

戴宏穎

0988639320

林致民

0928240312

王佩心

0910920142

複習

cin

```
std::cin>>a;
```

cout

```
std::cout<<"Hello!";
```

複習(cont)

加減乘除： $+$, $-$, $*$, $/$

取餘數： $\%$

小括號： $()$

複習(cont)

屬性(型態)

int, double, char, etc

ex: $20/3$ 輸出結果是 6

複習(cont)

宣告變數並給予初始值

```
int number = 0;
```

複習 (cont)

關係判斷符號			
小於	<	大於	>
小於等於	<=	大於等於	>=
相等	==	不相等	!=

複習(cont)

"="並不是真的等於,而是把等號右邊的值放到等號左邊

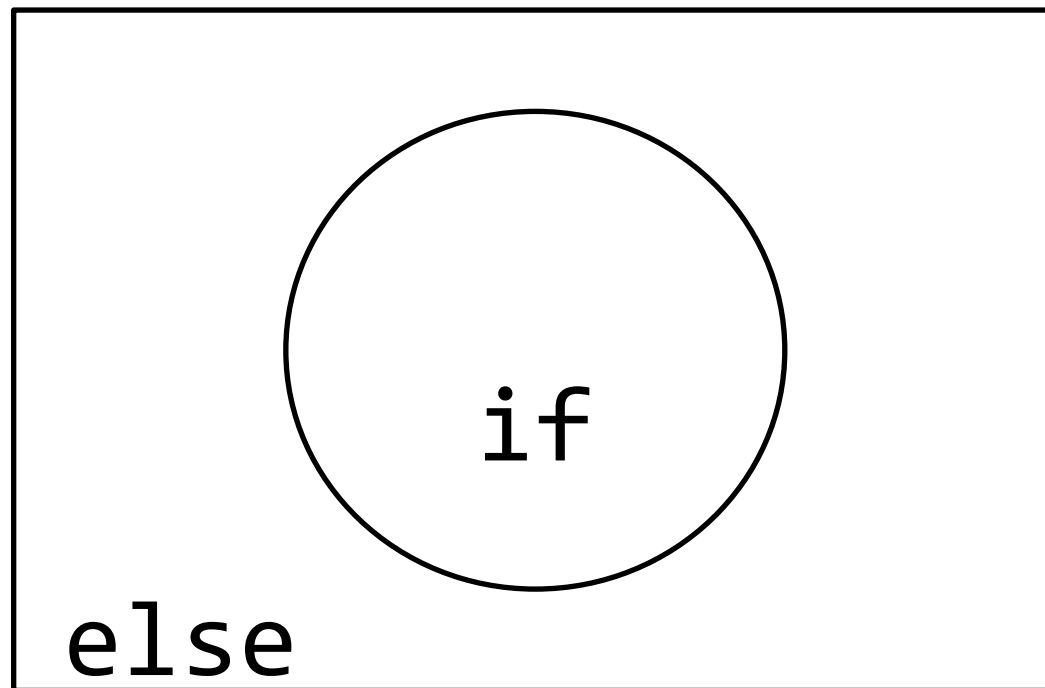
Ex: `number = number + 100;`

`a = 5`

`a == 5`

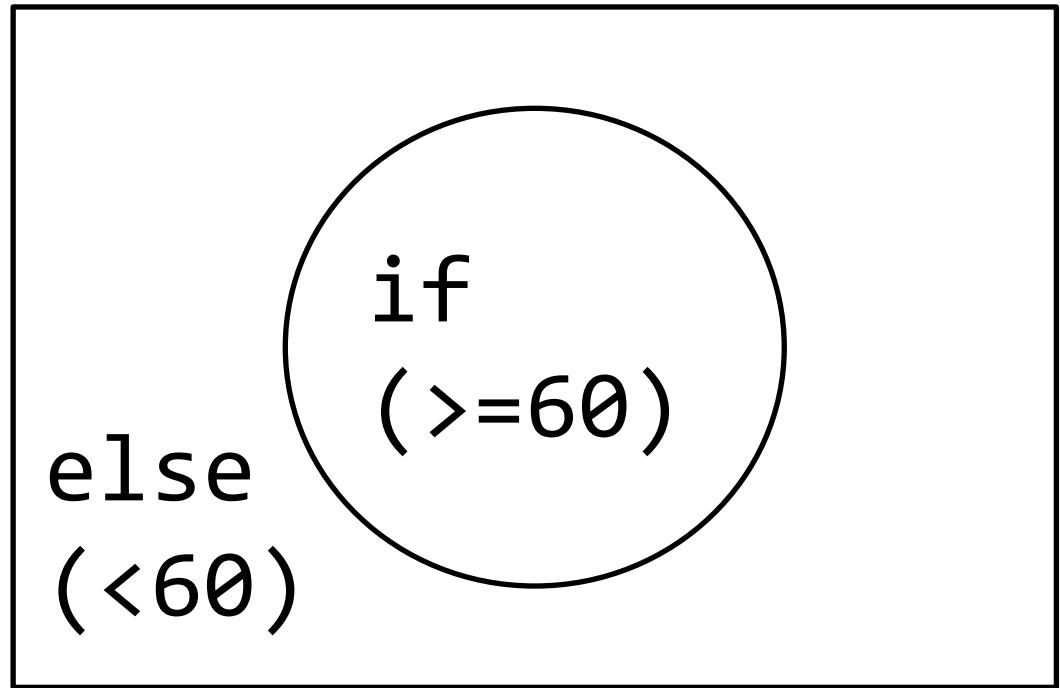
今日課程


```
if (判斷條件){  
    要做的事(指令);  
}  
else{  
    要做的事(指令);  
}
```



最外面的矩形是全部的集合，
圓圈內是**if**成立的集合，
圓圈外是**if**不成立(即**else**)的集合

全部的集合
(整數)



Ex: 輸入一個整數, 如果 ≥ 60 輸出 "Good!",
如果 < 60 則輸出 "Keep going!"

怎麼轉換成程式碼？

```
int num;
```

```
std::cin>>num;
```

```
if (???) {
```

```
    std::cout<<"Good!";
```

```
}
```

```
else {
```

```
    std::cout<<"Keep going!";
```

```
}
```

```
int num;  
std::cin>>num;  
if (num>=60){  
    std::cout<<"Good!";  
}  
else{  
    std::cout<<"Keep going!";  
}
```

Example

Input: 一個整數

Output: (某整數) is an integer.

Example(cont)

```
#include <iostream>
int main(){
    int a;
    if (std::cin>>a){
        std::cout<<a<<" is an integer."
        <<std::endl;
    }
    return 0;
}
```

Example(cont)

3

3 is an integer.

Process exited after 1.992 seconds with return value 0
請按任意鍵繼續 . . . _

Example(cont)

如果輸入的不是整數呢？

a

```
-----  
Process exited after 2.023 seconds with return value 0  
請按任意鍵繼續 . . .
```

Coding...

剛才的題目,現在自己動手打一遍~

Input: 一個整數

Output: (某整數) is an integer.

