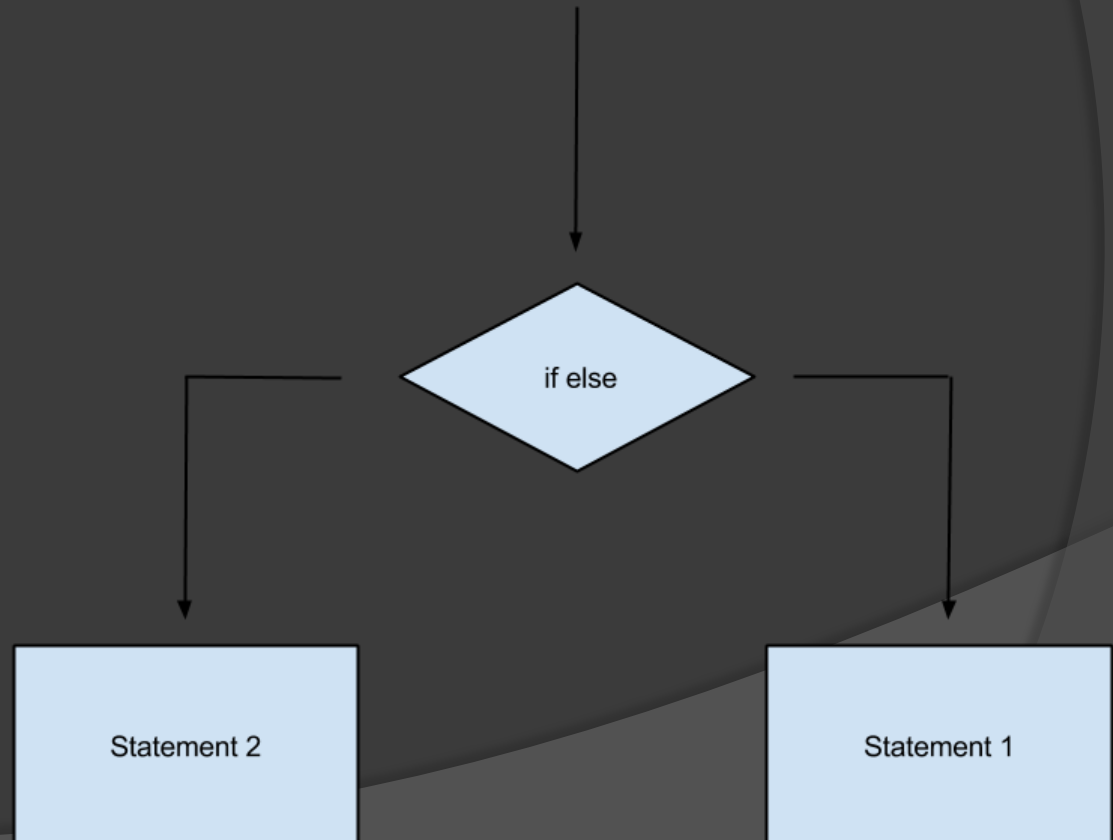
順序性結構 Sequential

- ◎ 一個敘述由上而下接著另一個敘述執行



選擇性結構 Selection

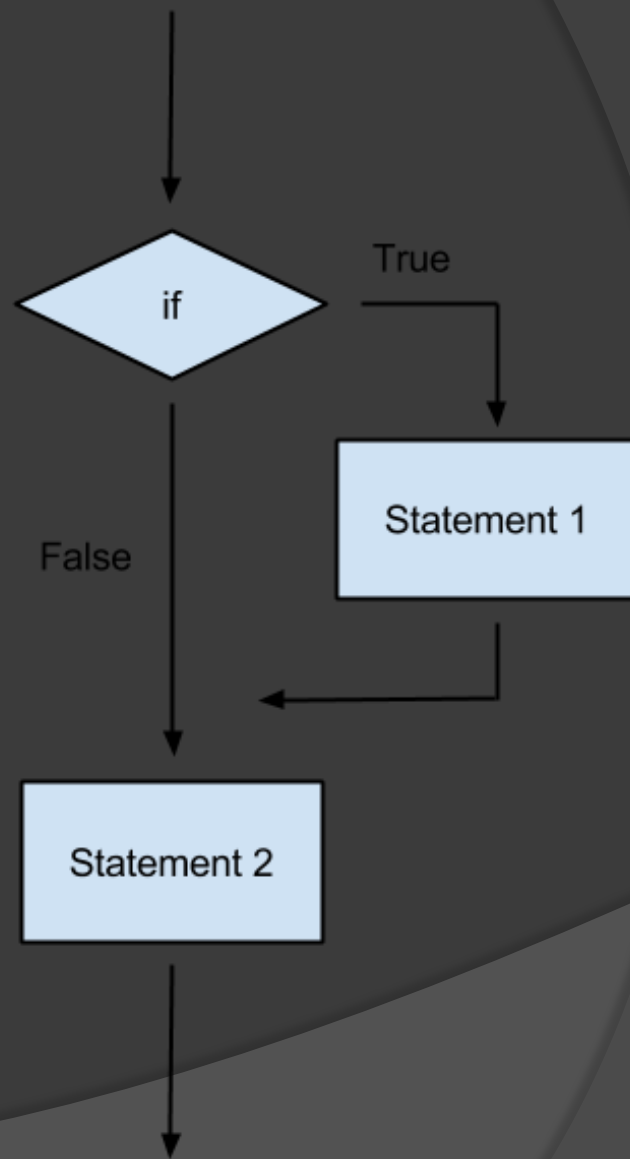
