

二版序

C 程式語言是一種靈活且非常專業化的程式語言，深受全世界業餘或專業的程式設計師所喜愛，從西元 1972 年發展於美國貝爾實驗室之後，直至今日，仍是許多學校在培養科學人才時，所選擇的電腦程式語言。

筆者從事資訊教育工作已經多年，在教學的過程中，發現很多學生學習程式設計時，往往都是失敗在基礎的概念沒有學好或是對於程式的運作流程無法掌握，進而發生學習狀況。往往老師在教到迴圈或陣列的時候，學生就因為練習太少而跟不上進度，嚴重者甚或放棄學習，實在十分可惜。因此本書規劃了大量的程式範例，題目超過百題，讓同學們可以不斷地實際思考與練習，以『從做中學』的方式來學習 C 程式語言。

本書一共分為 18 章，在章節安排上由淺入深，用循序漸進的方式來介紹 C 語言最核心的知識，是一本相當適合教學或自修的工具書。當各位讀者學習了 C 語言的語法與各種資料結構後，將會漸漸具備製作資訊專題的實力，本書在第 17、18 章，介紹了 2 個相當實用的資訊專題。這次第二版新增了多個程式範例，幫助讀者進一步掌握程式的開發技巧，並提升解決問題與設計專題的能力，在本書的目錄中，會在新增範例前加個★(星號)來標示。

書中的程式範例架構，大致分為『程式設計目標』、『參考程式碼』和『程式碼解說』等三個部分，讓同學們先從程式設計目標開始瞭解題目要求，自行去思考並設計解題策略，如果遇到困難，再去參考書上的程式碼，最後，可以從程式碼解說部分得到詳細的說明。

本書附錄介紹了『Online Judge』網站，也就是所謂的『線上評測系統』，其中最推薦新手入門的是『ZeroJudge』網站，因為它為中文題目且是中文操作介面，因此最好上手，而且有不少的程式設計基礎題可供入門練習之用，如果有志於程式設計競賽，可以利用線上評測系統多多練習。另外，本書也對 C++ 程式語言的語法與觀念進行介紹，並且提供了一些 C++ 程式語言的程式範例，給各位讀者參考練習。

學習程式設計需要花費大量的時間和精力，但是只要真正付出心力，其成果也將會是非常甜美的。很多的程式語言如 Java、C++、PHP...等，都是以 C 語言為根基的，所以只要把 C 語言學好，以後要再學其他程式語言，將會事半功倍，更加容易。最後，特別感謝基峰的伙伴們，奉獻無比的心力，使得本書得以更加完善 ^_^

敬祝大家

心想事成！身體健康！天天開心！

李啟龍

謹識

流程圖與選擇性敘述

- ❖ 4-1 流程圖之表示符號說明
- ❖ 4-2 判斷結構之 if 敘述
- ❖ 4-3 判斷結構之 if...else...敘述
- ❖ 4-4 判斷結構之 if...else if...else...敘述
- ❖ 4-5 判斷結構之巢狀 if
- ❖ 4-6 判斷結構之 switch 敘述
- ❖ 4-7 程式練習
- ❖ 4-8 習題

4-1 | 流程圖之表示符號說明

我們現在所使用的流程圖(Flow Chart)是在西元 1940 年由 John von Neumann 所訂定的。在進行較複雜的程式設計時，往往會藉由繪製流程圖，來輔助程式的設計。藉由流程圖的繪製，可以讓程式設計者更清楚地掌握程式的運作流程，並且對於程式的邏輯思考架構，有更深刻的瞭解。

由於電腦只能依照「程式」指示逐步完成指定的工作，因此在設計程式時必須先將問題分解成許多小步驟，然後再依一定的次序逐步執行，而這個描述問題解決程序的方法便稱做**演算法(algorithm)**。在此引用 Horowitz、Sahni 和 Dinesh 在《Fundamental of Data Structures in C++》一書對「演算法」的定義：**為解決某一問題或完成特定工作，一系列有次序且明確的指令集合**，所有演算法都會包含以下特性：

- 輸入(input)

演算法在運算前通常會有一些事先給定的輸入資料，這些資料通常是由使用者事先給予，或是在演算法的執行步驟中指定。

- 輸出(output)

演算法的目的就是產生結果，至少要有一項的輸出結果。

- 明確性(definiteness)

每個執行步驟都必須明確而清楚，不可存在模稜兩可的情況。

- 有限性(finiteness)