Practice(#208)

題目說明：**1-3月**是春天，**4-6月**是夏天，
7-9月是秋天，**10-12月**是冬天

Input

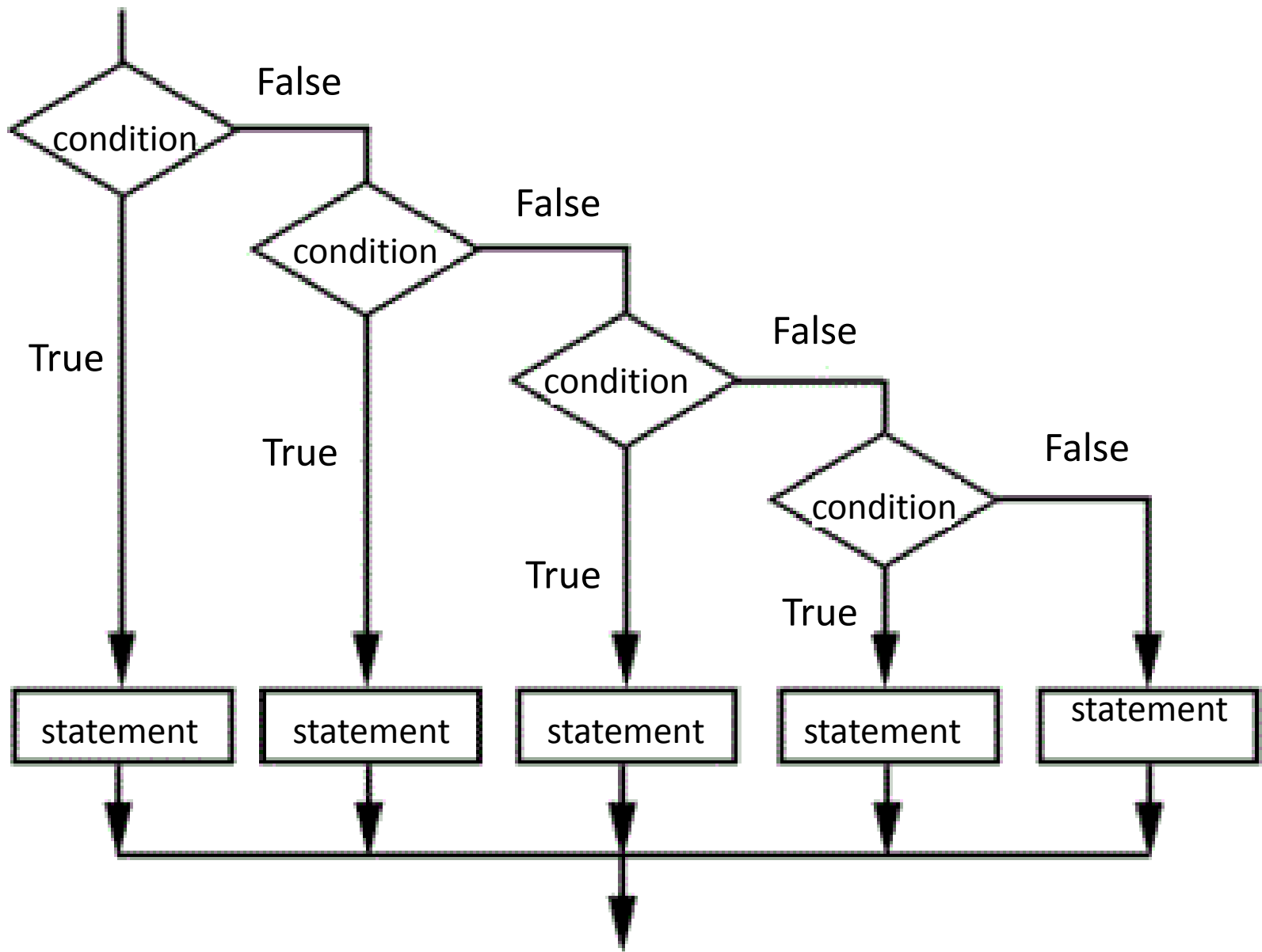
一個整數(**1-12**)，代表月分

Output

這個月是哪個季節(春/夏/秋/冬)，
結尾要記得換行

另一種寫法

```
if (month>9){  
    std::cout<<"winter"<<std::endl;  
}  
else if (month>6){  
    std::cout<<"fall"<<std::endl;  
}  
else if (month>3){  
    std::cout<<"summer"<<std::endl;  
}  
else{  
    std::cout<<"spring"<<std::endl;  
}
```

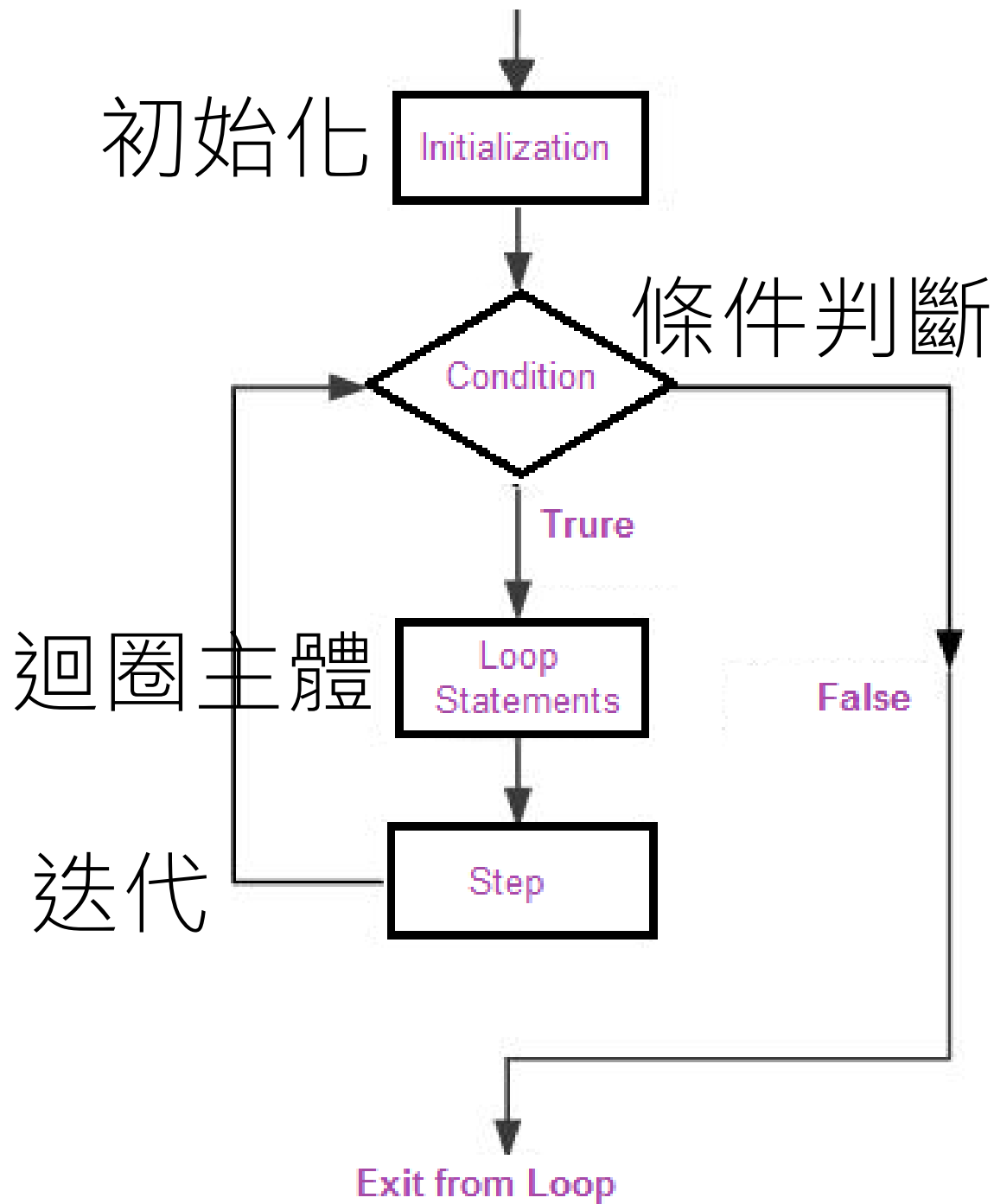


```
if (判斷條件1){  
    要做的事(指令);  
}  
else if (判斷條件2){  
    要做的事(指令);  
}  
...  
else {  
    要做的事(指令);  
}
```

for迴圈介紹

什麼是迴圈？

- **Loop**(迴圈)
- 當需要重複做同一件事很多次的時候
- 什麼時候停止？
- 符合某些條件的時候！
- **Ex：** 印出數字**1~100**在螢幕上
重複做的事：印出數字
什麼時候停：印了**100**次之後



