

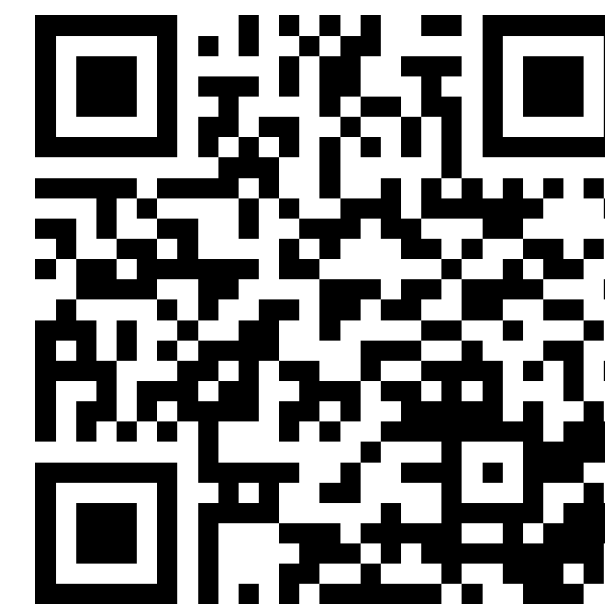
計算機程式設計

Computer Programming

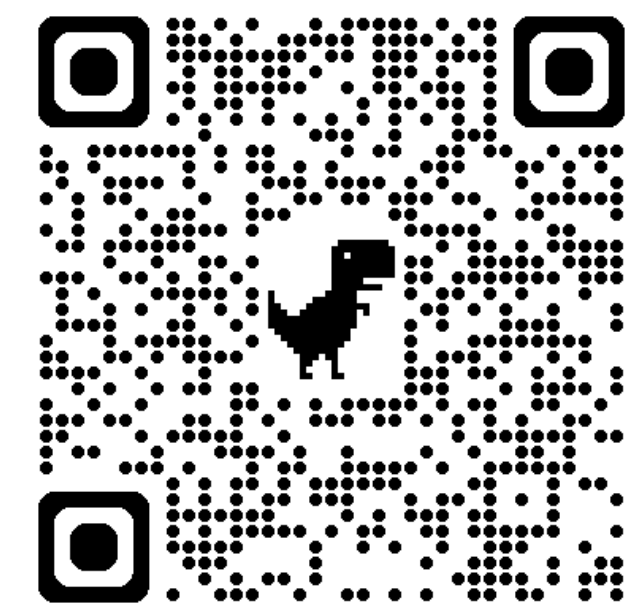
Program Control (1)

Instructor: 林英嘉

2024/09/23



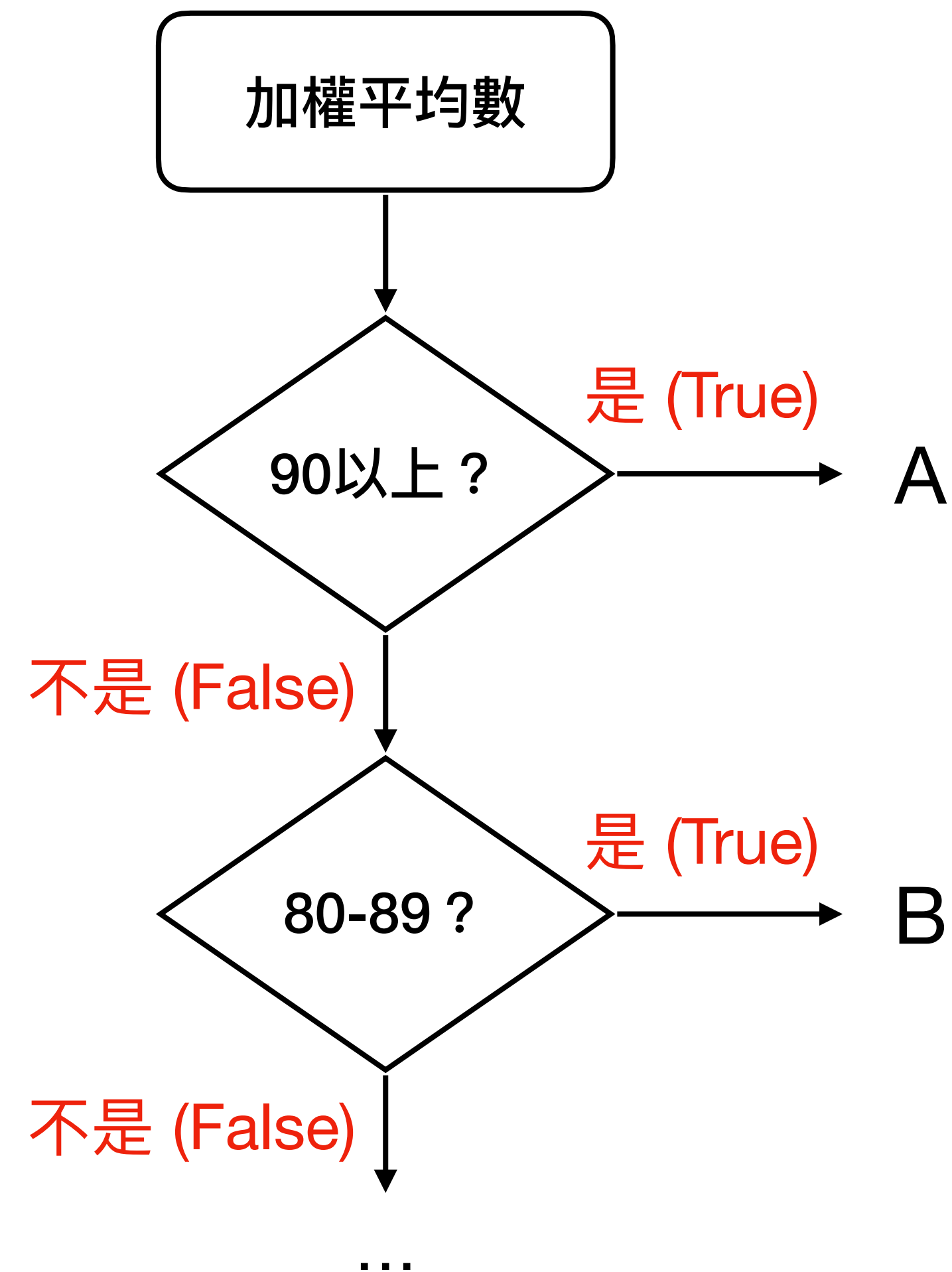
[W3 Slido: #1016922](#)



[GitHub repo](#)