在任何情況下演算法一定要在有限的步驟內完成，不能無限期執行。

- 有效性(effectiveness)

演算法所描述的執行過程可以用人工的方式（例如用紙、筆），在一定時間內推算出相同的結果。

比起文字敘述，使用流程圖描述演算法有以下的好處：

- 對於複雜的演算法結構，流程圖比虛擬碼更能瞭解整個流程，方便進行除錯。

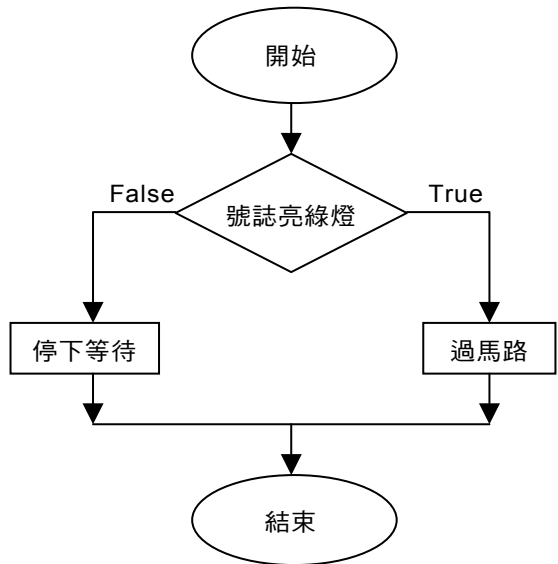
- 修改方便，可隨時加入或刪減流程圖的圖形符號。
- 可增加演算法的可讀性。

移交程式給他人進行維護，附上流程圖可幫助對方瞭解整個程式的處理流程和細部結構。

本書從第 4 章開始，就陸續有流程圖出現。這裡先針對繪製流程圖所常使用的流程圖符號，進行列表說明：

流程圖符號	名稱	說明
	程序	代表一個程式區塊，具備執行特定工作的能力。
	決策	依條件式內容來進行條件判斷，以決定程序的下一條路徑。
	流程	用來表示程式的執行方向，具有方向性。
	開始/結束	用來表示演算法的開始或是結束。

我們用下面的例子來說明如何使用流程圖來幫助我們思考與解決問題。例如當我們走到十字路口時，我們需要依據號誌的燈號，做一個判斷動作，以決定是否要過馬路或停下等待。參考的流程圖如右，假如在十字路口，遇到號誌亮綠燈才可以過馬路，否則就停下等待。當號誌亮綠燈時，條件判斷式為真（True），隨後進入過馬路動作，否則，條件判斷式為偽（False），必須進入停下等待動作。



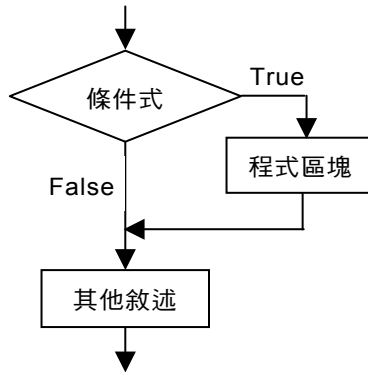
基本上流程圖所表達的是一套我們制定好的規則，並且用圖形來表達，善用它可以讓一些複雜的規則更容易傳達，也可以幫助我們去思考問題的解決方式。使用流程圖的優點在於讓人較容易了解整個作業流程，易於程式的除錯，也有助於程式的修改與維護。本書的程式流程圖為了減少頁面篇幅空間，以下的參考流程圖將會省略開始及結束等流程圖符號。

4-2 | 判斷結構之 if 敘述

if 敘述的用途是用來判斷是否進入程式區塊，如果條件式成立，則會進入程式區塊，如果條件式不成立，則會進入 if 敘述之後的其他敘述。其使用的語法如下：

```
if (條件式) {  
    程式區塊;  
}
```

使用流程圖的表示法如下：



if 敘述並不複雜，可以直接解釋為：如果條件式為真，就進入程式區塊執行敘述。參考下面的程式碼：


```
if (a%2==0)  
    a=a/2;
```

上面的兩行敘述會判斷變數 a 是否為偶數，此處是使用『取餘數』算術運算子，如果餘數等於 ($=$) 0 ，則進入 if 敘述將其值除以 2 ；也就是說如果 a 對 2 取餘數其值等於 0 ，則表示 a 是 2 的倍數，也代表 a 為偶數。