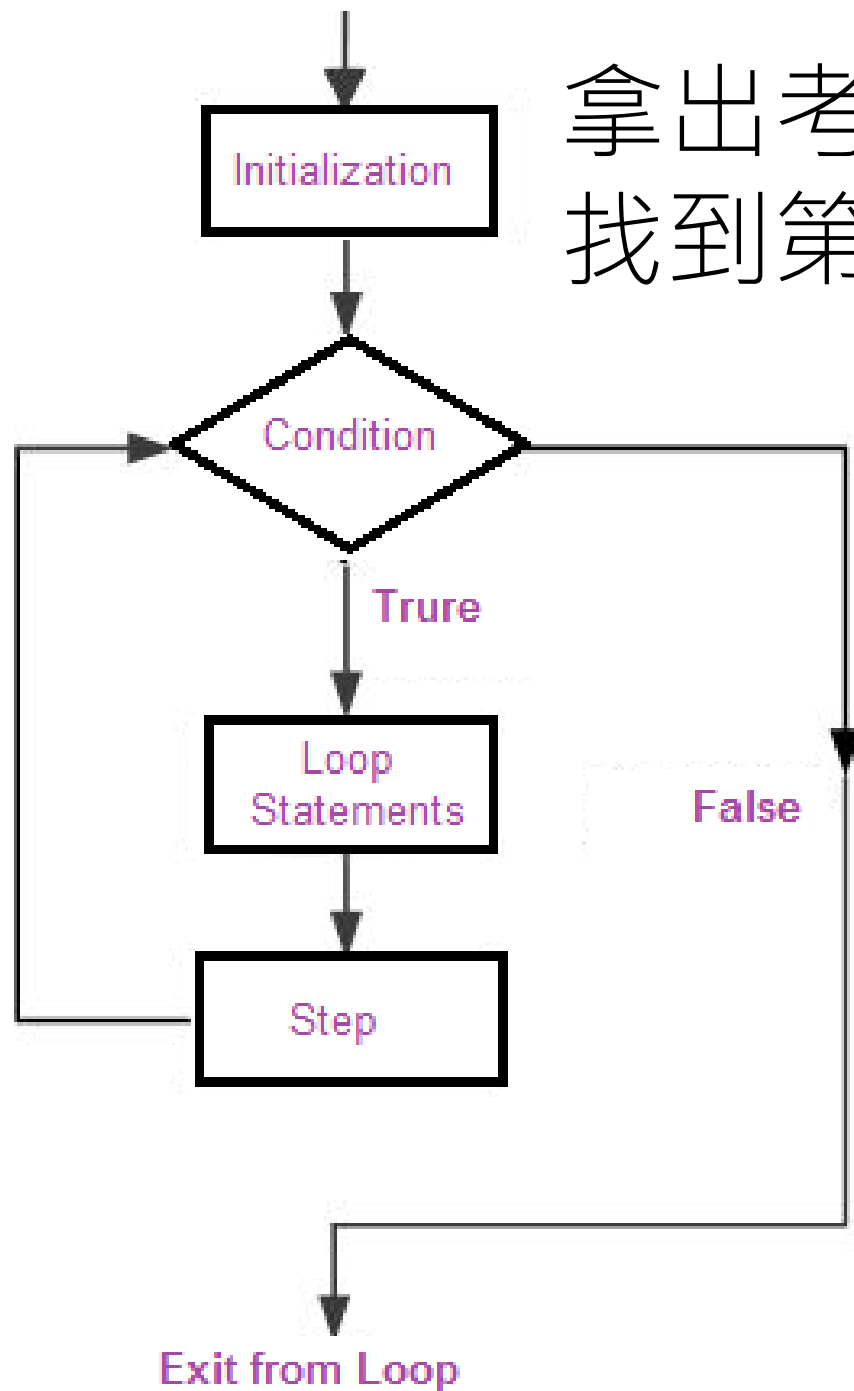
跟**if**的不同之處

可以一直重複執行

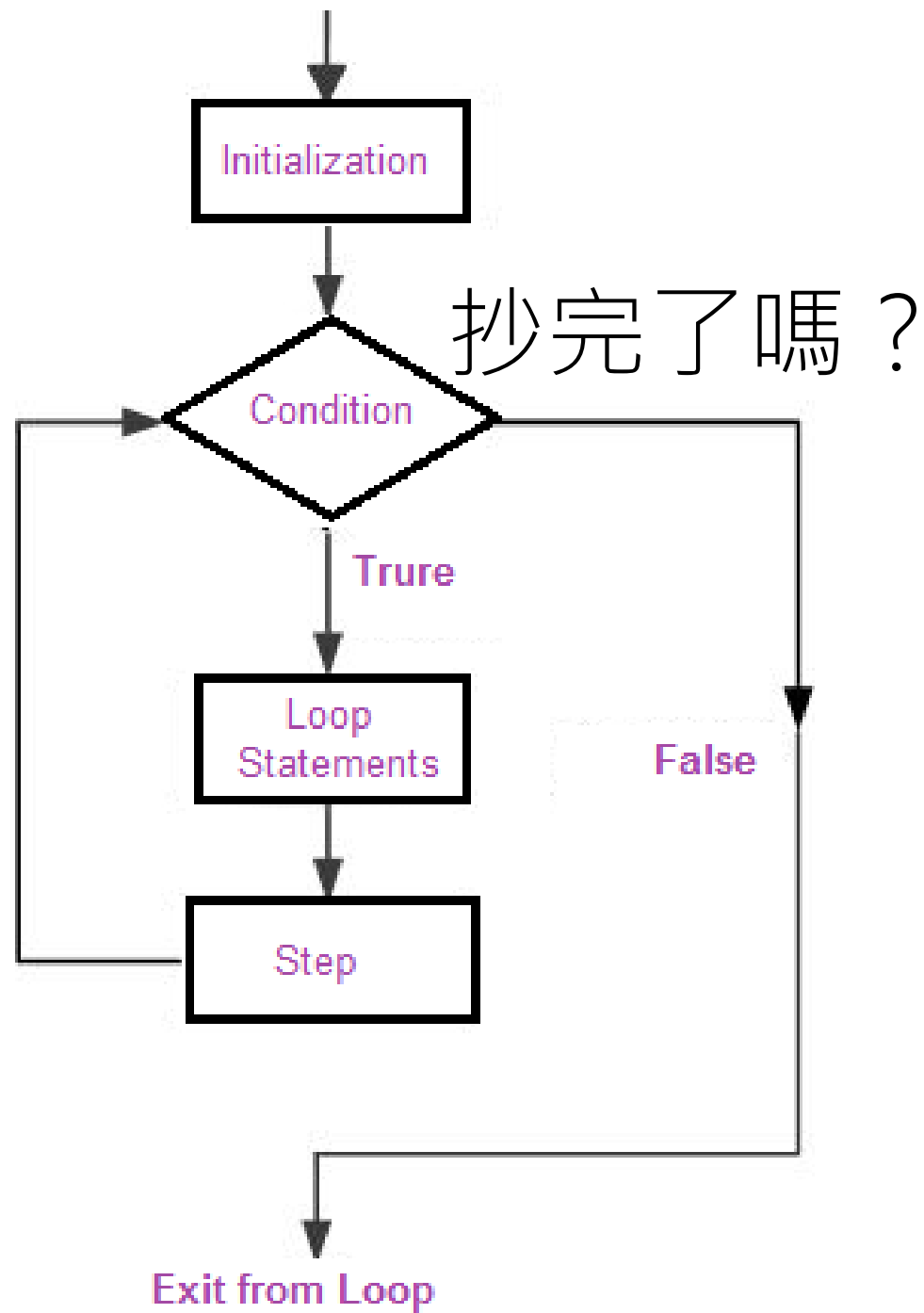
可以設定初始值

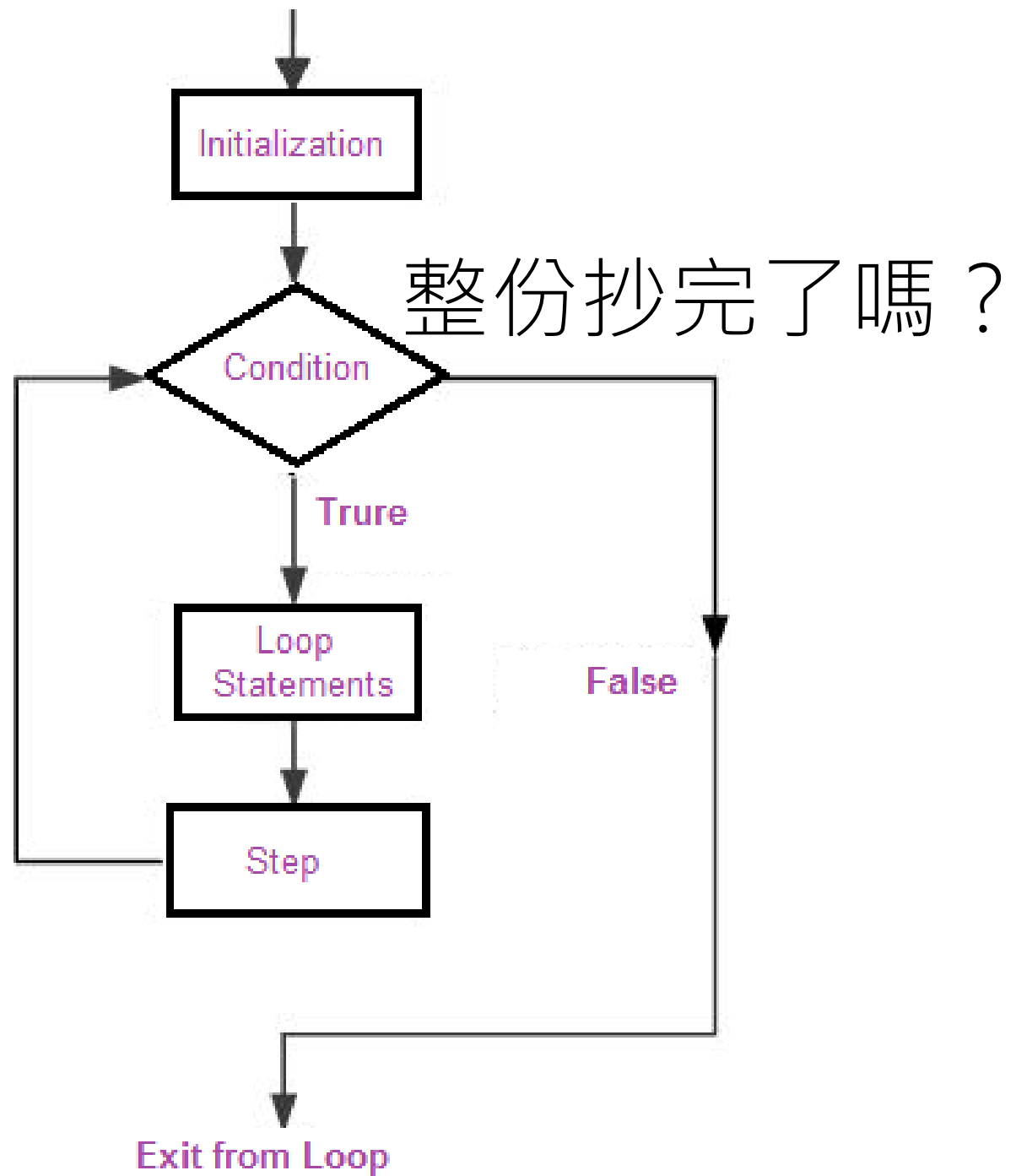
每次做完後還會更新變數值

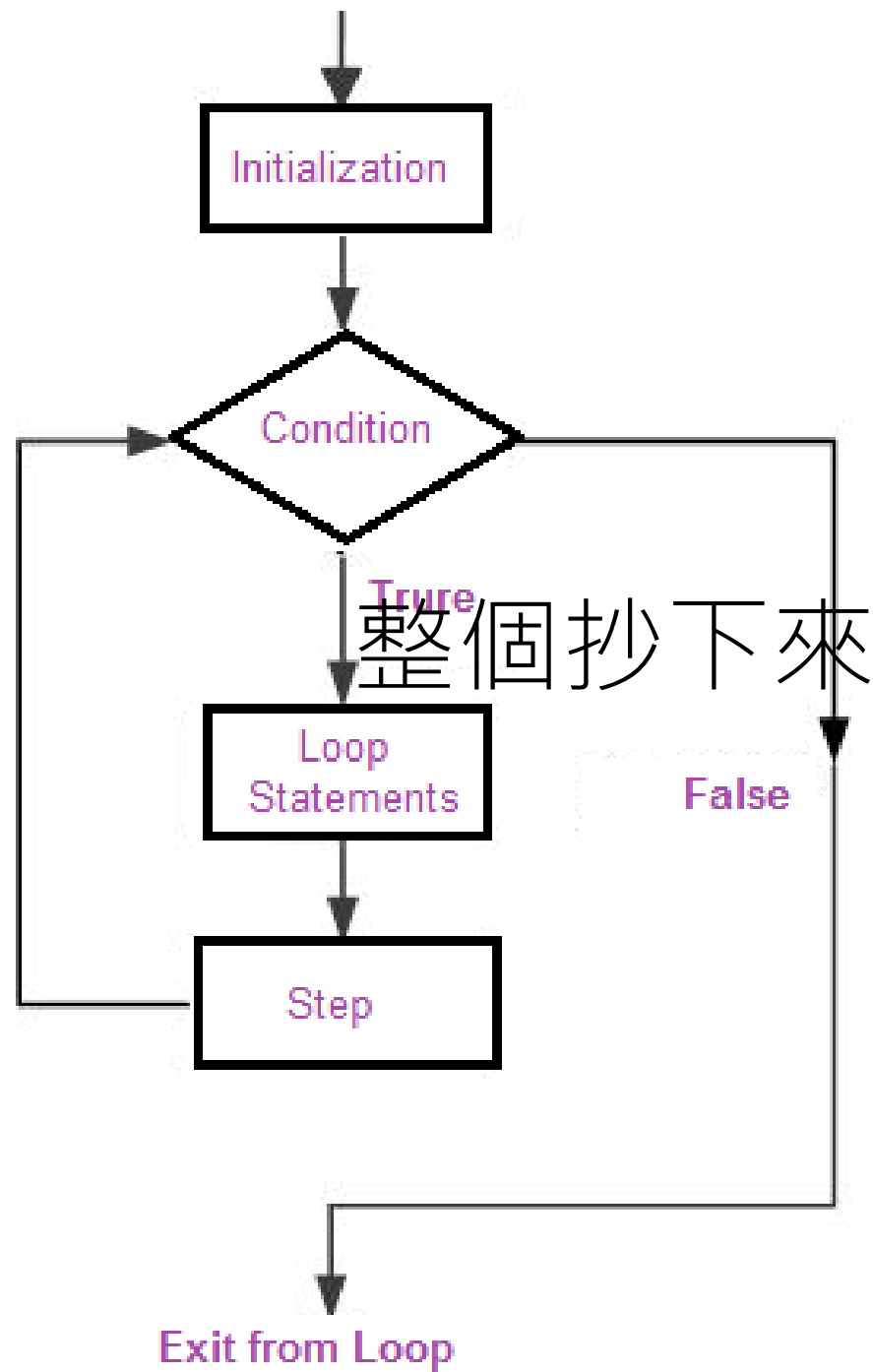
Example:
抄考卷

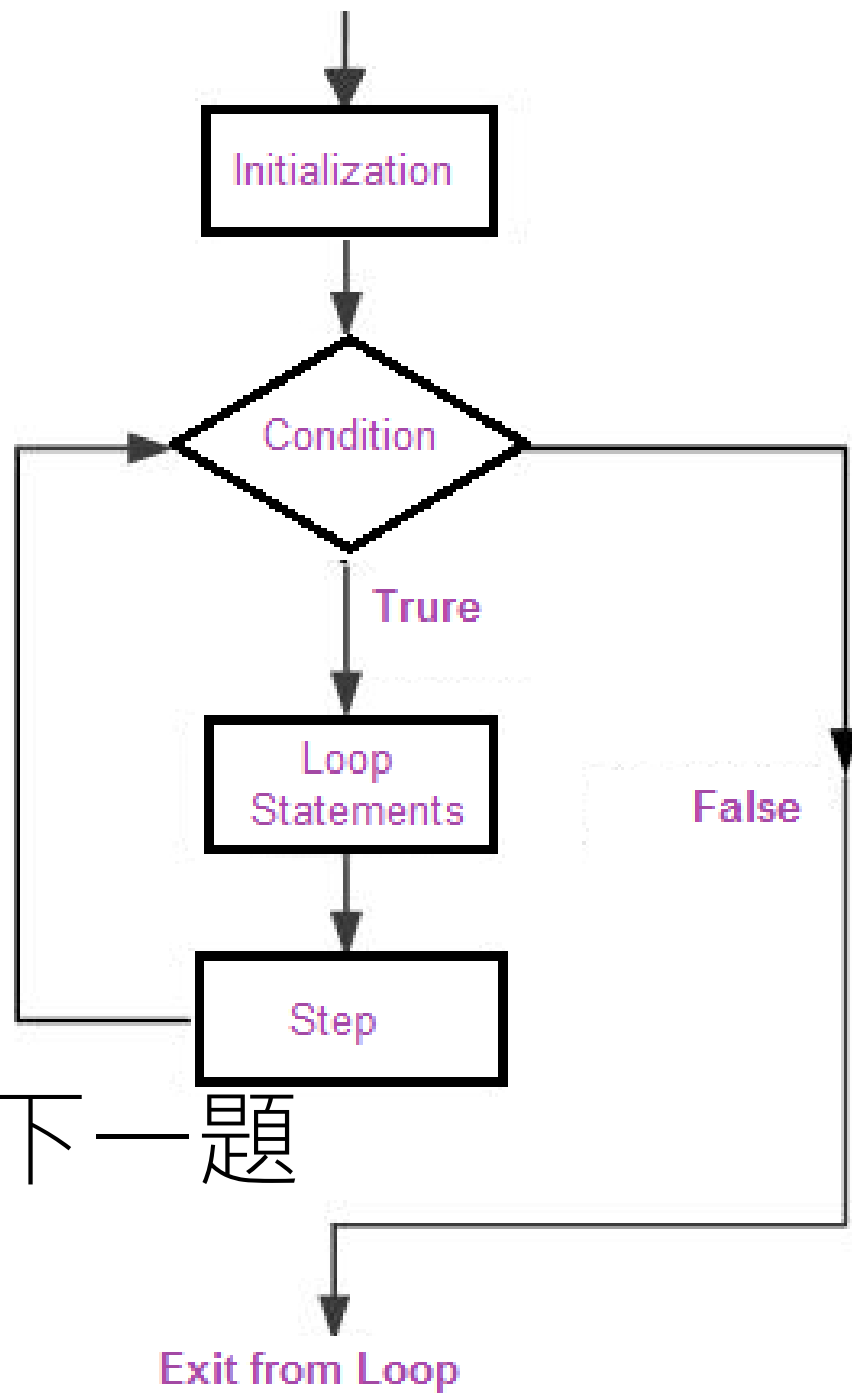