- ◎ 程式執行過程中，依據某條件成立與否去執行不同工作。



If...

```
if(條件式) {  
    程式敘述  
}
```

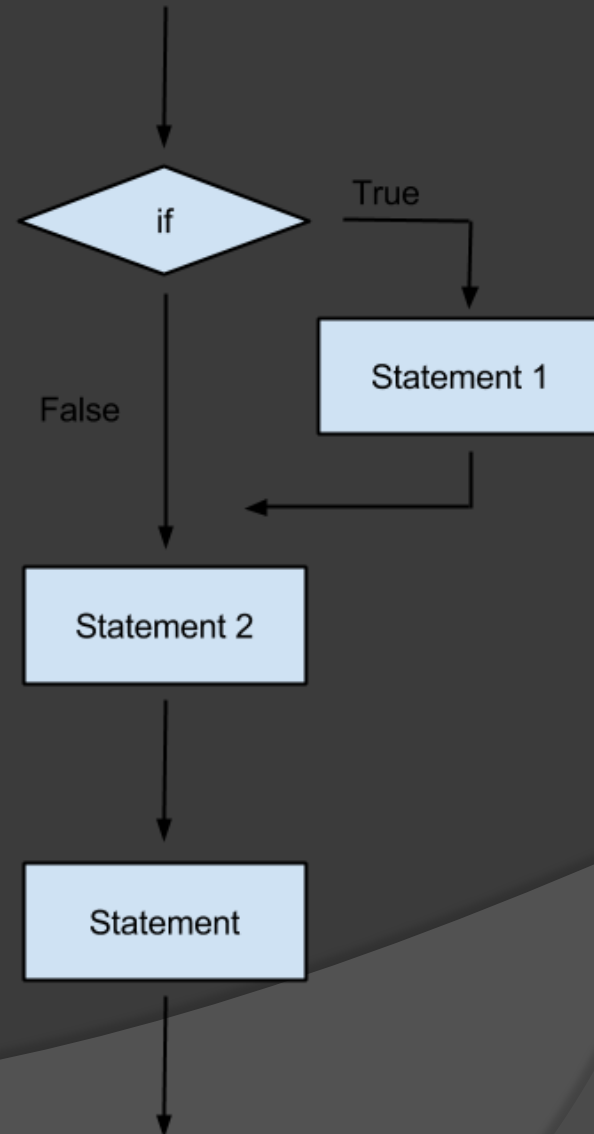
```
If(a%2 == 0) {  
    a = a/2;  
}
```



