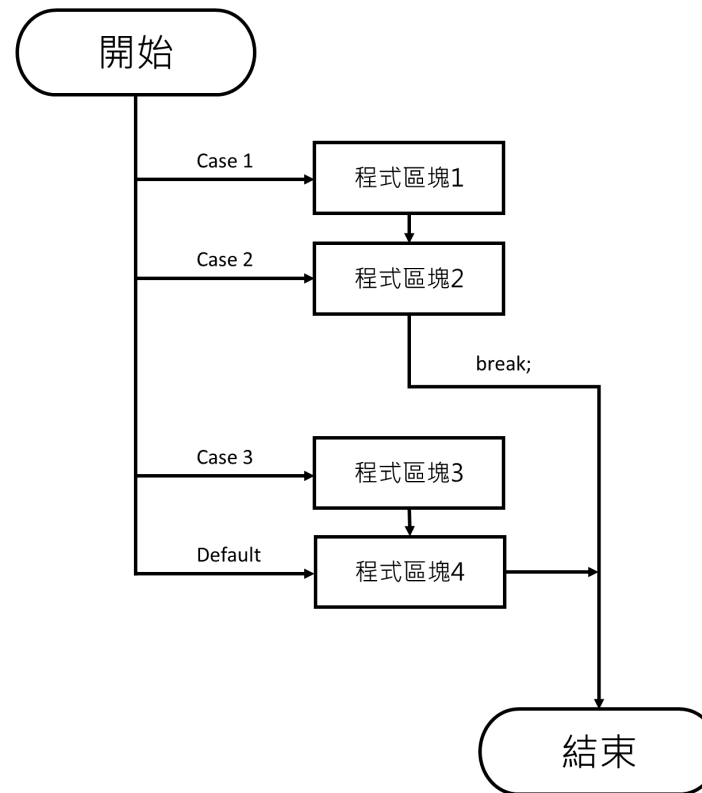
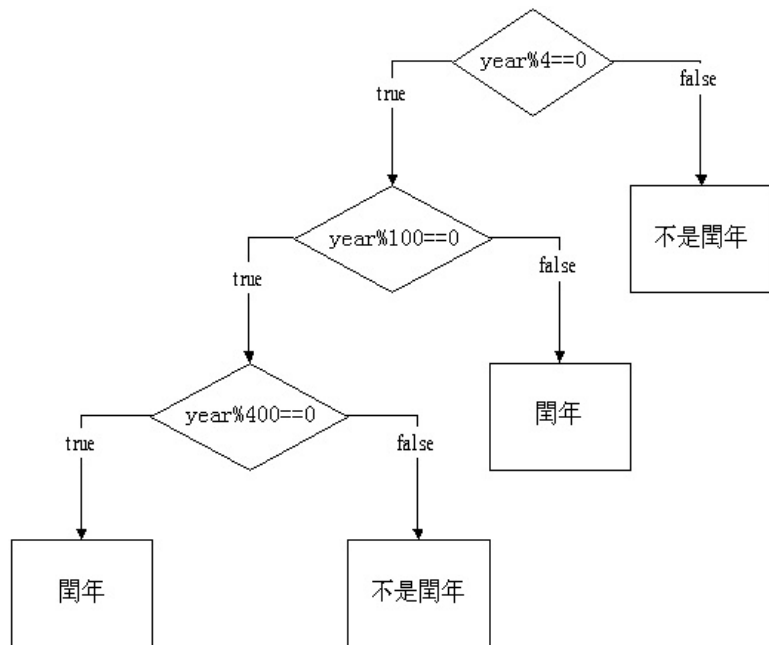


選擇 & 迴圈結構

switch

這個函數相對來說不太重要
稍微了解寫法跟概念就好



```
1 int main() {
2     cin >> n;
3     switch (n) {
4         case 5:
5             cout << "u get a 300$\n";
6         case 4:
7             cout << "u get a 200$\n";
8         case 3:
9             cout << "u get a 100$\n";
10            break;
11        default:
12            cout << "u get nothing :owhy:\n";
13            break;
14    }
15    return 0;
```

Input:

Copy

5

Expected Output:

Copy

Received Output:

Copy

u get a 300\$
u get a 200\$
u get a 100\$
total get: 600\$

^ Testcase 2 Failed 29ms



Input:

Copy

4

Expected Output:

Copy

Received Output:

Copy

u get a 200\$
u get a 100\$
total get: 300\$

Input:

Copy

3

Expected Output:

Copy

Received Output:

Copy

u get a 100\$
total get: 100\$

^ Testcase 4 Failed 35ms



Input:

Copy

2

Expected Output:

Copy

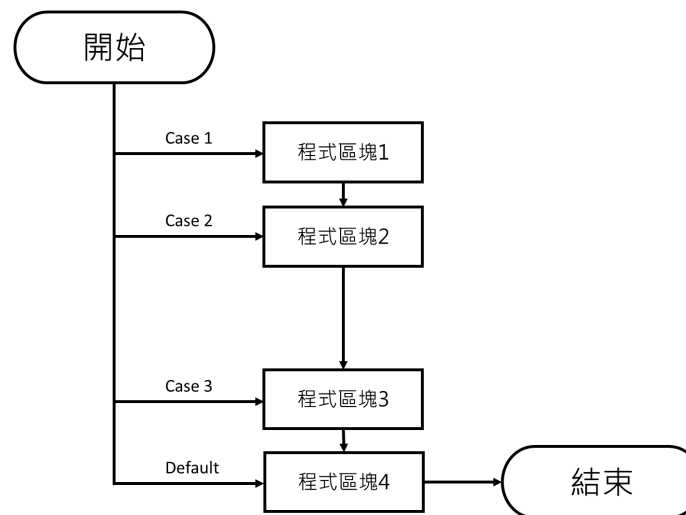
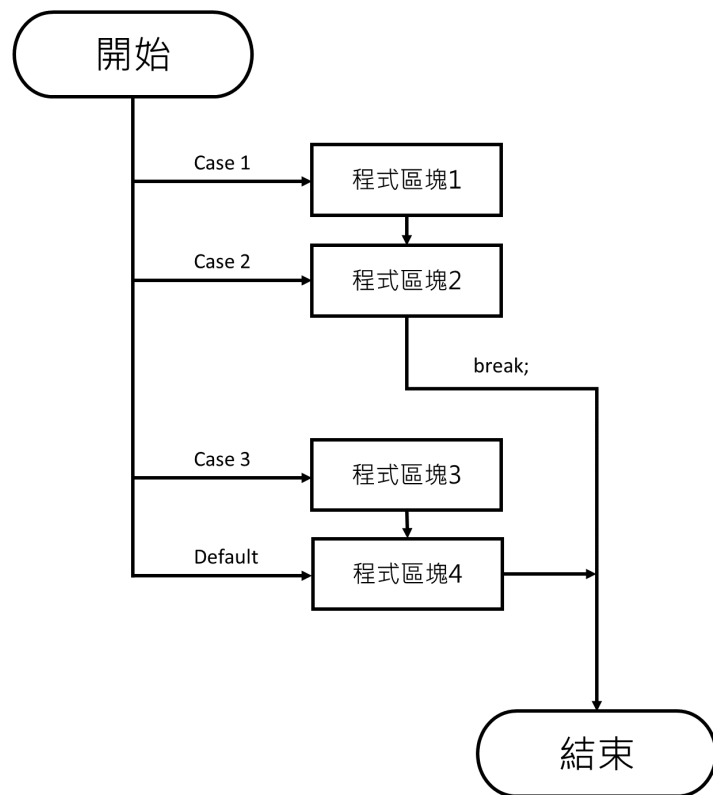
Received Output:

Copy

u get nothing :owhy:
total get: 0\$

架構解釋

```
1 switch (要比對的變數) {  
2     case 比較的值1:  
3         // 要做的事1  
4         // [可選] break;  
5     case 比較的值2:  
6         // 要做的事2  
7         // [可選] break;  
8     default: // [可選]  
9         // 預設的事  
10        // [可選] break;  
11        break;  
12 }
```



三元運算子

精簡版的if-else

在大多數情況下也不會用到，但是有些地方可以用
這個寫成精簡的寫法

```
int main() {  
    // declare  
    bool check;  
  
    // input  
    cin >> check;  
  
    // output  
    cout << (check==1 ? "Yes\n" : "No\n");  
  
    /*  
    if (check==1) {  
        cout << "Yes\n";  
    }else{  
        cout << "No\n";  
    }  
    */  
}
```

Input:

Copy

0

Expected Output:

Copy

Received Output:

Copy

No

^ Testcase 2 Failed 31ms





Input:

Copy

1

Expected Output:

Copy

Received Output:

Copy

Yes

架構解釋

條件式 ? 條件式為`true`時執行的陳述句 : 條件式為`false`時執行的陳述句

for 迴圈

架構解釋

```
for ([1] ; [2] ; [4]) {  
    [3]  
}
```

1. 初始化
2. 判斷是否繼續
3. 每次迴圈要做的事
4. 完成迴圈後要做的事

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    for (int i=1 ; i<=5 ; i++){
        cout << i << " ";
    }
    return 0;
}
```

output

1 2 3 4 5

上例題

例題1

例題2