流程控制 (Program Control)

流程控制指的是電腦在執行程式中敘述的「順序」



Outline: C語言程式流程控制

if-else

switch

for-loop

while-loop

迴圈

if-else 條件式

- 語意：如果 ... 就 ...

```
if (條件)  
    敘述1; // if成立後執行  
else  
    敘述2; // if不成立後執行
```

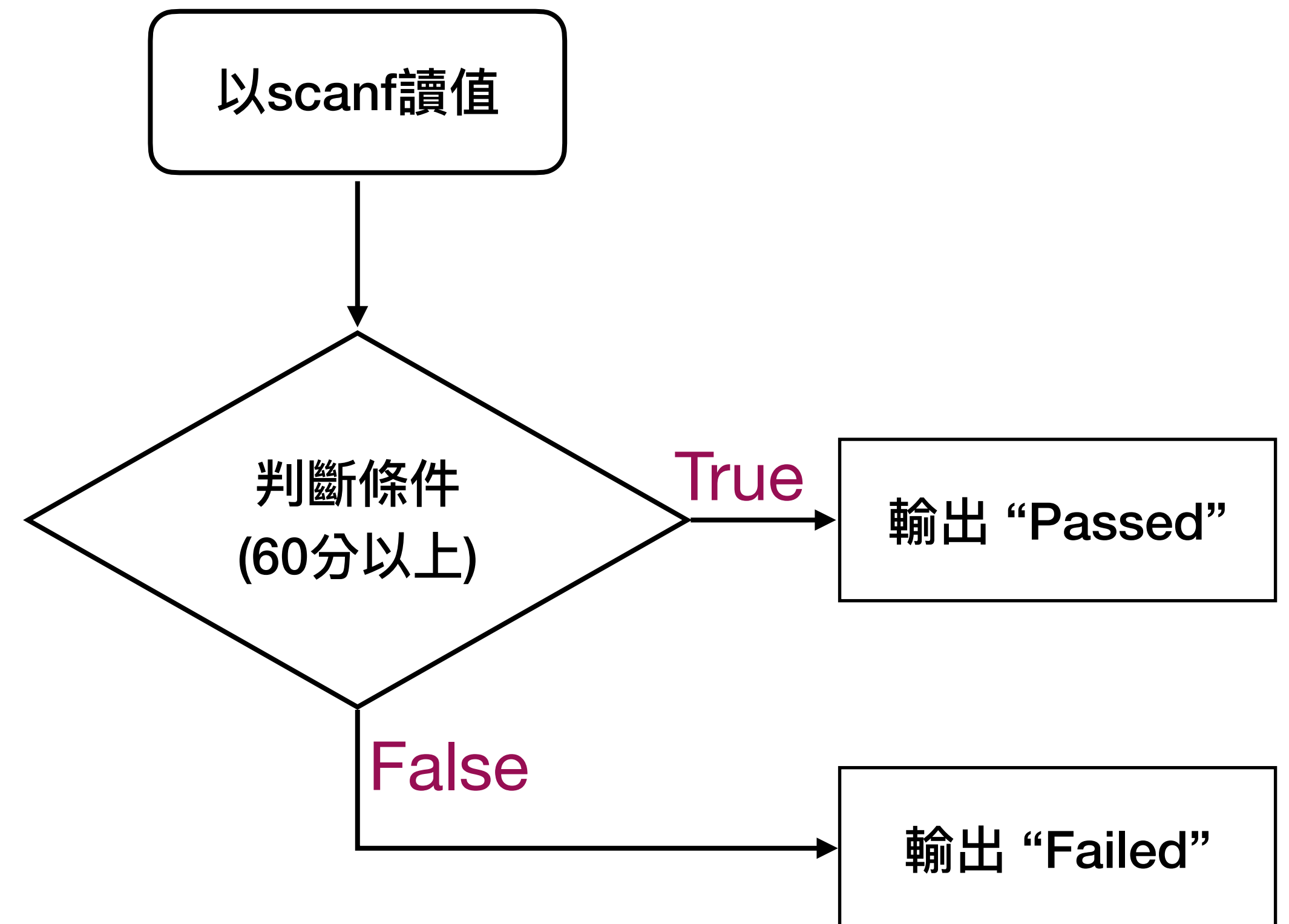
- 如果條件成立，就執行敘述1，否則，執行敘述2

if-else 敘述範例

- 範例：如何寫出一支程式，判斷輸入值60分以上為及格，59分以下為不及格？
 - 單純以簡單例子觀察if跟else的寫法

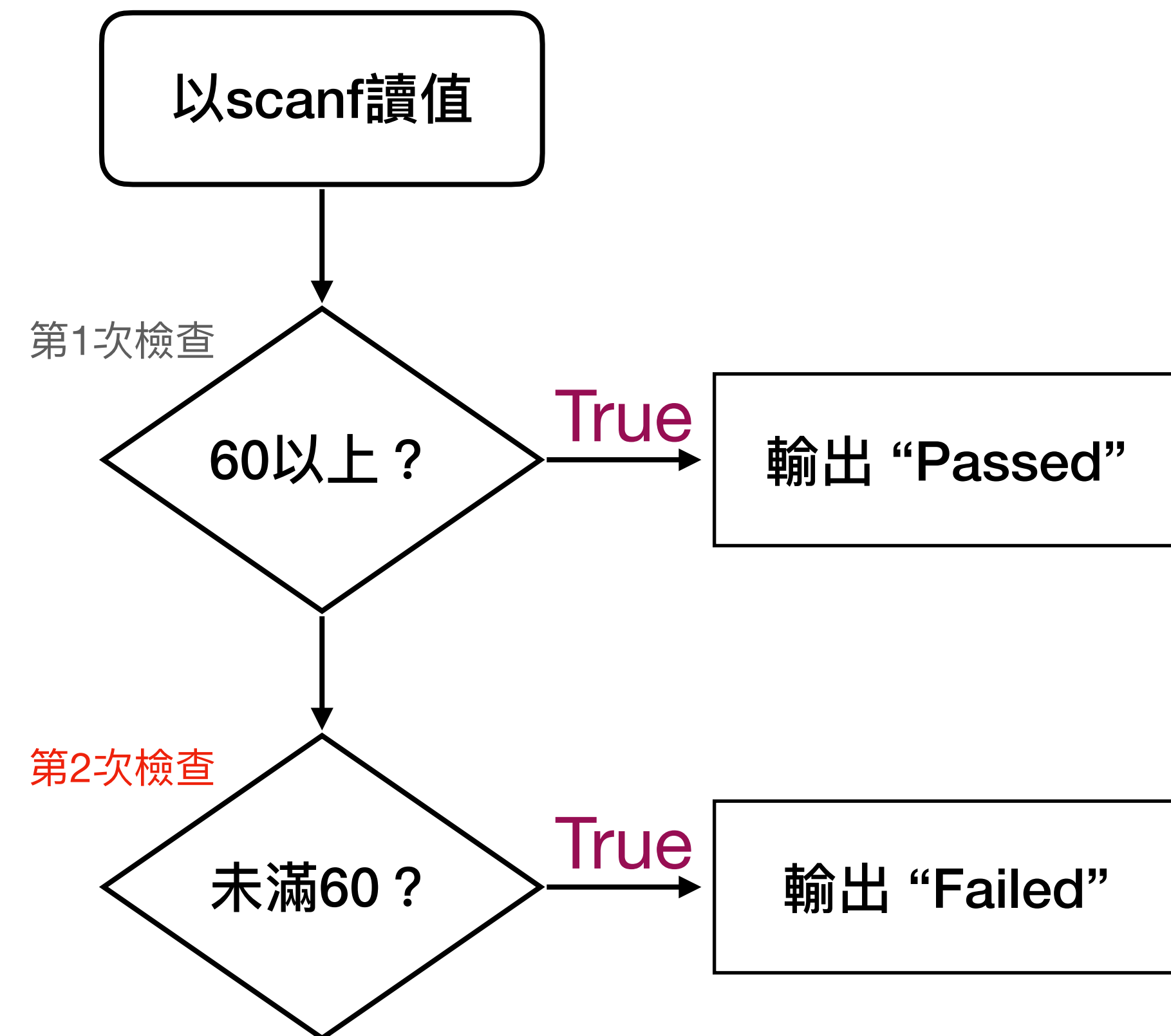
if-else 敘述 (Example 1)

```
#include <stdio.h>
int main(){
    int score;
    scanf("%d", &score);
    if (score >= 60)
        printf("Passed!");
    else
        printf("Failed!");
}
```