If...else...

```
if(條件式) {  
    程式敘述1  
}
```

```
else {  
    程式敘述2  
}
```

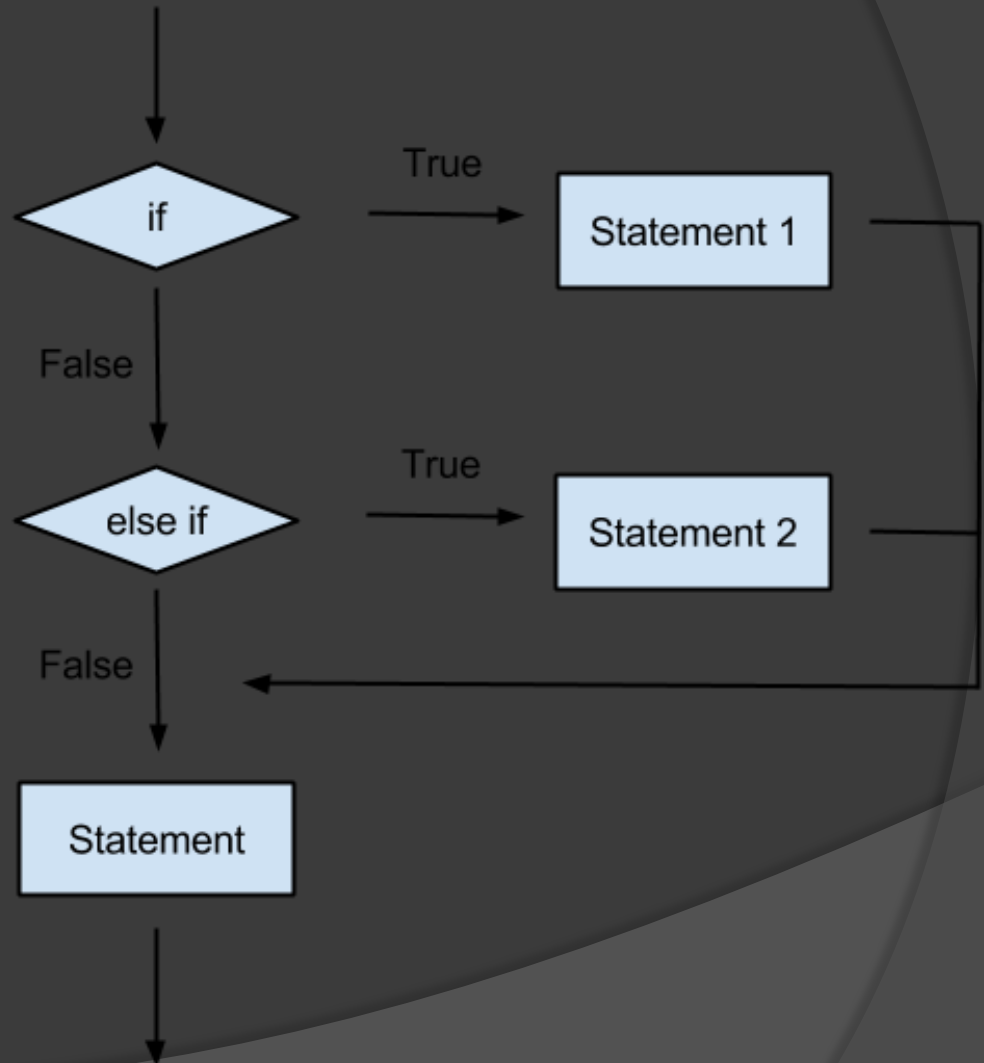


If...else if...else...

```
if(條件式) {  
    程式敘述1  
}
```

```
else if(條件式2) {  
    程式敘述2  
}
```

```
else {  
    程式敘述3  
}
```



範例：if...else if...else...

```
#include<stdio.h>

int main(){
    int score;
    scanf("%d",&score);
    if(score>=90){
        printf("Excellent!\n");
    }
    else if(score>=80){
        printf("Very good!\n");
    }
    else if(score>=70){
        printf("Good!\n");
    }
    else {
        printf("Bad!\n");
    }

    return 0;
}
```

C:\Users\Administrator\Desktop\Untitled1.exe

92
Excellent!

C:\Users\Administrator\Desktop\Untitled1.exe

67
Bad!

Process returned 0 (0x0) execution time : 3.454 s
Press any key to continue.

練習題1

請寫出一程式，輸入西元年分，判斷該年是否為閏年。

(四年一閏，逢百年不閏，四百年又閏)

測資：2100 不為閏年

2000 為閏年

480 為閏年

Switch 多重選擇控制

只能比較數值或字元，某些狀況下比If判斷更有效率。

Switch 多重選擇控制

```
switch(字元或數值) {  
    case 值:  
        statement 1  
        break;  
    case 值2:  
        statement 2  
        break;  
    default:  
        statement 3  
}
```