在上面這個例子中並沒有加上大括號，因為如果只有一程式敘述，大括號可以省略，如果超過一行，就必須加上大括號。再來看一個類似的例子：

```
if(a>=10 && a<=99) {
    b=a/10;
    c=a%10;
}
```

上面這四行敘述（假設變數 a, b, c 為整數型態）會判斷變數 a 是否為二位數，如果 a 為二位數（ a 介於 $10 \sim 99$ 之間），則 if 敘述會將變數 a 的十位數部分，指定給變數 b ，再將變數 a 的個位數部分，指定給變數 c 。

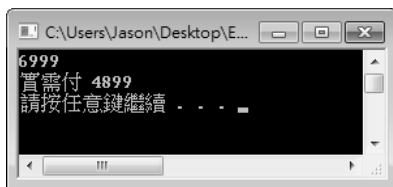


程式範例：百貨公司週年慶打折程式

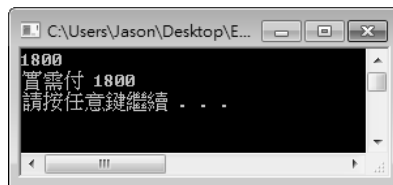
學習重點：熟悉 if 敘述的使用 / 參考檔案：4-2-1.c

📁 程式設計目標

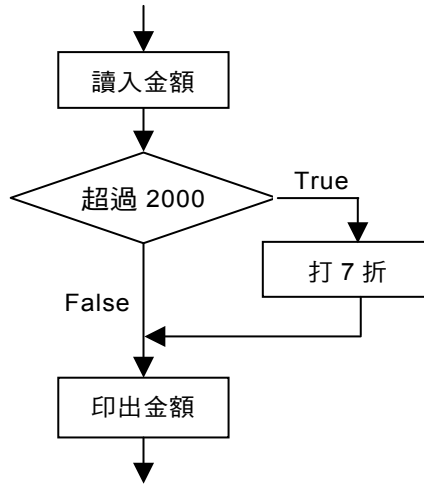
真神氣百貨公司週年慶，公司決定在週年慶期間對消費超過 2000 元的顧客打 7 折，來增加買氣，請幫該公司寫出一個收銀台程式，輸入顧客購買總金額後，計算顧客實際需付的錢。下圖為輸入『 6999 』的執行結果。



右圖為輸入『 1800 』的執行結果。



參考流程圖：



📁 參考程式碼

```
01: /*百貨公司週年慶打折程式 */
02: #include <stdio.h>
03: #include <stdlib.h>
04: int main(void)
05: {
06:     int money;
07:     scanf("%d",&money);          /* 輸入買了多少錢*/
08:     if(money > 2000)
09:         money = money*7/10;
10:     printf("實需付 %d \n",money);
11:     system("pause");           //使程式暫停在執行畫面讓我們看到結果
12:     return 0;
13: }
```

📁 程式碼解說

程式碼的第 1 行是程式的註解。

第 2~3 行：含括 (#include) 進來標準輸入與輸出標頭檔 `stdio.h` 和標準函式庫標頭檔 `stdlib.h`。

第 4~13 行：為 `main()` 函式。

第 6 行：宣告整數變數 `money`。

第 7 行：使用 `scanf()` 函式將使用者輸入的消費金額存入 `money` 整數變數。

第 8 行：用 if 敘述來判斷消費金額 money 是否超過 2000 元，如超過 2000 元，就進入第 9 行程式敘述來打七折。

第 9 行：計算打七折後的金額。

第 10 行：印出整數變數 money。

第 11 行：為使用 system() 函式，讓程式暫停在執行畫面讓我們可以看到執行結果。

第 12 行：代表 main() 函式的傳回值為整數 0。通常我們以傳回 0 代表程式正常執行完成。

4-3 | 判斷結構之 if...else...敘述

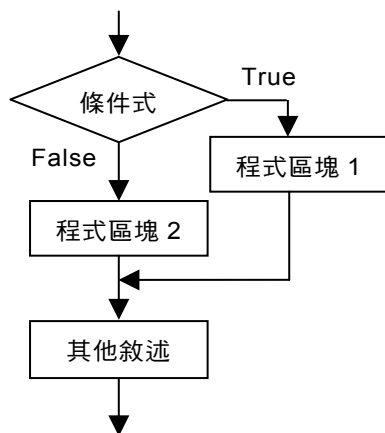
if...else...敘述會根據條件式的真偽，決定要進入的程式區塊為何。如果 if 條件式成立，則會執行 if 部分的『程式區塊 1』；如果不成立則會執行 else 部分的『程式區塊 2』，使用的語法如下：

```
if(條件式) {
    程式區塊 1;
} else {
    程式區塊 2;
}
```