可以不要用 if-else 嗎？

```
#include <stdio.h>
int main(){
    int score;
    scanf("%d", &score);
    if (score >= 60)
        printf("Passed!");
    if (score < 60)
        printf("Failed!");
}
```



if 與 if-else 流程說明

- 每個 if 條件都是獨立的，這意味著你可以針對不同的條件分別進行處理，並且條件之間沒有互斥關係
- if-else可以帶來較好的效能，因為當 if 條件成立時，else 不會再檢查，這樣可以避免不必要的判斷
- 什麼時候使用「多個 if」或是 if-else? 如果條件互斥時，就使用 if-else

if-else 條件式中有多行敘述

```
if (條件){  
    敘述1_1; // if成立後執行的第1個敘述  
    敘述1_2; // if成立後執行的第2個敘述  
    ...  
}  
else{  
    敘述2_1; // if不成立後執行的第1個敘述  
    敘述2_2; // if不成立後執行的第2個敘述  
    ...  
}
```

若有兩種以上的條件式 (if 和 else)，條件式中有兩行以上的敘述，須加上大括號{ }

if-else 敘述 (Example 2): 邏輯錯誤 (else沒大括號)

```
#include <stdio.h>
int main(){
    int score;
    scanf("%d", &score);
    if (score >= 60){
        printf("分數為%d", score);
        printf("Passed!");
    }
    else
        printf("分數為%d", score);
        printf("Failed!"); // 這行一定會執行
}
```

若有兩種以上的條件式 (if 和 else)，條件式中有兩行以上的敘述，須加上大括號{ }

if-else 敘述 (Example 3): 編譯錯誤 (if沒大括號)

```
#include <stdio.h>
int main(){
    int score;
    scanf("%d", &score);
    if (score >= 60)
        printf("分數為%d", score);
        printf("Passed!");
    else
        printf("分數為%d", score);
        printf("Failed!");
}
```

若有兩種以上的條件式 (if 和 else)，且if中有兩行以上的敘述，則if須加上大括號{ }

if-else 使用說明

- 如果 if 區塊中只有一個敘述，大括號是可選的；但當有多個敘述時，大括號必須加上，否則會導致編譯錯誤。
- else 區塊一定要搭配 if 區塊才能使用
- else 區塊的大括號是可選的，但未加上大括號可能會導致邏輯錯誤 (結果錯誤)

條件運算子 (Conditional Operator)

- 條件運算子為問號搭配冒號使用 (又稱三元運算子: Ternary Operator)

(條件) ? 表達式1 : 表達式2;

條件運算子 (Conditional Operator)

- 條件運算子為問號搭配冒號使用 (又稱三元運算子: Ternary Operator)

```
(條件) ? 表達式1 : 表達式2;
```

- 等同於

```
if (條件)  
    表達式1  
else  
    表達式2;
```

條件運算子 (Conditional Operator)

- 條件運算子為問號搭配冒號使用 (又稱三元運算子: Ternary Operator)

(條件) ? 表達式1 : 表達式2;

↑ ↑
習慣上會在符號的前跟後都空一格

- 等同於

```
if (條件)  
    表達式1  
else  
    表達式2;
```

用條件運算子進行 if-else 敘述 (1)

```
#include <stdio.h>
int main(){
    int score;
    scanf("%d", &score);
    (score >= 60) ? printf("Passed!") : printf("Failed!");
}
```

條件

表達式1

表達式2

用條件運算子進行 if-else 敘述 (2)

- 也可以在表達式中進行計算 (假設我們讓不到60分的變60分，60分以上的+10分)

```
#include <stdio.h>
int main(){
    int score;
    scanf("%d", &score);
    (score >= 60) ? (score += 10) : (score = 60);
    printf("Adjusted score = %d\n", score);
}
```

用條件運算子進行 if-else 敘述 (3)

- 也可以將結果直接賦值給 `score`

```
#include <stdio.h>
int main(){
    int score;
    scanf("%d", &score);
    score = (score >= 60) ? (score + 10) : 60;
    printf("Adjusted score = %d\n", score);
}
```

用條件運算子進行 if-else 敘述 (4)

- 巢狀條件運算子 (Nested Ternary Operator)

```
#include <stdio.h>
int main(){
    int score = 49;
    scanf("%d", &score);
    score = (score >= 60) ? (score + 10) : ((score >= 50) ?
60 : (score + 10));
    printf("Adjusted score = %d\n", score);
}
```

