迴圈結構

while

這個函式跟for很像，但是更常用在**不知道/不需要迴圈次數**或是**無限迴圈**上

架構解釋

```
while (判斷式){
    迴圈內容
}
```

```cpp
int i=0;
while (i<5){
    cout << "i(while): " << i << "\n";
    i++;
}

for (int i=0 ; i<5 ; i++){
    cout << "i(for): " << i << "\n";
}
```

```
i(while): 0
i(while): 1
i(while): 2
i(while): 3
i(while): 4
i(for): 0
i(for): 1
i(for): 2
i(for): 3
i(for): 4
```

可以發現**while加上變數就和for一樣**，換句話說

while就是沒有變數的for

常見用途

- 需要維護的變數過多
- 不定量輸入
- task數

需要維護的變數過多

```
1 for (int i=0, j=0, k=0 ; i<10||j<30||k<69 ; i++, j+=2, k+=
2     cout << i << " " << j << " " << k << "\n";
3 }
```

```
 1 int i=0, j=0, k=0;
 2 while (1){
 3     if (sth){
 4         i++;
 5     }
 6     if (sth){
 7         j+=2;
 8     }
 9     if (sth){
10         k+=3;
11     }
12 }
```

不定量輸入

cin其實會回傳值，如果輸入成功就回傳true，否則是false

```
1 int i;
2 while (cin >> i){
3     cout << "i: " << i << "\n";
4 }
```

task數

通常在寫題目的時候，大多數會給一個t代表題目要求的任務(task)數量

這時候用while可以寫出很簡潔的程式碼

```
1 int t=5; // 有五個task
2 while (k--){
3     sth.
4 }
```

k實際是從5慢慢減到1（總共跑五次），當k等於0時，程式會認為這是false

根據while的語法，這時候迴圈就會停止

⚠️ 以上的東西務必要會，未來的題目不再只是只有單個測資

例題

例題1
例題2

continue & break

有時候我們不希望迴圈一直跑下去，這時候可以怎麼做呢?

C++提供了兩個函式來**控制迴圈**

continue

當碰到continue的時候，就會直接跑到迴圈的最上面

break

當碰到break的時候，就會直接結束當前迴圈