使用流程圖的表示法如下：

if...else...敘述有一個特點，就是在的程式區塊和 else 之後的程式區塊，其中有一段必會被執行，如果判斷式為真，就執行 if 之後的程式區塊，如果為偽，就執行 else 之後的程式區塊，參考下面的程式碼：

```
if(a%2)
    printf("a is odd");
else
    printf("a is even");
```



上面的四行敘述會判斷 `a` 為偶數或奇數。如為奇數，對 2 取餘數會得到 1，也就是 `if` 敘述內的條件式為 `True`，因此會執行 `if` 下的『`printf("a is odd");`』敘述。如為偶數，對 2 取餘數會得到 0，也就是 `if` 敘述內的條件式為 `False`，因此會執行 `else` 下的『`printf("a is even");`』敘述。

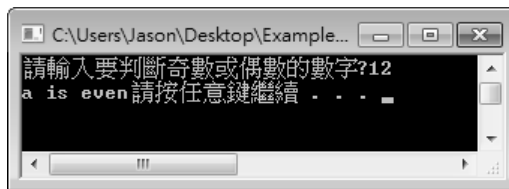
小提示 還記得 3-8 所提的三元運算子嗎？判斷印出 `a` 是奇數或偶數的 `if...else...` 程式，也可以改用三元運算子簡潔地撰寫出來，參考程式碼如下：

參考檔案：4-3-1.c

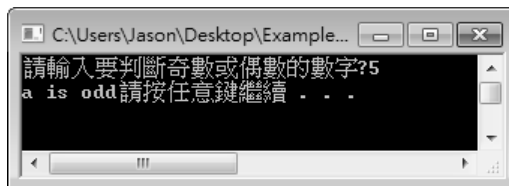
```
/* 奇數偶數判斷程式 */
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int a;
    printf("請輸入要判斷奇數或偶數的數字?");
    scanf("%d",&a);          //字元型態變數，格式符號用%c
    a%2 ? printf("a is odd") : printf("a is even") ;
    system("pause");        //使程式暫停在執行畫面讓我們看到結果
    return 0;
}
```

執行結果：

1. 輸入偶數 12。



2. 輸入奇數 5。



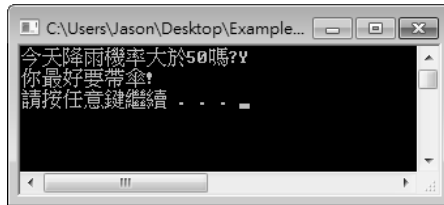


程式範例：雨天帶傘判斷程式

學習重點：熟悉 if...else...敘述的使用／參考檔案：4-3-2.c

程式設計目標

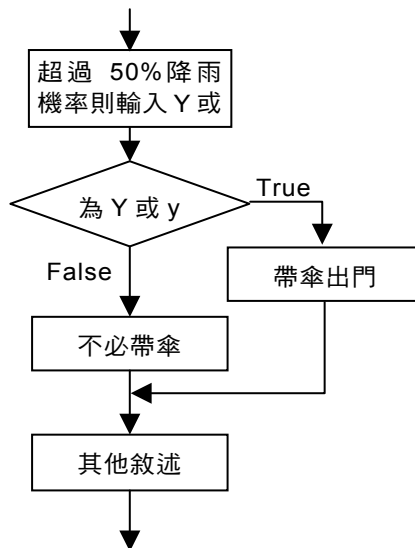
寫出一個程式，請問使用者今天是否有下雨，根據使用者的回答，輸出今天是否應該帶傘。下圖為使用者回答『Y』的執行結果。



下圖為使用者回答『N』的執行結果。



參考流程圖：



參考程式碼

```

01: /* 雨天帶傘程式 */
02: #include <stdio.h>
03: #include <stdlib.h>
04: int main(void)
05: {
06:     char rain;
07:     printf("今天降雨機率大於 50%嗎?");
08:     scanf("%c",&rain);           //字元型態變數，格式符號用%c
09:     if(rain == 'y' || rain == 'Y') //判斷輸入是否為 y 或是 Y
10:         printf("你最好要帶傘!\n");
11:     else //當使用者沒有按下 y 或 Y 時，進入 else 敘述中
12:         printf("可能不會下雨，不必帶傘了!\n");
13:     system("pause");             //使程式暫停在執行畫面讓我們看到結果
14:     return 0;
15: }