拿出考卷
找到第一格

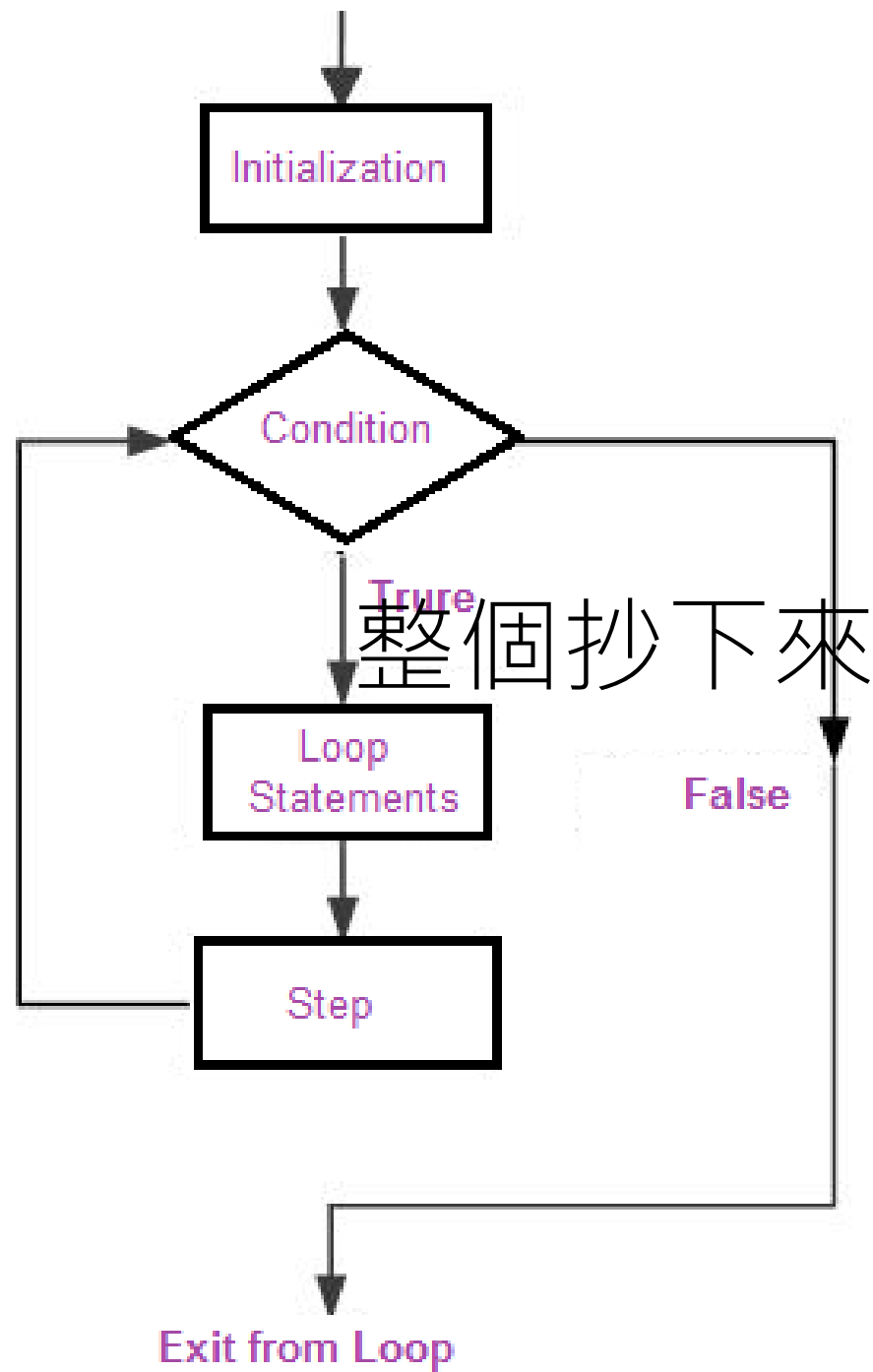


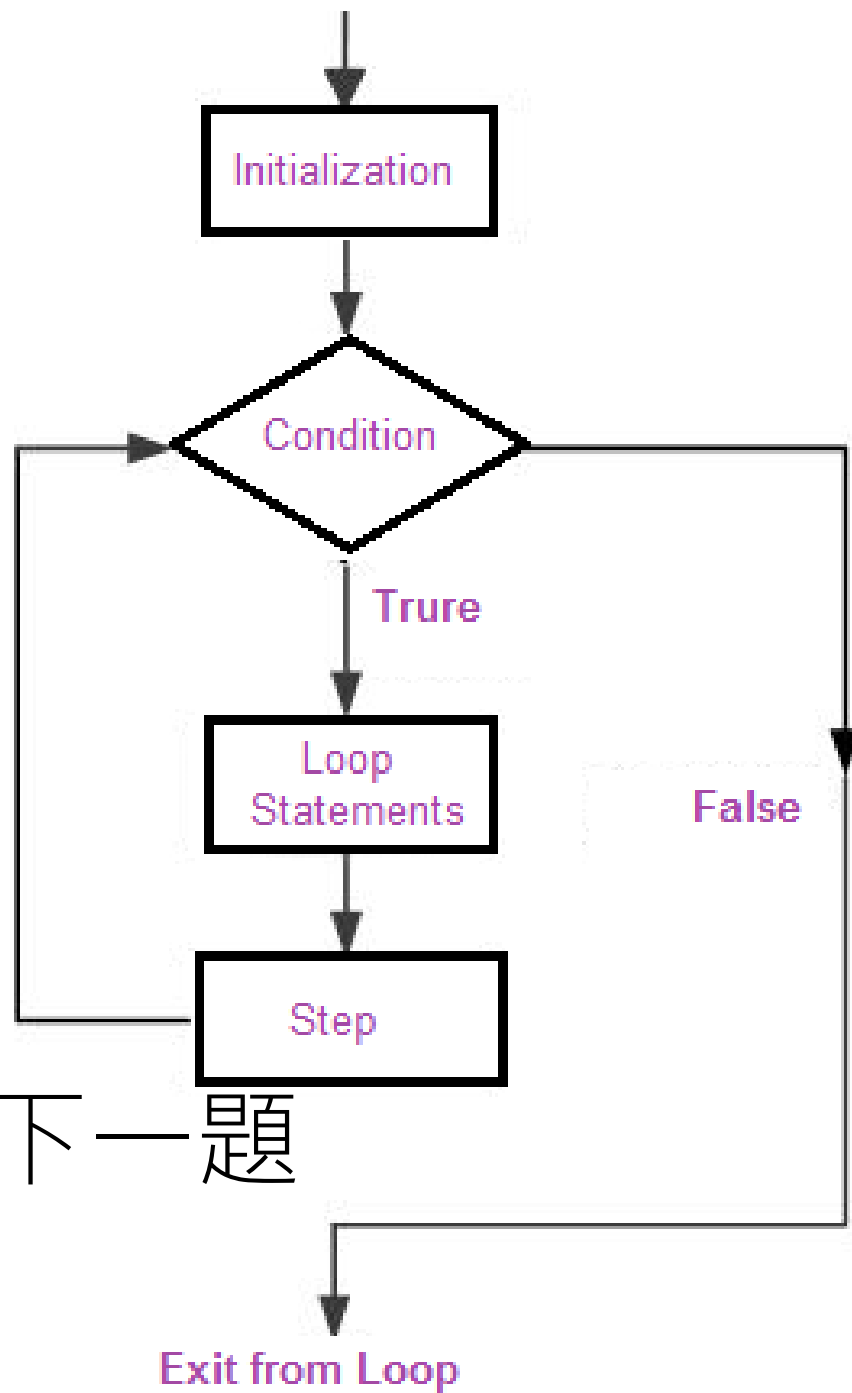




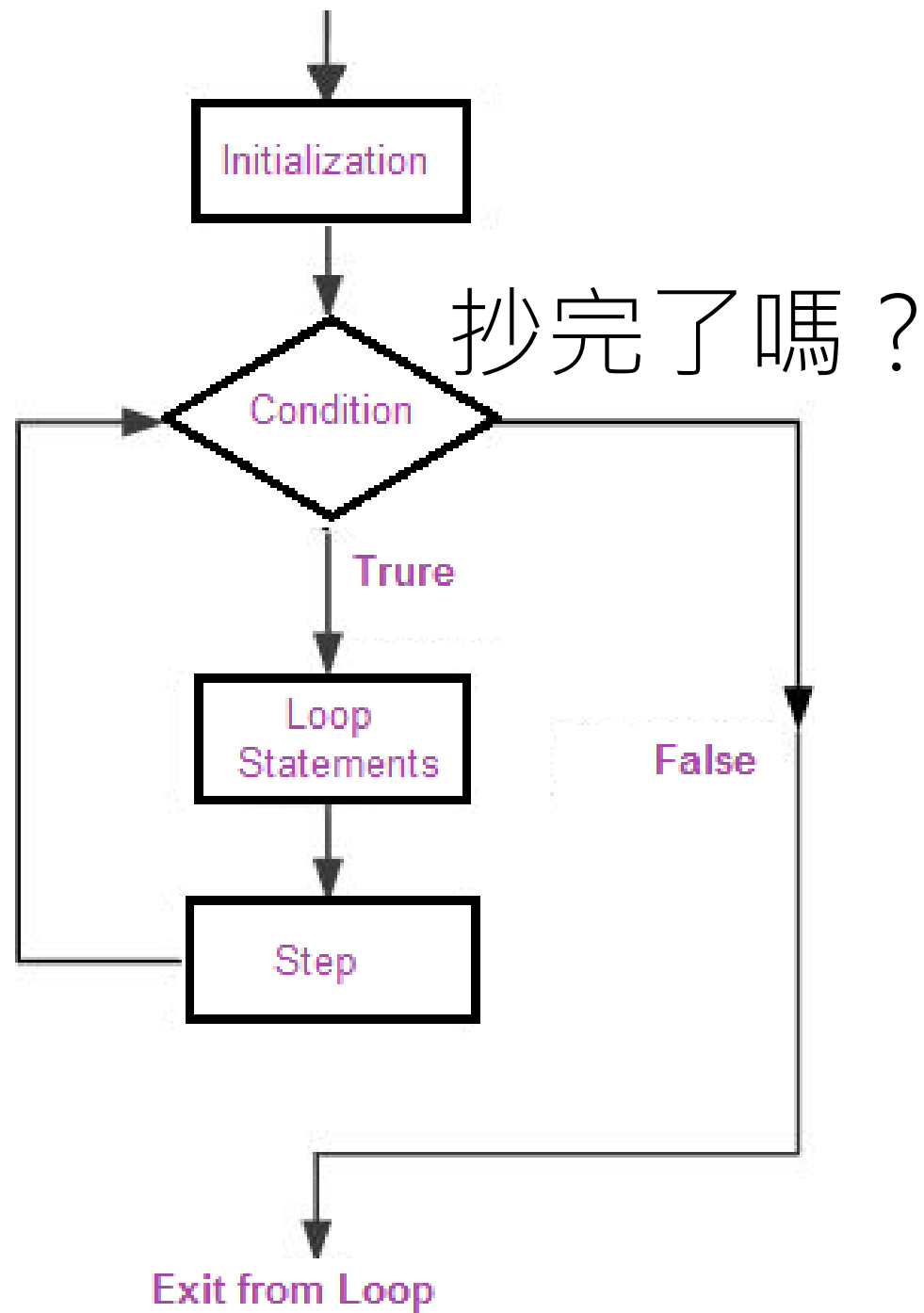


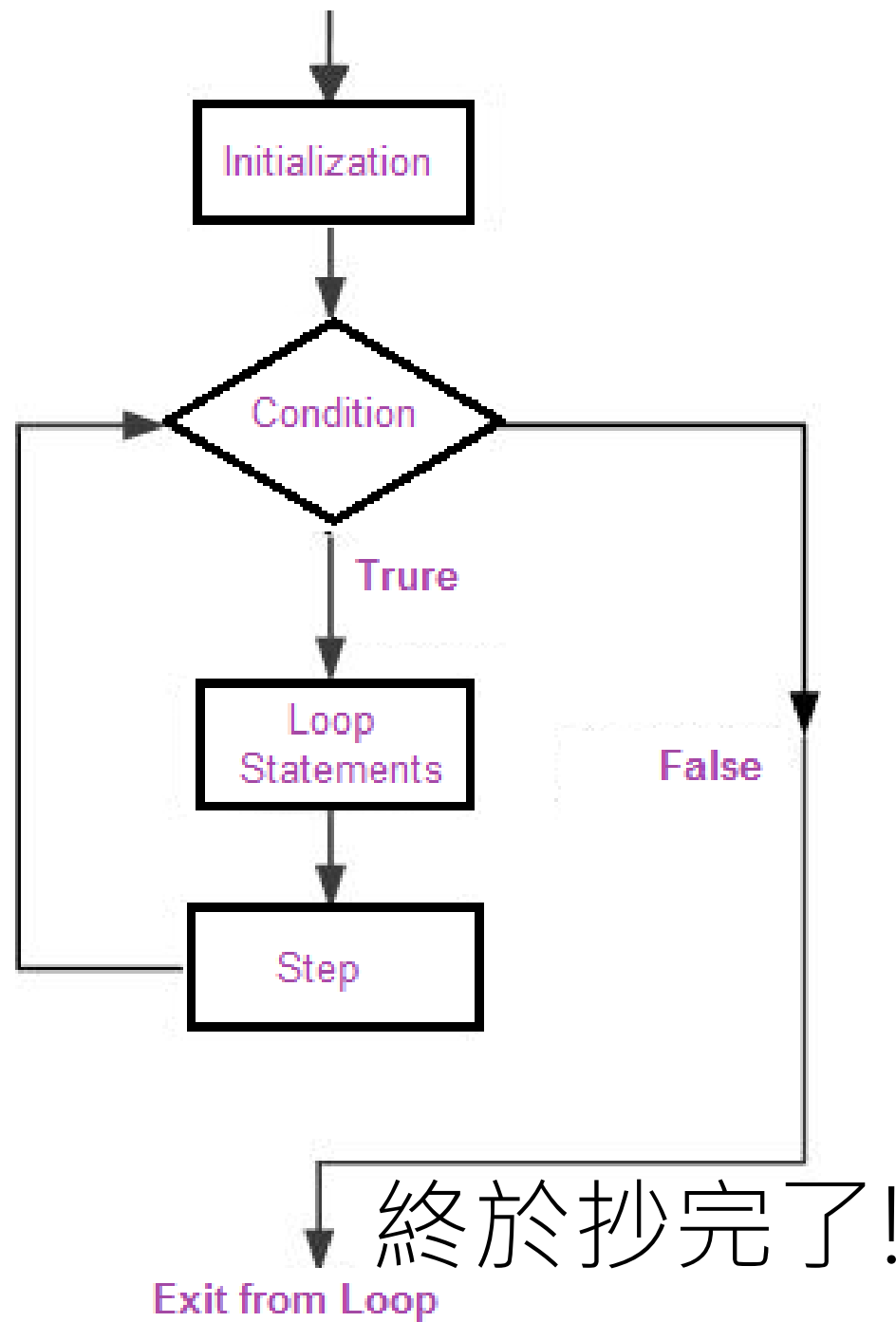
看他的下一題





看他的下一題





怎麼轉換成程式碼？

先來看看**for**的主體架構...

```
for(初始化;判斷是否繼續執行的條件;  
更新變數值){  
    要做的事(指令);  
}
```

怎麼轉換成程式碼?(cont)

```
for( 初始值=???; 判斷條件=???; 更新=??? )  
{  
    要做的事情=???;  
}
```


怎麼轉換成程式碼?(cont)

```
for(拿出考卷找第一題;抄完了沒?;抄下一  
題)  
{  
    整個抄下來;  
}
```

Example

- 印出1~10

```
for (i = 1; i <= 10; i++)  
    std::cout << i << " ";
```

Example

- 印出1~10

```
for (i = 1; i <= 10; i++)  
    std::cout << i << " ";
```

Example

- 印出1~10

```
for (i = 1; i <= 10; )  
    std::cout << << " " ;
```

Example

- 印出1~10

```
for (i = 1; i <= 10; i++)  
    std::cout << << " " ;
```

Example

- 印出1~10

```
for (i = 1; i <= 10; i++)  
    std::cout << i << " ";
```

Practice

輸出 $1+2+3+\dots+10$ 的結果

Practice(cont)

你可以.....

```
std::cout<<1+2+3+...+10;
```

也可以用**for**迴圈

Practice(cont)

在打**code**之前,想想我們需要什麼?

- 一個變數儲存每次輸入的值

i = i + i; (?)