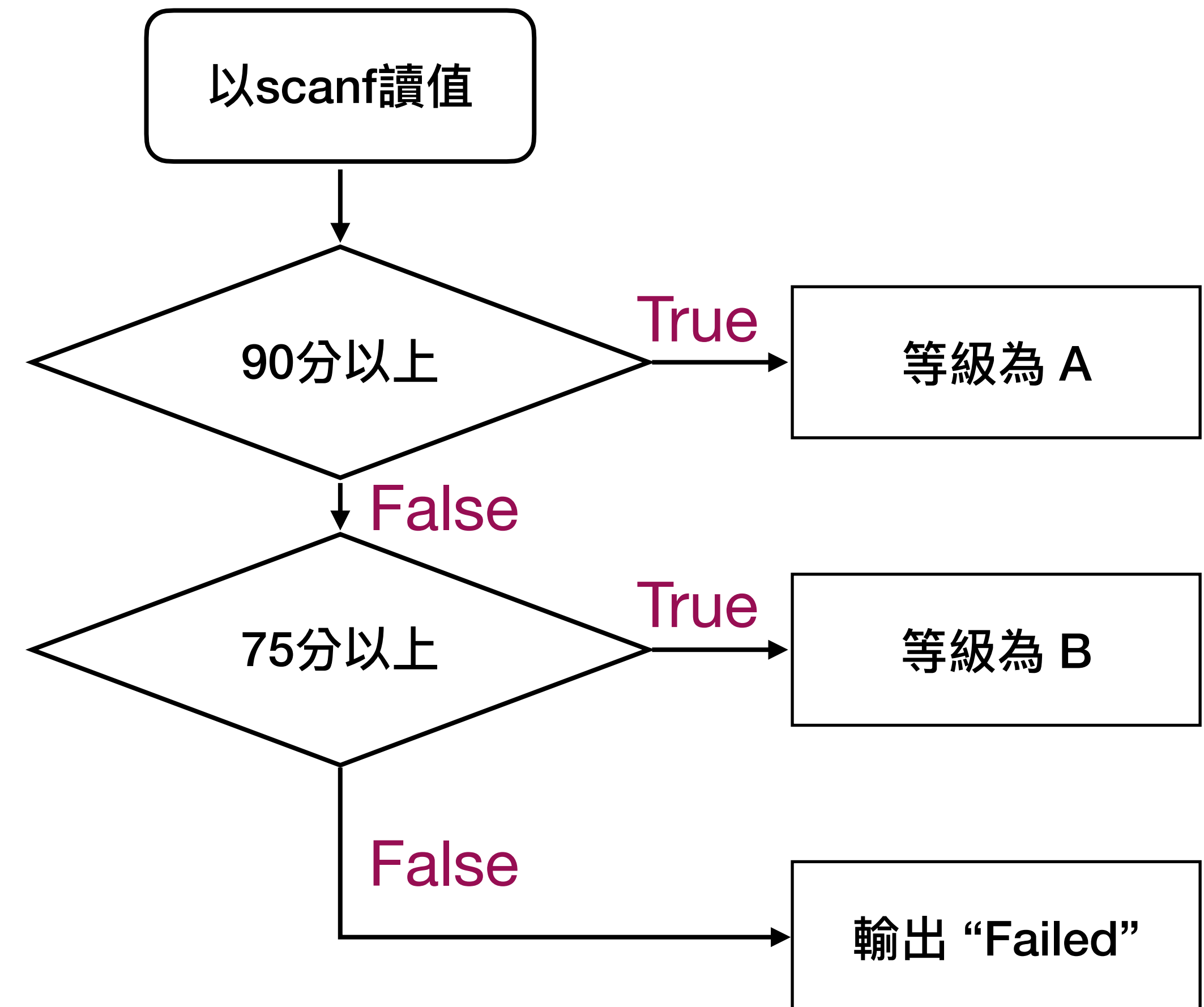
可不可以不要只有兩種結局？if-else if

以三種選項為例

```
if (條件1){  
    敘述主體1  
}  
else if (條件2){  
    敘述主體2  
}  
else{  
    敘述主體3  
}
```

使用 if-else if 結構來判斷成績等級

```
#include <stdio.h>
int main() {
    int score;
    scanf("%d", &score);
    if (score >= 90) {
        printf("等級為 A");
    } else if (score >= 75) {
        printf("等級為 B");
    } else {
        printf("Failed!");
    }
}
```



if-else if 敘述說明

- 由上往下執行，需要注意邏輯
- else if 可以有很多組，但太多 else if 會影響程式碼可讀性
- 最後面不一定要有 else

巢狀 if-else 敘述 (page1)

Code太長了分成兩頁 (1/2)

- if 巢狀結構：代表 if 條件式之中還有 if 條件式

```
#include <stdio.h>
int main(){
    int score;
    scanf("%d", &score);
    // 使用巢狀 if 結構來判斷成績等級
    if (score >= 90) {
        if (score == 100) {
            printf("等級為 A+，滿分！");
        } else {
            printf("等級為 A");
        }
    }
}
```

巢狀 if-else 敘述 (page2)

Code太長了分成兩頁 (2/2)

- 巢狀結構也可以放在 else if 之中

```
else if (score >= 70) {  
    if (score >= 85) {  
        printf("等級為 B+");  
    } else {  
        printf("等級為 B");  
    }  
}  
else  
    printf("等級為 C");  
}
```


if-else 的配對問題

- else 會與其上方最接近的 if 配對
- 使用多個 if 和 else 敘述時需要特別注意大括號的位置

if-else 的配對問題

if 代表else的配對對象

```
// 範例 1：有大括號
// 輸入 74，結果：沒有輸出
#include <stdio.h>
int main(){
    int score;
    scanf("%d", &score);
    if (score >= 60){
        if (score >= 75)
            printf("等級為 B");
    }
    else
        printf("Failed!");
}
```

```
// 範例 2：沒有大括號
// 輸入 74，結果：印出 Failed!
#include <stdio.h>
int main(){
    int score;
    scanf("%d", &score);
    if (score >= 60)
        if (score >= 75)
            printf("等級為 B");
    else
        printf("Failed!");
}
```

switch 架構概要

```
switch (flag){  
  case 選擇值1:  
    敘述主體1;  
    break;  
  case 選擇值2:  
    敘述主體2;  
    break;  
  ...  
  default:  
    敘述主體;  
}
```