```

程式碼解說

程式碼的第 1 行是程式的註解。

第 2~3 行：含括 (#include) 進來標準輸入與輸出標頭檔 `stdio.h` 和標準函式庫標頭檔 `stdlib.h`。

第 4~15 行：為 `main()` 函式。

第 6 行：宣告字元變數 `rain`。

第 7 行：使用 `printf()` 函式印出『今天降雨機率大於 50%嗎?』。

第 8 行：使用 `scanf()` 函式讀入使用者的回答，將結果存到 `rain` 變數。因為是字元型態變數，所以格式符號用 `%c`。

第 9~12 行：用 `if` 判斷式來判斷使用者的輸入。如果是輸入『`y`』或『`Y`』，表示今天降雨機率大於 50%，程式會告訴使用者『你最好要帶傘!』；如果使用者輸入非『`y`』或『`Y`』，表示降雨機率沒有大於 50%，則會回應『可能不會下雨，不必帶傘了!』。

第 13 行：為使用 `system()` 函式，讓程式暫停在執行畫面讓我們可以看到執行結果。

第 14 行：代表 `main()` 函式的傳回值為整數 0。通常我們以傳回 0 代表程式正常執行完成。

4-4 | 判斷結構之 if...else if...else...敘述

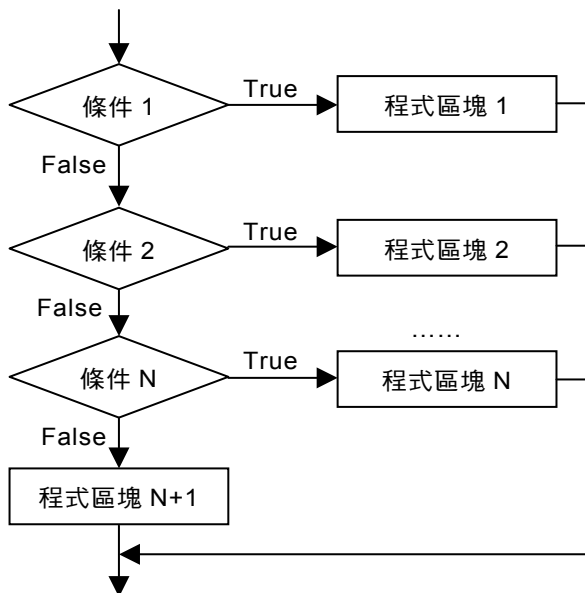
if...else if...else...敘述提供了多種狀況的選擇判斷，根據不同的條件判斷式，一一判斷要進入哪一個程式區塊。

使用的語法如下：

```

if(條件式 1) {
    程式區塊 1;
}
else if(條件式 2) {
    程式區塊 2;
}
...
else if(條件式 N) {
    程式區塊 N;
}
else {
    程式區塊 N+1;
}
    
```

使用流程圖的表示法如下：



參考下面的程式碼，此範例程式會依據條件式來判斷 a 是正數、零或是負數。當 a 值大於 0 時，會印出『a is positive』；當 a 值等於 0 時，會印出『a is zero』；當 a 值小於 0 時，則會印出『a is negative』。

```
if(a>0)
    printf("a is positive");
else if (a==0)
    printf("a is zero");
else
    printf("a is negative");
```