範例：switch

```
#include<stdio.h>

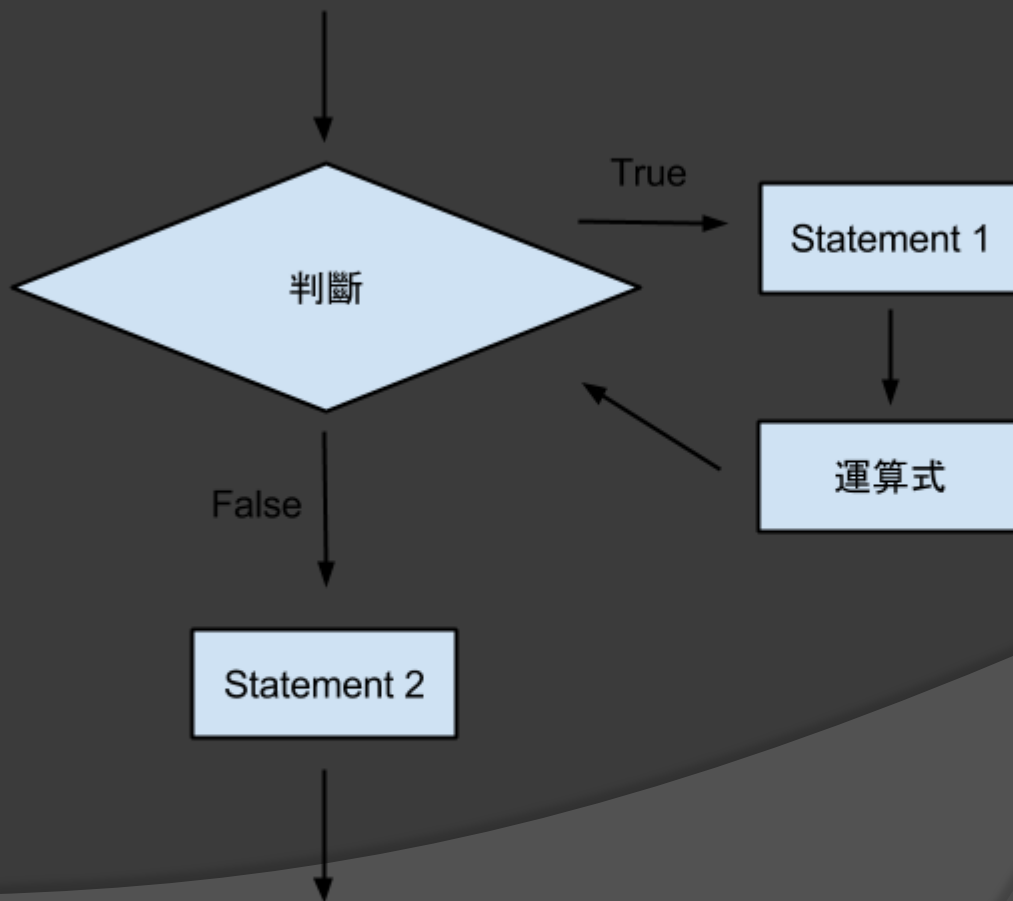
int main(){
    char ch;
    scanf("%c",&ch);
    switch(ch){
        case '+':
            printf("a + b\n");
            break;
        case '-':
            printf("a - b\n");
            break;
        default:
            printf("Error!\n");
    }
    return 0;
}
```

Switch v.s. If

如果純粹比對數字或字元的話，建議使用switch，因為它只會在一開始的switch括號中取出變數值一次，然後將這個值與下面所設定的case比對。但如果您使用if的話，每次遇到條件式時，都要取出變數值。

重複性結構 Iteration

- 程式執行過程中，需要重複做同一個工作好幾次。



for

```
for(起始式 ; 判斷式 ; 運算式) {  
    statement  
}
```

```
for(i=0 ; i<3 ; i++) {  
    printf( “%d” ,i);  
}
```

範例：for

```
#include<stdio.h>

int main(){
    int i,total = 0;
    for(i=1;i<11;i++){
        total = total + i;
    }
    printf("1+2+...+10 = %d\n",total);
    return 0;
}
```