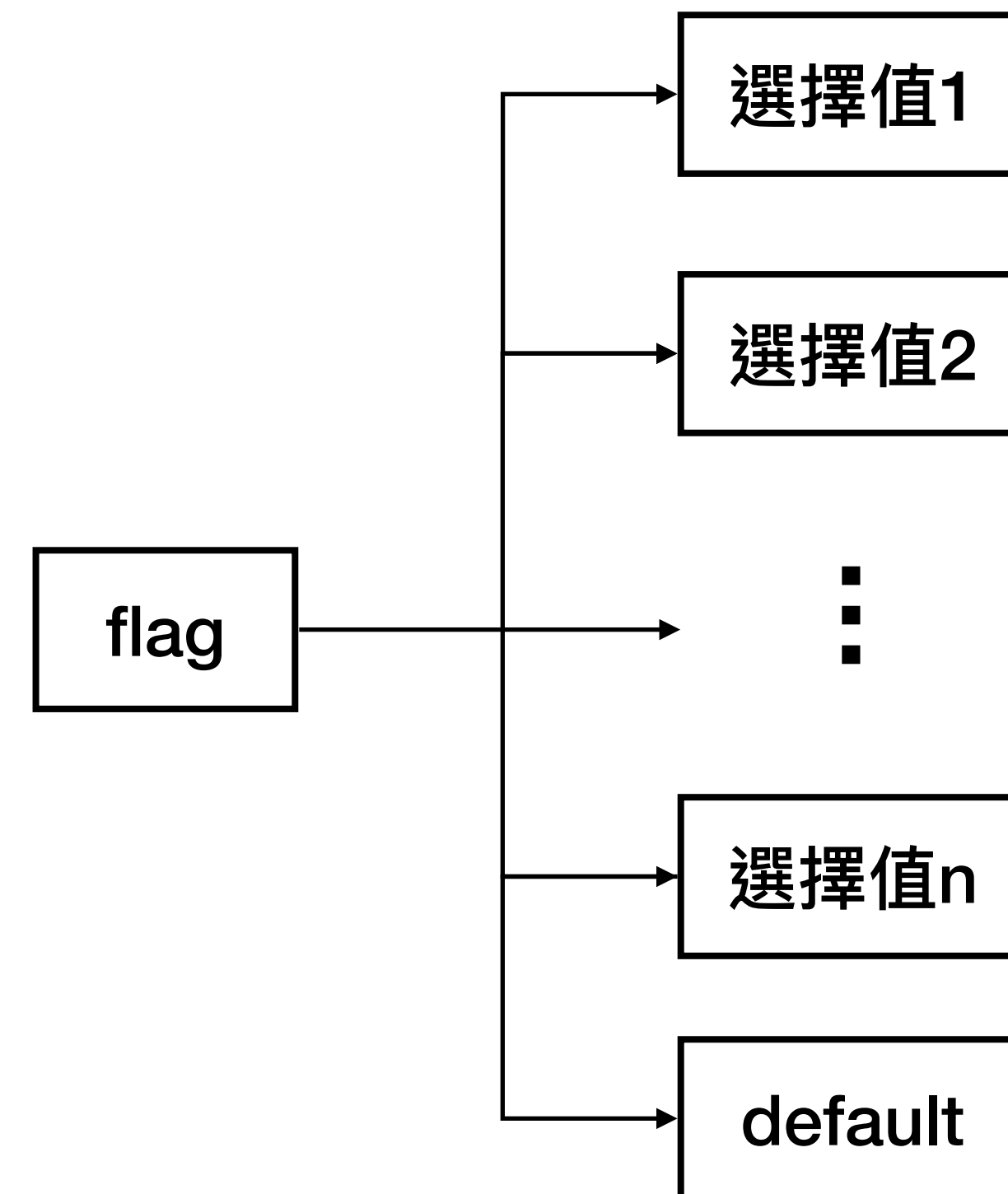
switch 說明: **flag**

```
switch (flag) {  
    case 選擇值1:  
        敘述主體1;  
        break;  
    case 選擇值2:  
        敘述主體2;  
        break;  
    ...  
    default:  
        敘述主體;  
}
```

- flag 是一個運算式或變數
- flag 的值代表要選擇哪一個 case

switch 說明: case

```
switch (flag){  
    case 選擇值1:  
        敘述主體1;  
        break;  
    case 選擇值2:  
        敘述主體2;  
        break;  
    ...  
    default:  
        敘述主體;  
}
```



根據flag的值走向其中一條case的路線，執行該case的敘述主體

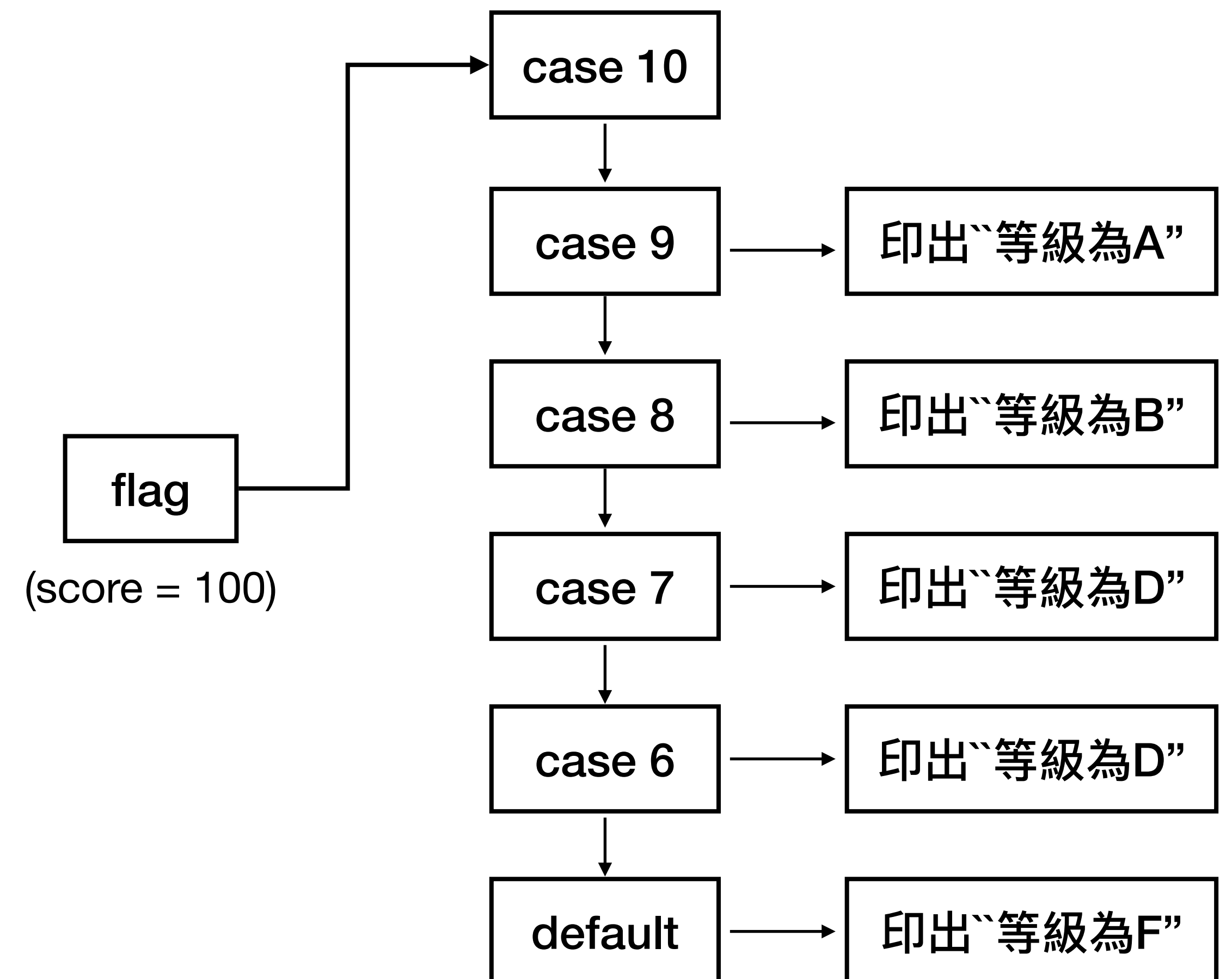
switch 範例

```
#include <stdio.h>
int main() {
    int score;
    scanf("%d", &score);
    switch (score / 10) {
        case 10: // 成績是100
        case 9:
            printf("等級為 A");
            break;
        case 8:
            printf("等級為 B");
            break;
        case 7:
            printf("等級為 C");
            break;
        case 6:
            printf("等級為 D");
            break;
        default:
            printf("等級為 F");
            break;
    }
}
```