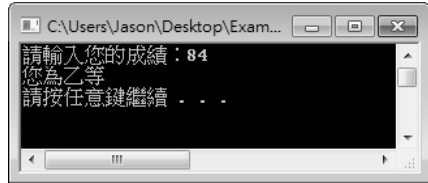


程式範例：成績區間判斷程式

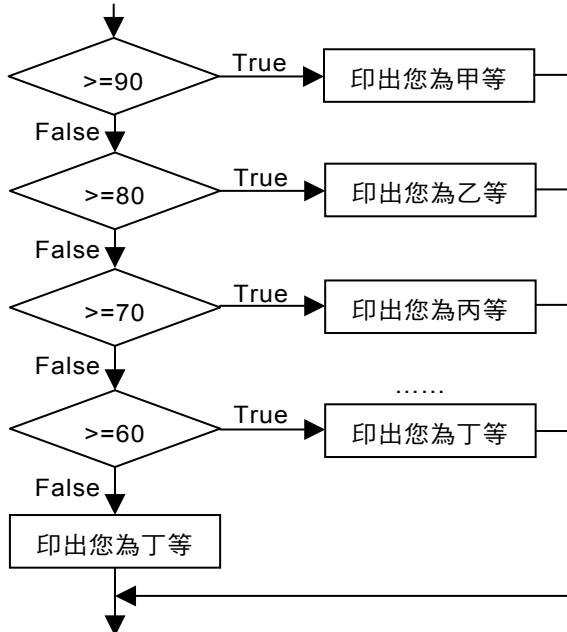
學習重點：熟悉 if...else if...else...敘述的使用／參考檔案：4-4-1.c


程式設計目標

依據使用者輸入的成績，判斷其成績等第為何？90 分以上為甲等，80 分到未滿 90 分之間為乙等，70 分到未滿 80 分之間為丙等，60 分到未滿 70 分之間為丁等，未滿 60 分為戊等。右圖為輸入『84』的執行結果，程式會告知使用者『您為乙等』。




參考流程圖：



 參考程式碼

```
01: /* 成績區間判斷程式 */
02: #include <stdio.h>
03: #include <stdlib.h>
04: int main(void)
05: {
06:     int score;
07:     printf("請輸入您的成績：");
08:     scanf("%d",&score);
09:     if(score >= 90)                //是否為 90 分以上
10:         printf("您為甲等\n");
11:     else if(score >= 80)          //小於 90 但 80 以上
12:         printf("您為乙等\n");
13:     else if(score >= 70)          //小於 80 但 70 以上
14:         printf("您為丙等\n");
15:     else if(score >= 60)          //小於 70 但 60 以上
16:         printf("您為丁等\n");
17:     else                            //小於 60 分
18:         printf("您為戊等\n");
19:     system("pause");              //使程式暫停在執行畫面讓我們看到結果
20:     return 0;
21: }
```

 程式碼解說

第 8 行：使用 `scanf()` 函式讀入使用者的分數，將結果存到 `score` 變數。

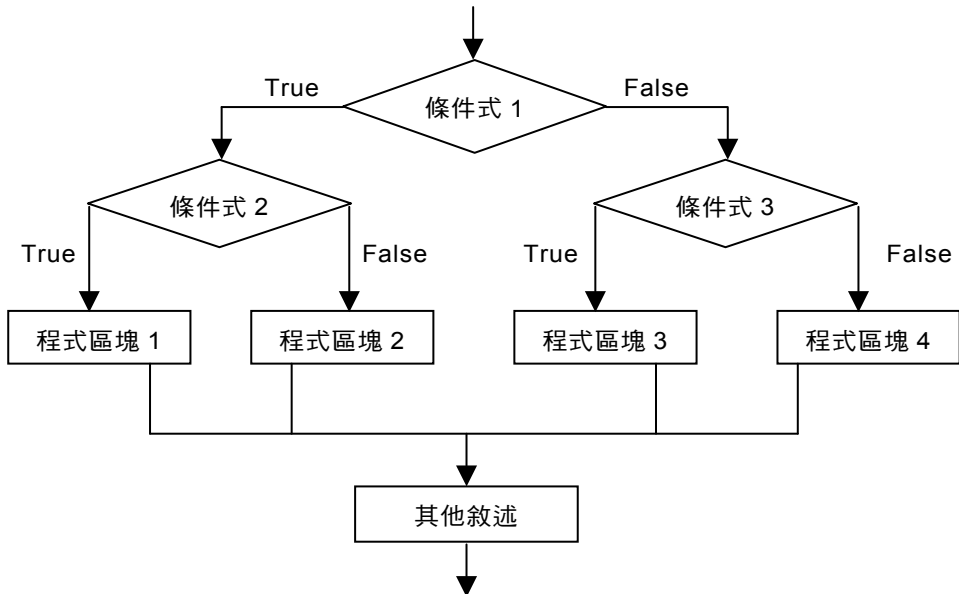
第 9~18 行：為 `if...else if...else...` 敘述，如果成績高於等於 90 分，會執行第 10 行，印出『您為甲等』。如果成績介於 80 分~89 分，會執行第 12 行，印出『您為乙等』。如果成績介於 70 分~79 分，會執行第 14 行，印出『您為丙等』。如果成績介於 60 分~69 分，會執行第 16 行，印出『您為丁等』。其他的成績則會印出『您為戊等』。

