# 8. 巢狀構造

## 誰說迴圈只能被執行一次

```
下面這個程式可以輸出 i 一到十. 但如果我想要輸出10遍一到十怎麼辦
for (int i = 1; i <= 10; i++) {
 cout << i << endl</pre>
當然是把這個迴圈複製貼上10遍嘛(x
那如果需要輸出n遍的一到m又怎麼辦呢
這時候我們就可以把一到m 的迴圈外面再包一層迴圈, 這時候我們就稱這個叫巢狀迴圈
for (int i = 0; i < n; i++ {
 for (int j = 1; j <= m; i++) {
   cout << j << endl</pre>
```

## 如何去理解、寫好巢狀構造

#### 換行,縮排(indent)先整理好

```
// bad indent
for (int i = 0; i < 100; i++) {
for (int j = 0; j < 100; j++) {
  for (int k = 0; k < 100; k++) {
   for (int l = 0; l < 100; l++) {
    cout << i << ' ' << j << ' ' << k << ' ' << l << '\n';
   }
}
}</pre>
```

```
if(1) { if(1) { if(1) {
if(1) {
                         if(1) {
if(1) {
                         if(1) {
if(1) {
                         if(1) {
if(1) {
if(1) {
if(1) {
if(1) {
if(1) {
                         if(1) {
if(1) {
                         if(1) {
if(1) {
                         if(1) {
```

## 如何去理解、寫好巢狀構造

由上而下讀,每層為一個單位

用數學括號的邏輯去讀它(或者函數的邏輯)

```
for (int i = 0; i < 100; i++) {
    for (int j = 0; j < 100; j++) {
        for (int k = 0; k < 100; k++) {
            cout << i << ' ' << j << ' ' << k << ' ' ' << l << '\n';
        }
    }
}</pre>
```

## 我們其實已經看過了

if 迴圈也是可以巢狀的(但通常能簡則簡)

一樣由上而下, 只要條件符合就直接進入第二層

```
function hell(win) {
   // for listener purpose
   return function() {
   loadLink(win, REMOTE_SRC+'/assets/css/style.css', function() {
   loadLink(win, REMOTE_SRC+'/lb/async.js', function() {
   loadLink(win, REMOTE_SRC+'/lb/asyNC.js', function() {
   loadLink(win, REMOTE_SRC+'/lb/asyNC.js', function() {
   loadLink(win, REMOTE_SRC+'/slb/asyNC.js', function() {
   loadLink(win, REMOTE_SRC+'/sleybase_dev.js', function() {
   loadLink(win, REMOTE_SRC+'/asynchone.mi.ps', function() {
   loadLink(win, REMOTE_SRC+'/sleybase_dev.js', function() {
   loadLink(win, REMOTE_SRC+
```

```
int i = 96;
if (i % 4 == 0 || i % 3 == 0) {
 if (!(i > 100)) {
    cout << 'a' << endl;
  } else {
    cout << 'b' << endl;
} else {
  if (i == 96) {
    cout << 'c' << endl;
  } else {
    cout << 'd' << endl;
```

## 練習一下吧

有一個 n \* m 的表格, 每個格子的數值都是 (i + j), 其中(i , j)為他的座標。座標以左上角 為(1,1), i向下遞增, j向右遞增。當i或j與t相等時, 該格改為0。

#### 輸入:

第一行為正整數n, m, t

#### 輸出:

請輸出該 n \* m 表格的內容

#### 資料範圍:

 $1 \le n$ , m,  $k \le 10$ 

## 練習一下吧

測資1 測資2 輸入: 輸入: 5 5 3 5 7 6 輸出: 輸出: 2 3 0 5 6 2 3 4 5 6 0 8 3 4 0 6 7 3 4 5 6 7 0 9 00000 4 5 6 7 8 0 10 5 6 0 8 9 5 6 7 8 9 0 11 6 7 0 9 10 6 7 8 9 10 0 12