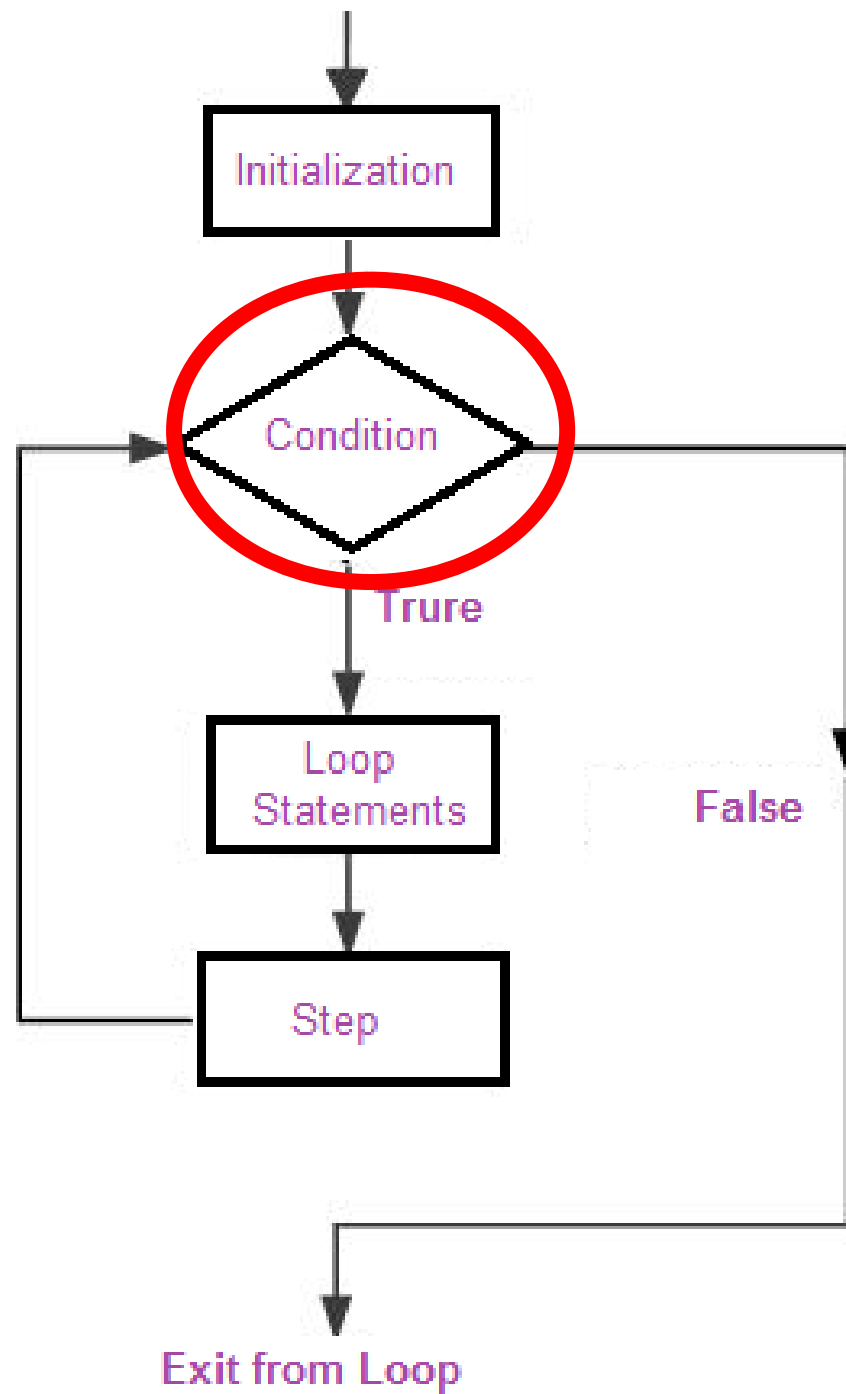
- 另一個變數負責做累加的動作

Practice(cont)

```
sum = 0;
for (int i=1; i<11; i++){
    sum = sum + i;
}
std::cout << sum << std::endl;
```

Practice(#204)

迷思：迴圈最少跑一次嗎？



Of course..... NOT

```
for (int i = 1; i < 1; i++)  
{  
    std::cout << "hello";  
}
```

巢狀回圈

巢 + 回圈 = ?



```
for(int i = 0; i < 100; ++i) {  
    std::cout << "Hello!" << std::endl;  
}
```



眉頭一皺... 發現案情不太單純...

就只是回圈中

包著另外一個回圈而已啦

看Code!!!!!!

```
1 #include <iostream>
2
3 int main(){
4
5     for(int i = 0; i < 10; i++){
6         for(int j = 0; j < 10; j++){
7             std::cout << "Hello!" << std::endl;
8         }
9     }
10    return 0;
11 }
```



提問：這樣Hello!會印幾次？

```
1 #include <iostream>
2
3 int main(){
4
5     for(int i = 0; i < 10; i++){
6         for(int j = 0; j < 10; j++){
7             std::cout << "Hello!" << std::endl;
8         }
9     }
10    return 0;
11 }
```



提問：這樣Hello!會印幾次？

```
1 #include <iostream>
2
3 int main(){
4
5     for(int i = 0; i < 10; i++){
6         for(int j = 0; j < 10; j++){
7             std::cout << "Hello!" << std::endl;
8         }
9     }
10    return 0;
11 }
```

答案：第一層跑十次
每一層又都再跑十次

提問：這樣Hello!會印幾次？

```
1 #include <iostream>
2
3 int main(){
4
5     for(int i = 0; i < 10; i++){
6         for(int j = 0; j < 10; j++){
7             std::cout << "Hello!" << std::endl;
8         }
9     }
10    return 0;
11 }
```

我是數學王：10 X 10 = 100次

如何辦到的？黑魔法？

```
1 #include <iostream>
2
3 int main(){
4
5     for(int i = 0; i < 10; i++){
6         for(int j = 0; j < 10; j++){
7             std::cout << "Hello!" << std::endl;
8         }
9     }
10    return 0;
11 }
```



```

1 #include <iostream>
2
3 int main(){
4
5     for(int i = 0; i < 10; i++){
6         for(int j = 0; j < 10; j++){
7             std::cout << "Hello!" << std::endl;
8         }
9     }
10    return 0;
11 }

```

= 0	i = 9, j = 0.....
= 1	i = 9, j = 1.....
⋮	⋮.....
= 8	i = 9, j = 8.....
= 9	i = 9, j = 9.....

乖，印出來看看

```
1 #include <iostream>
2
3 int main(){
4
5     for(int i = 0; i < 10; i++){
6         for(int j = 0; j < 10; j++){
7             std::cout << i << " " << j << std::endl;
8         }
9     }
10    return 0;
11 }
```



兩層可以... 那三層呢？

```
1 #include <iostream>
2
3 int main(){
4
5     for(int i = 0; i < 10; i++){
6         for(int j = 0; j < 10; j++){
7             for(int k = 0; k < 10; k++){
8                 std::cout << i << " " << j << " " << k << std::endl;
9             }
10        }
11    }
12    return 0;
13 }
```



```
1 #include <iostream>
2
3 int main(){
4     for(int i = 0; i < 10; i++){
5         for(int j = 0; j < 10; j++){
6             for(int k = 0; k < 10; k++){
7                 for(int l = 0; l < 10; l++){
8                     std::cout << i << " " << j << " " << k << " " << l << std::endl;
9                 }
10            }
11        }
12    }
13 }
14 return 0;
15 }
```



五層... 別鬧了，自己回家玩

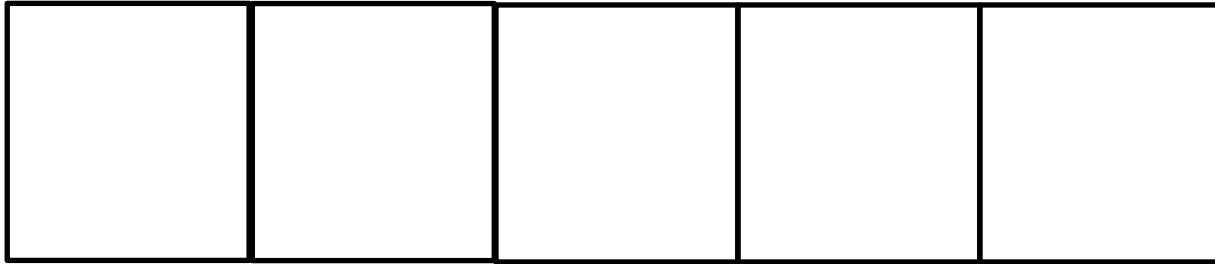
哈哈
你看看你!!



簡單介紹陣列

陣列的編號

array



0

1

2

3

4

陣列的編號

array

2	4	6	8	10
0	1	2	3	4

陣列的宣告

```
int array[] = {2,4,6,8,10};
```

```
int array[5] = {2,4,6,8,10};
```

初始化陣列

```
int array[5] = {0};
```

陣列的編號

```
array[0] = 2;   array[3] = 8;  
array[1] = 4;   array[4] = 10;  
array[2] = 6;
```


陣列的編號

array

2	4	6	8	10
---	---	---	---	----

0 1 2 3 4

```
array[0] = 2;    array[3] = 8;  
array[1] = 4;    array[4] = 10;  
array[2] = 6;
```

陣列搭配for迴圈

Ex： 數學老師想記錄全班**35**位同學的成績

可以怎麼做？

陣列搭配for迴圈(cont)

你可以宣告35個整數變數

```
int grade1, grade2, ...,  
grade35;
```

```
std::cin>>grade1>>grade2>>...  
>>grade35;
```

陣列搭配for迴圈(cont)

或是.....

宣告一個陣列！

```
int grade[35] = {0};
```

```
std::cin>>grade[0]>>grade[1]
```

```
>>...>>grade[34];
```

陣列搭配for迴圈(cont)

再搭配for迴圈~

```
int grade[35] = {0};  
for (i=0; i<35; i++){  
    std::cin>>grade[i];  
}
```

Practice(練習用陣列搭配for)

輸入5個整數後,倒序輸出

Sample input:

5 6 7 8 9

Sample output:

9 8 7 6 5

Practice(cont)

```
#include <iostream>
int main(){
    int num[5] = {0};
    for (int i=0; i<5; i++){
        std::cin>>num[i];
    }
    for(int i=4; i>-1; i--){
        std::cout<<num[i]<<" ";
    }
    return 0;
}
```

Practice(#209)

Homework

上課練習題 #208 #204 #209

(若於課堂中未AC, 要回家繼續練習至AC)

& #206

Bonus: #210 #211 #212