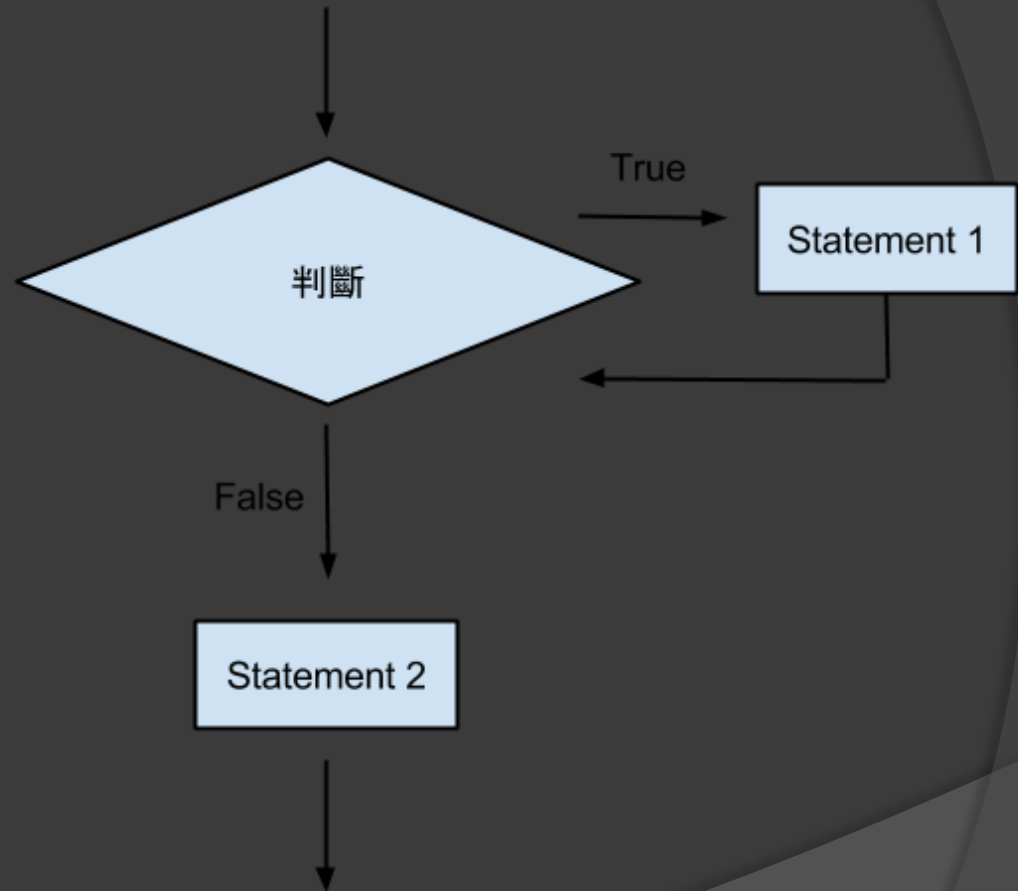
C:\Users\Administrator\Desktop\Untitled2.exe

1+2+...+10 = 55

Process returned 0 (0x0) execution time : 0.014 s
Press any key to continue.

while

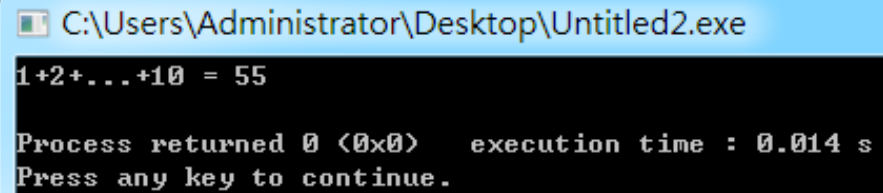
```
while(判斷式) {  
    statement  
}
```



範例：while

```
#include<stdio.h>

int main(){
    int i = 1,total = 0;
    while(i<11){
        total = total + i;
        i++;
    }
    printf("1+2+...+10 = %d\n",total);
    return 0;
}
```