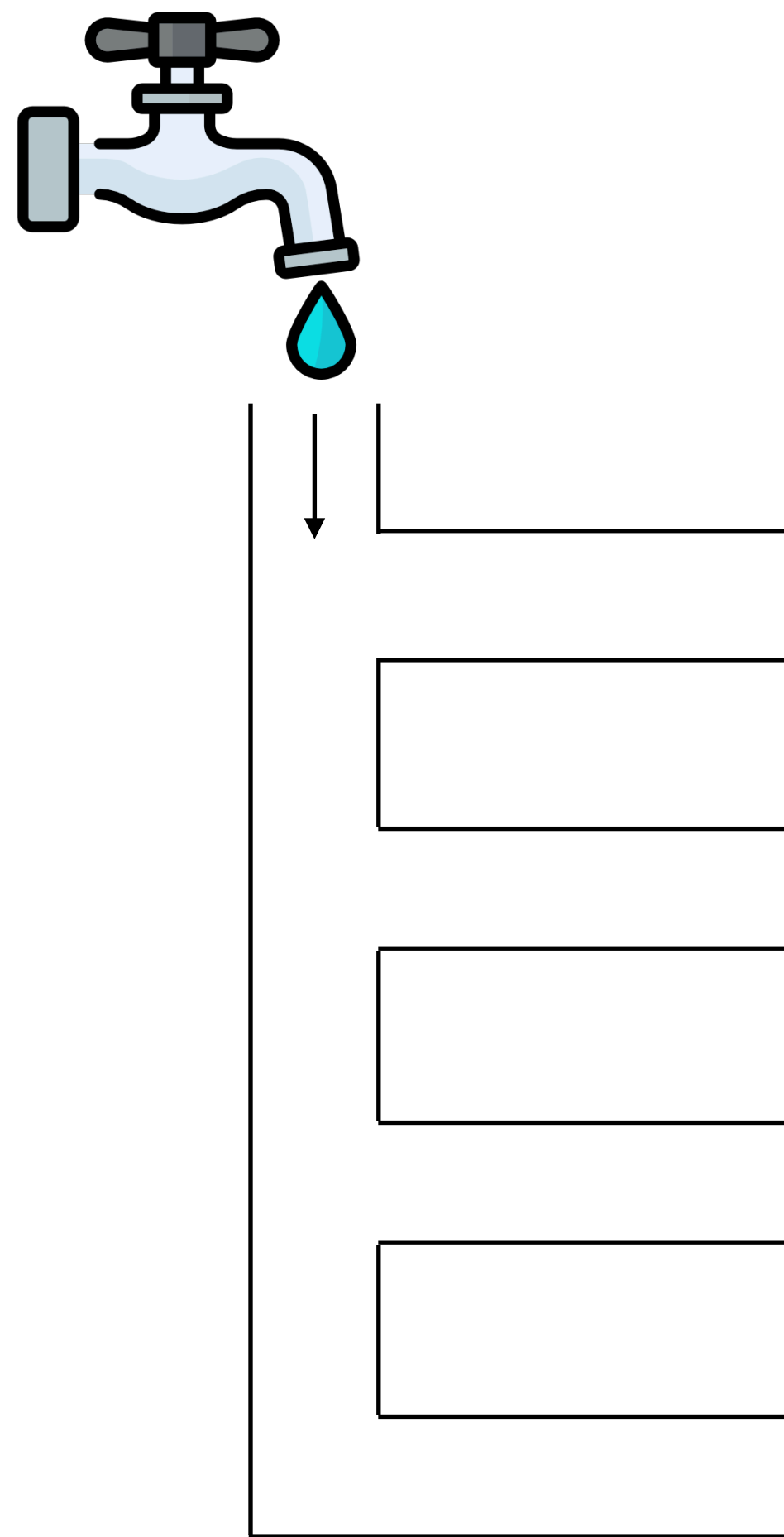
- 使用 switch 敘述來判斷 `score` 等級
- 注意！若 case 敘述中沒有break，則會往下找到最近的一個敘述
 - 輸入 100 會走到 case 9
- 代表 score / 10 的值為 10 或 9 時印出 等級為 A