4-5 | 判斷結構之巢狀 if

當 if 敘述的程式區塊中包含了其他的 if 敘述時，我們稱此種 if 敘述為巢狀 if 敘述，我們舉一個巢狀 if 的語法架構如下：

```
if(條件式 1) {  
    if(條件式 2) {  
        程式區塊 1;  
    }  
    else {  
        程式區塊 2;  
    }  
else {  
    if(條件式 3) {  
        程式區塊 3;  
    }  
    else {  
        程式區塊 4;  
    }  
}
```

使用流程圖的表示法如下：



當我們在撰寫巢狀 if 的時候，盡量將相對應的 if...else...敘述對齊，以利未來程式之維護或修改。



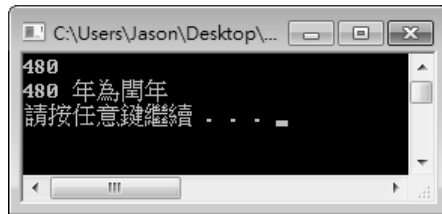
程式範例：閏年判斷程式

學習重點：熟悉巢狀 if 敘述的使用／參考檔案：4-5-1.c

📁 程式設計目標

請寫出一程式，輸入西元年份，判斷該年是否為閏年（閏年判斷方式為：四年一閏，逢百年不閏，四百年又閏）。如果輸入『2100』，程式會輸出『2100 年不為閏年』。如果輸入『2000』，程式會輸出『2000 年為閏年』。

如圖為輸入『480』的執行結果。



📁 參考程式碼

```
01: #include <stdio.h>
02: #include <stdlib.h>
03: int main(void)
04: {
05:     int year;
06:     scanf("%d",&year);
07:     if(!(year%4)) {                               /*可被 4 整除*/
08:         if(!(year%100) && year%400)             /*被 4 和 100 都整除，但不被 400 整除*/
09:             printf("%d 年不為閏年\n",year);
10:         else
11:             printf("%d 年為閏年\n",year); /*被 4 和 400 都整除，但不被 100 整除*/
12:     }
13:     else
14:         printf("%d 年不為閏年\n",year); /*無法被 4 整除*/
15:     system("pause"); //使程式暫停在執行畫面讓我們看到結果
16:     return 0;
17: }
```

📁 程式碼解說

第 6 行：使用 scanf() 函式讀入使用者輸入的西元年，將結果存到 year 變數。

第 7~12 行：此段 if 敘述處理 year 是 4 的倍數的情況，用的運算是取餘數運算，如果餘數為 0，表示該數為倍數。第 8、9 行處理被 100 整除，但不被 400 整除的狀況，印出『不為閏年』。第 10、11 行處理被 4 和 400 都整除，但不被 100 整除的狀況，印出『為閏年』。

第 13、14 行：此段 else 敘述處理 year 不是 4 的倍數的情況，只要不是 4 的倍數，就一定也不可能是 100 的倍數或是 400 的倍數，在這種情況下印出『不為閏年』。



程式範例：計程車資計費程式

學習重點：熟悉巢狀 if 敘述的使用／參考檔案：4-5-2.c

程式設計目標

請寫出一程式，輸入搭乘的里程（公尺），然後計算出車費。假設搭乘里程在 1500 公尺以下，則價格皆為 70 元；爾後每超過 500 公尺加收 5 元，不足 500 公尺部分以 500 公尺計算。如圖為輸入『1300』的執行結果，未超過 1500 公尺，所以收基本車資 70 元。

```

D:\Book3_第一次學C就上手... - □ ×
請輸入里程<單位為公尺>:1300
總共車資為: 70 元
請按任意鍵繼續 . . .


```

如圖為輸入『2800』的執行結果，超過 1500 公尺的部分為 1300 公尺，每 500 公尺加收 5 元，總共加收 15 元，總計車資為 85 元。


```

D:\Book3_第一次學C就上手... - □ ×
請輸入里程<單位為公尺>:2800
總共車資為: 85 元
請按任意鍵繼續 . . .