C:\Users\Administrator\Desktop\Untitled2.exe

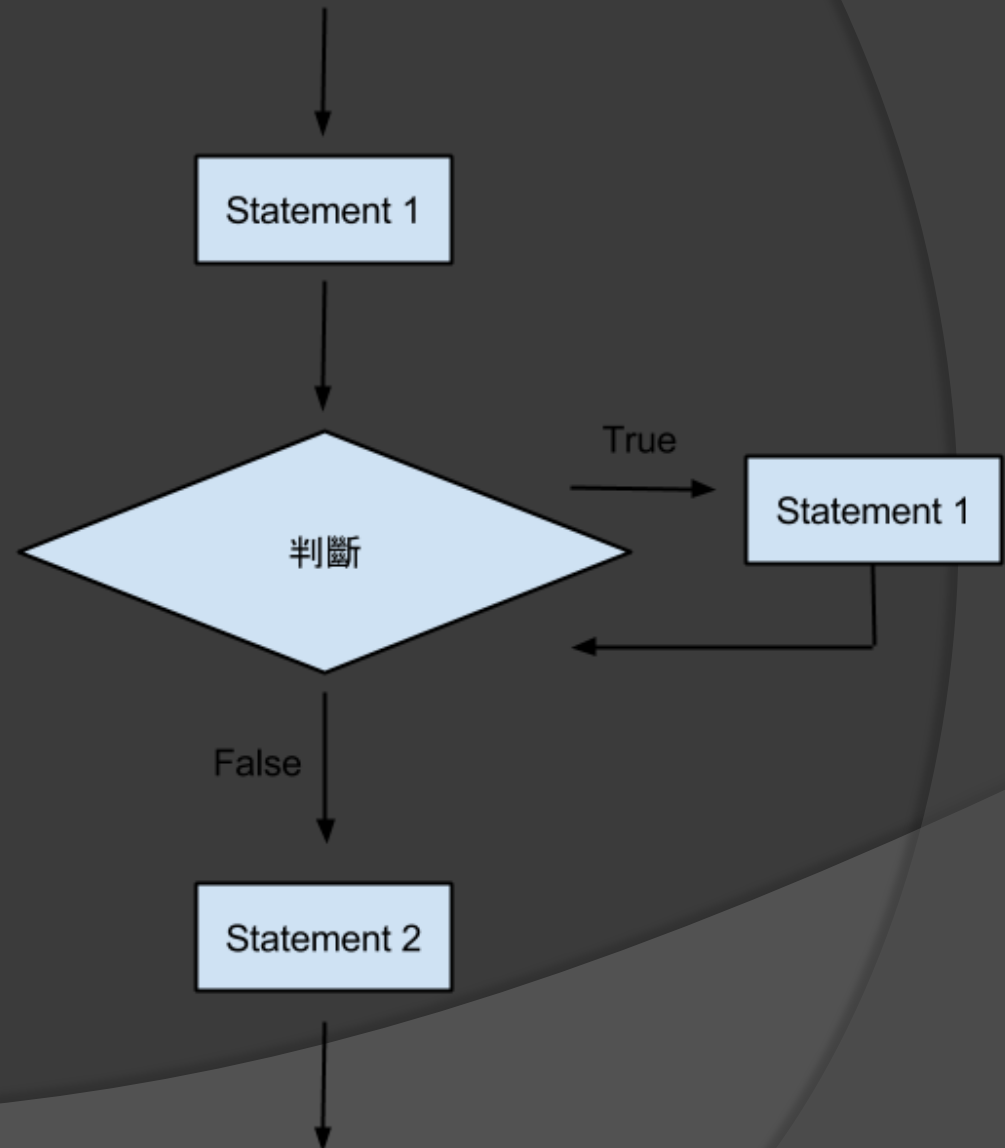
1+2+...+10 = 55

Process returned 0 (0x0) execution time : 0.014 s

Press any key to continue.

do...while

```
do{  
    statement  
}while(判斷式);
```



範例：do...while

```
#include<stdio.h>

int main(){
    int i = 1, total = 0;
    do{
        total = total + i;
        i++;
    }while(i<11);
    printf("1+2+...+10 = %d\n", total);
    return 0;
}
```