switch 範例：如果都不加 break

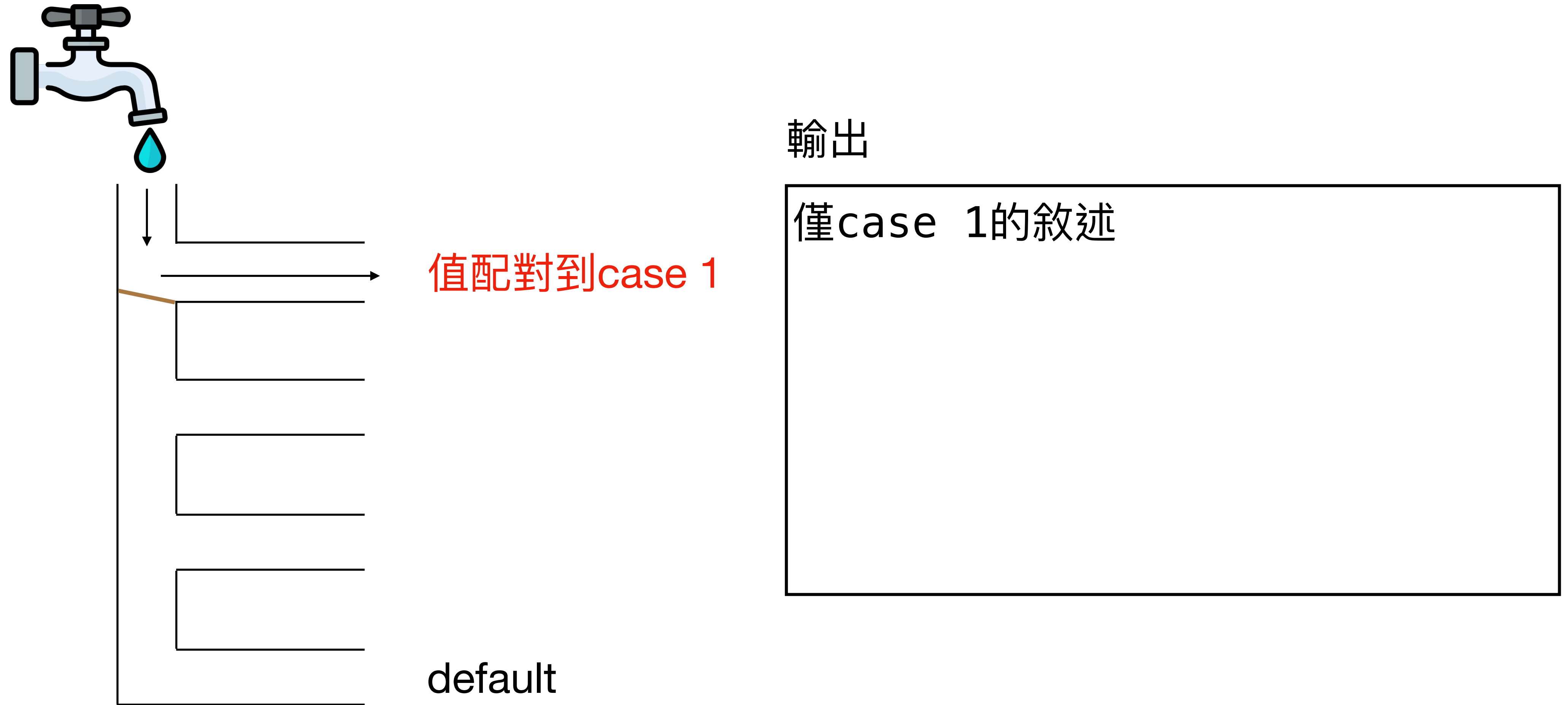
```
#include <stdio.h>
int main() {
    int score;
    scanf("%d", &score);
    switch (score / 10) {
        case 10: // 成績是100
        case 9:
            printf("等級為 A");
        case 8:
            printf("等級為 B");
        case 7:
            printf("等級為 C");
        case 6:
            printf("等級為 D");
        default:
            printf("等級為 F");
    }
}
```