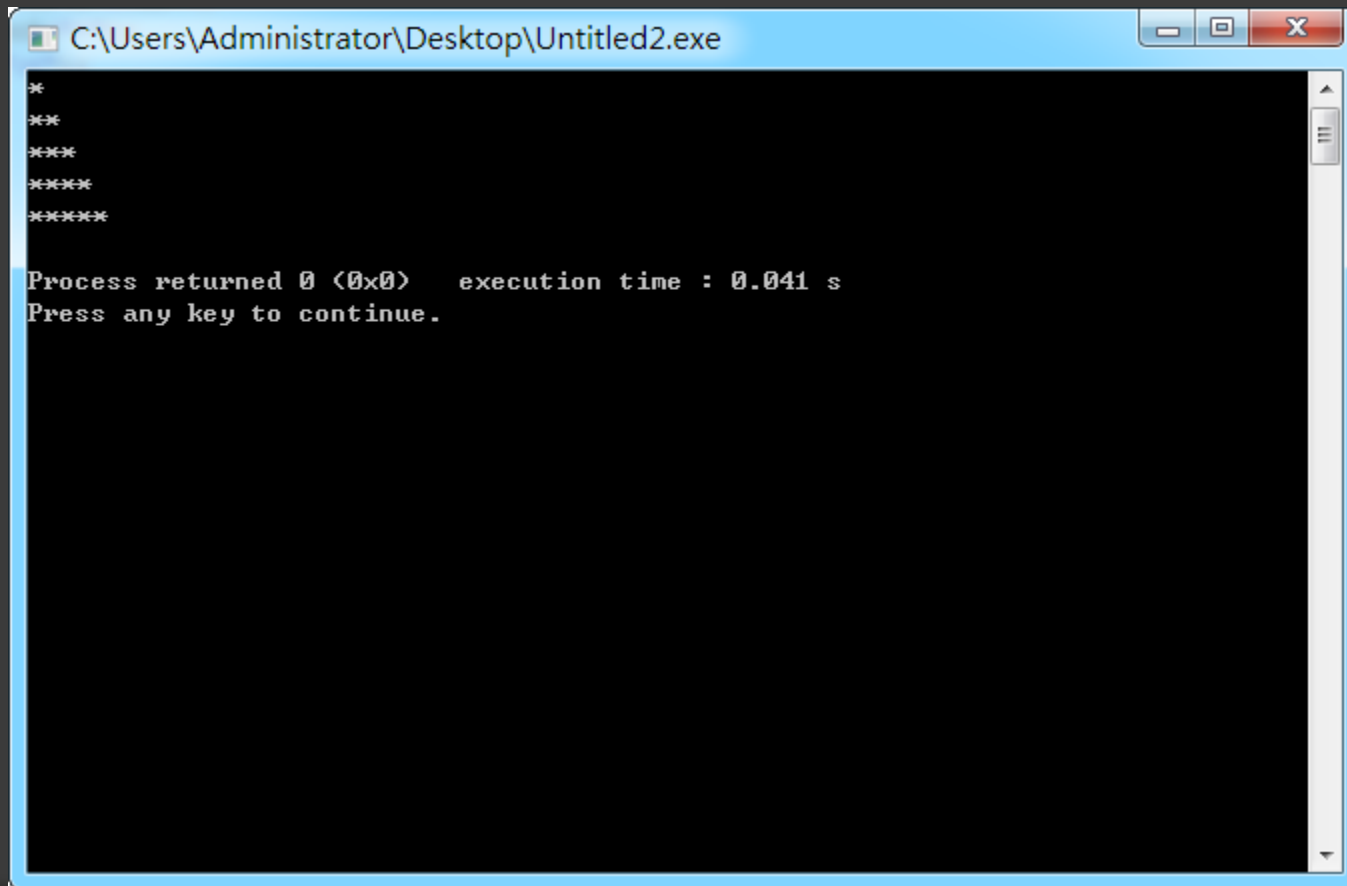
C:\Users\Administrator\Desktop\Untitled2.exe

1+2+...+10 = 55

Process returned 0 (0x0) execution time : 0.014 s
Press any key to continue.

練習題2

印出右斜三角程式



A screenshot of a Windows command prompt window titled "C:\Users\Administrator\Desktop\Untitled2.exe". The window has a blue title bar and standard Windows window controls (minimize, maximize, close). The command prompt area is black with white text. It displays a right-angled triangle of asterisks (*). The first row has 1 asterisk, the second has 2, the third has 3, the fourth has 4, and the fifth has 5. Below the triangle, the text "Process returned 0 (0x0) execution time : 0.041 s" is visible, followed by "Press any key to continue." on the next line. A vertical scrollbar is visible on the right side of the command prompt window.

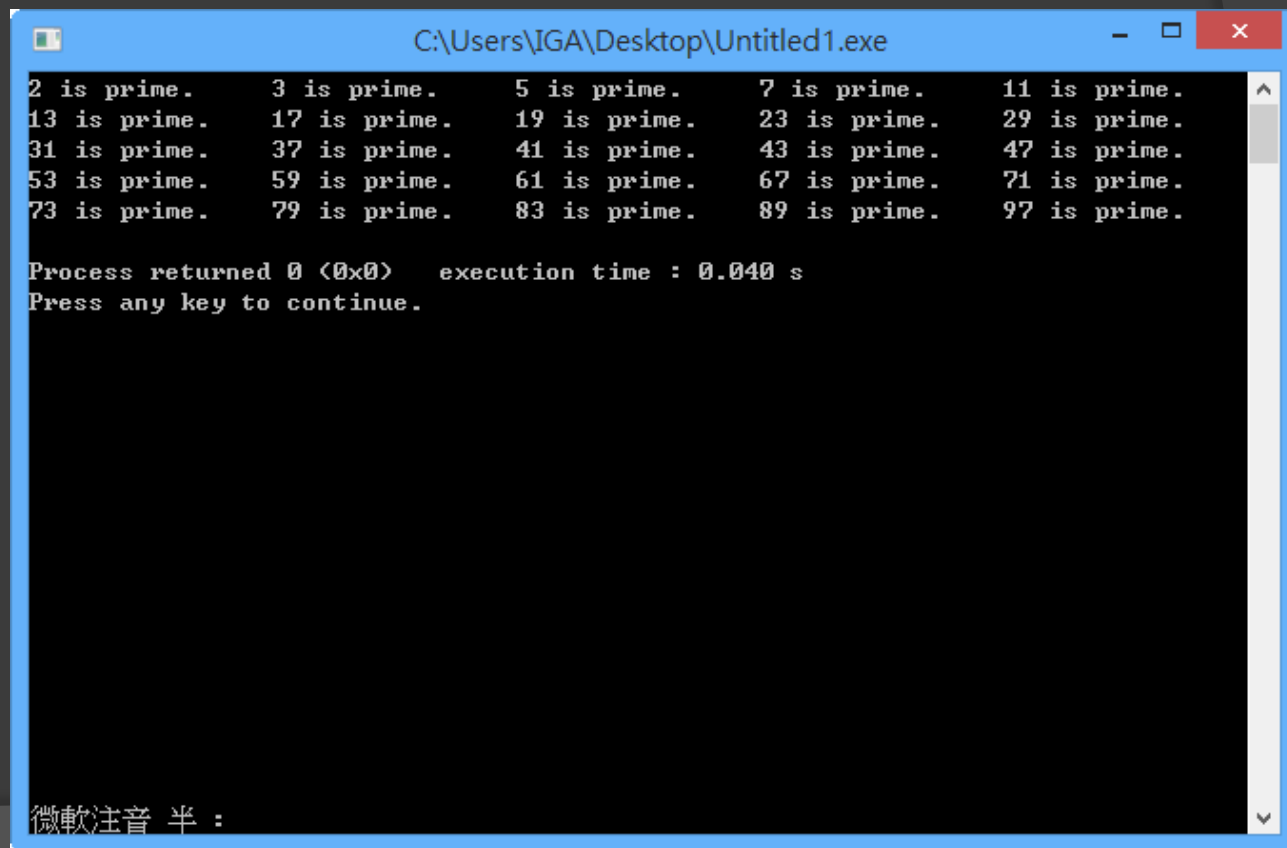
```
*  
**  
***  
****  
*****  
  
Process returned 0 (0x0) execution time : 0.041 s  
Press any key to continue.
```

break & continue

- ◎ break：程式中斷，跳出所在迴圈或 Switch
- ◎ continue：回到最近的迴圈的程式開頭

綜合練習

- 印出1~100間所有質數。
- 提示：for迴圈搭配if判斷式。



A screenshot of a Windows command prompt window titled "C:\Users\IGA\Desktop\Untitled1.exe". The window displays a list of prime numbers from 2 to 97, arranged in five columns. Below the list, it shows "Process returned 0 (0x0) execution time : 0.040 s" and "Press any key to continue.". At the bottom left of the window, there is a small text label "微軟注音 半 :".

```
C:\Users\IGA\Desktop\Untitled1.exe

2 is prime.    3 is prime.    5 is prime.    7 is prime.    11 is prime.
13 is prime.   17 is prime.   19 is prime.   23 is prime.   29 is prime.
31 is prime.   37 is prime.   41 is prime.   43 is prime.   47 is prime.
53 is prime.   59 is prime.   61 is prime.   67 is prime.   71 is prime.
73 is prime.   79 is prime.   83 is prime.   89 is prime.   97 is prime.

Process returned 0 (0x0)   execution time : 0.040 s
Press any key to continue.

微軟注音 半 :
```