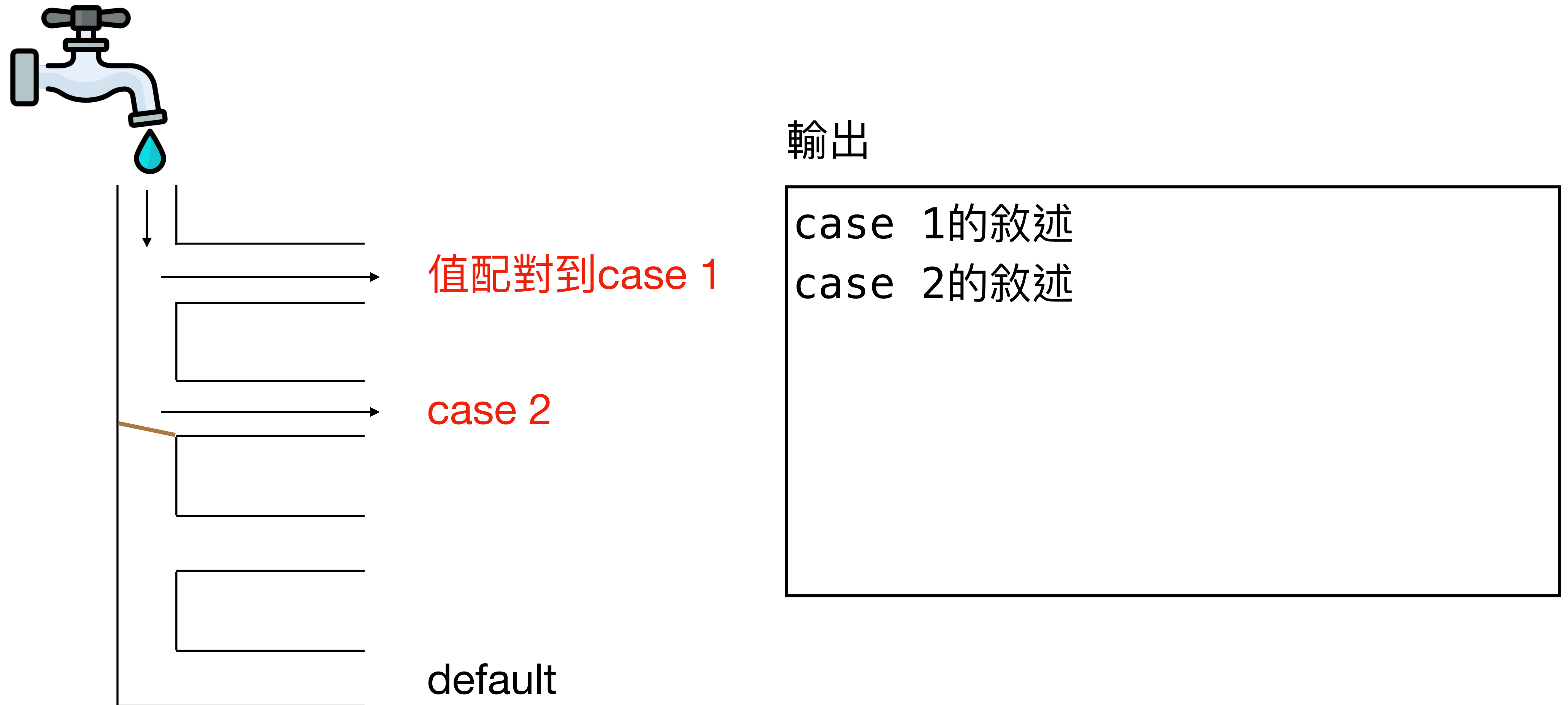
switch case 與 break 的概念



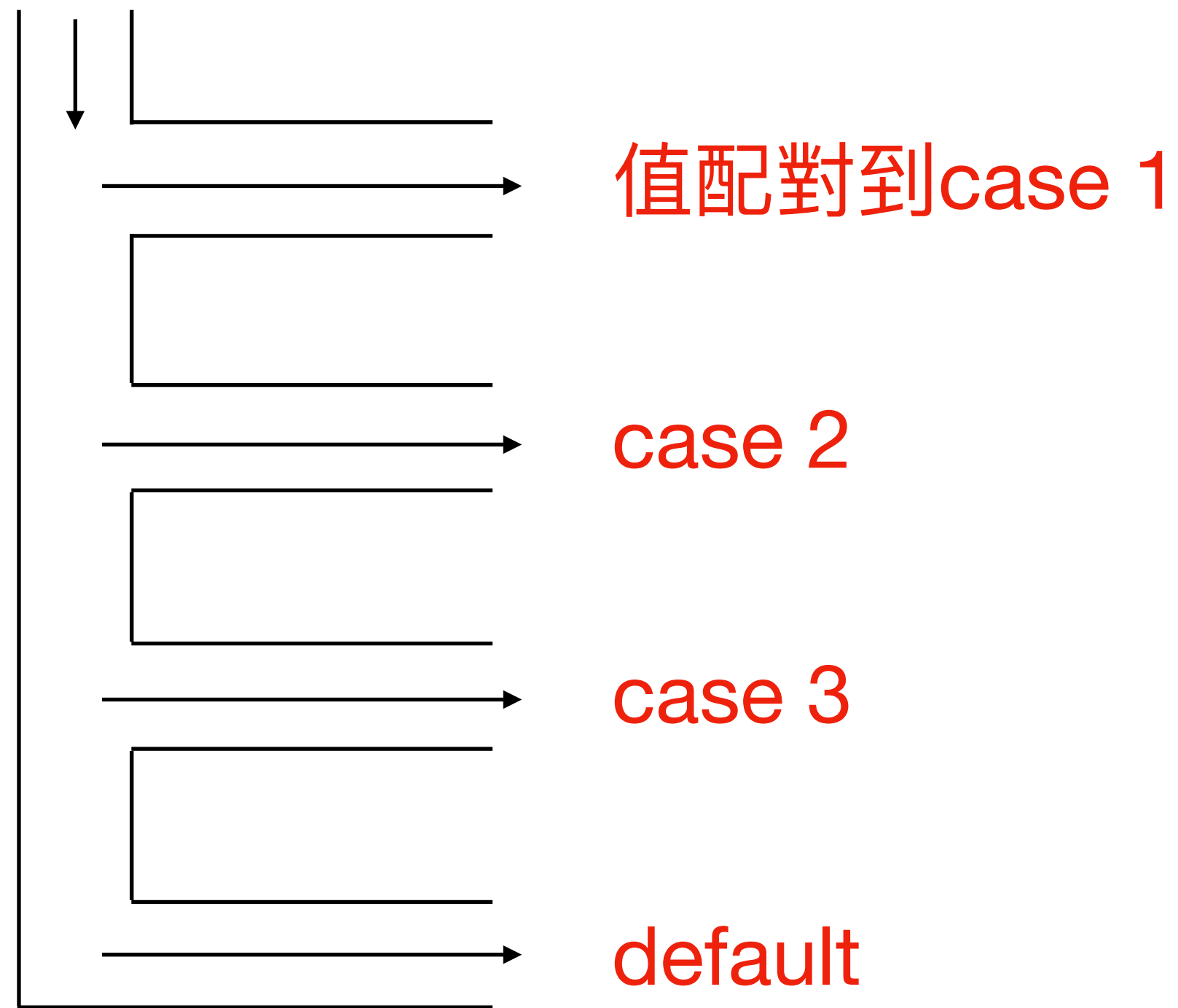
switch case 與 break 的概念



switch case 與 break 的概念



switch case 與 break 的概念



輸出

case 1的敘述
case 2的敘述
case 3的敘述
default的敘述

多個cases共用一段敘述主題

```
#include <stdio.h>
int main() {
    int score;
    scanf("%d", &score);
    switch (score / 10) {
        case 10: // 成績是100
        case 9:
            printf("等級為 A");
            break;
        case 8:
            printf("等級為 B");
        case 7:
            printf("等級為 C");
        case 6:
            printf("等級為 D");
        default:
            printf("等級為 F");
    }
}
```

```
#include <stdio.h>
int main() {
    int score;
    scanf("%d", &score);
    switch (score / 10) {
        case 10: case 9:
            printf("等級為 A");
            break;
        case 8:
            printf("等級為 B");
        case 7:
            printf("等級為 C");
        case 6:
            printf("等級為 D");
        default:
            printf("等級為 F");
    }
}
```

switch 補充說明

- default 的好處：處理**未列出**的情況，類似於 else 的概念
 - 實際上不需要 default，也能執行 switch 敘述
 - 只要你非常確定所有的可能性都有被switch列出
- break 的存在會影響到 switch 的執行邏輯，進而影響結果
 - default可以不用break;
- 注意格式 (case加冒號、敘述和break加分號)

switch的特殊問題 (少見)

- default一定要放在switch中最下面嗎？
 - Ans: 不用，但是會影響人類思考邏輯與程式碼可讀性
- default之後還可以接case的敘述嗎？
 - Ans: 可以，但非常不常見這樣的寫法 (可讀性差)
 - 如果default裡面沒有break就會影響到下一個case的結果

在if-else中可以使用break嗎？

- 答案是不行

控制結構	是否可以使用break?	用途
單獨的 if	否	break 不能在單獨的 if 語句中使用
if-else 語句	否	break 不能在 if-else 語句中使用
switch 語句	是	跳出當前 case，避免執行後續的 case

什麼時候使用switch？

- 確定值是多少 -> switch
- 知道值的範圍 -> if-else
- 可讀性 switch 比 if-else 佳
- 執行速度 switch 比 if-else 快 (通常)