```

 參考程式碼

```
01: #include <stdio.h>
02: #include <stdlib.h>
03: int main(void)
04: {
05:     int i,diff,cost;
06:
07:     printf("請輸入里程(單位為公尺):");
08:     scanf("%d",&i);
09:     if (i<=1500) /*里程數小於等於 1500 公尺*/
10:         cost=70;
11:     else /*計算里程數超過 1500 公尺的里程費率*/
12:     {
13:         diff=i-1500;
14:         if (diff%500==0)
15:             cost=70+(diff/500)*5;
16:         else
17:             cost=70+((diff/500)+1)*5;
18:     }
19:     printf("總共車資為: %d 元\n",cost);
20:     system("pause");
21:     return 0;
22: }
```

 程式碼解說

第 6 行：使用 `scanf()` 函式讀入使用者輸入的搭乘的距離，將結果存到 `i` 變數。

第 9 行：此行 `if` 敘述處理里程數小於等於 1500 公尺的情況，搭乘的車資固定為 70 元，並將此值指定給 `cost` 變數。

第 11~18 行：此段 `else` 敘述處理里程數大於 1500 公尺的情況，此為巢狀 `if` 敘述的情況。

第 13 行：將超過 1500 公尺的距離算出，並且指定給 `diff` 變數。

第 14、15 行：『%』為取餘數運算，如 `diff % 500` 運算等於 0 成立，代表 `diff` 距離為 500 的倍數。所以，直接將 70 元加上 $(diff/500)*5$ 之值為車資。

第 16、17 行：如 `diff % 500` 運算等於 0 不成立，代表 `diff` 距離不為 500 的倍數，其不足 500 公尺的距離需要加上 1 個 5 元。例如：1300 公尺為 2 個 500 公尺再加上 300 公尺，也就是 10 元再加上 1 個 5

元（300 公尺）。因此超過基本費的車資需要將(diff/500)加 1 後再乘以 5。

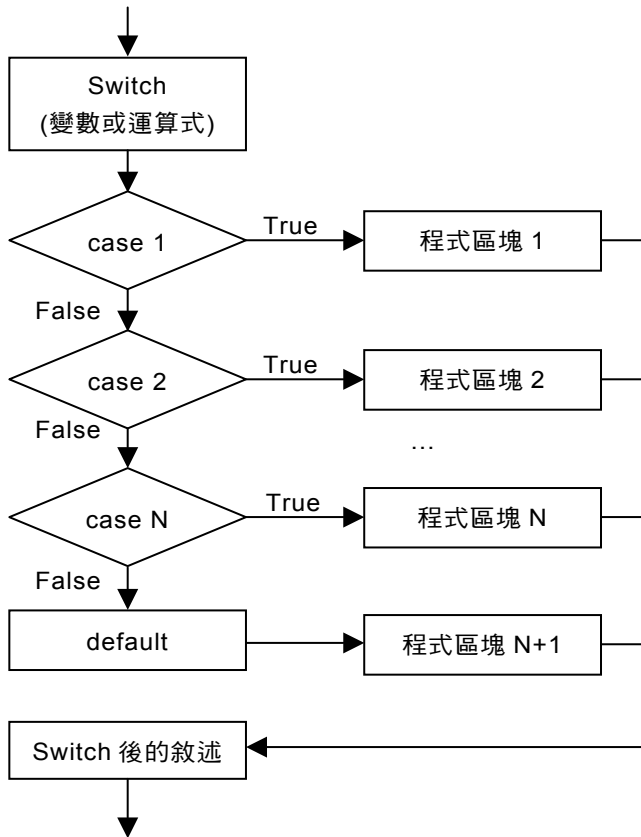
第 19 行：印出 cost 變數值為車資。

4-6 | 判斷結構之 switch 敘述

在我們想要提供兩個選擇時，我們可以用 if 來做判斷，選擇要執行的程式區塊。當有多重選擇要進行判斷，要多選一時，我們可以用 if...else if...else...敘述來做判斷。另外，C 語言為了將多選一的情況簡化，還提供了 switch 敘述，switch 會根據某一個字元或是整數變數，來判斷要進入哪一個程式區塊，適用的語法如下：

```
switch(變數或運算式) {  
    case 值 1:  
        程式區塊 1;  
        break;  
    case 值 2:  
        程式區塊 2;  
        break;  
    ...  
    case 值 N:  
        程式區塊 N;  
        break;  
    default:  
        程式區塊 N+1;  
}
```

也可以用下方的流程圖表示：



switch 敘述會根據放在括號內的變數或是運算式的結果，比對下方 case 內的資料，如果有 case 指定的值符合，就執行該程式區段，直到遇到 break 敘述，再跳出 switch 的程式區塊；如果沒有值符合，會進入 default 之後的程式區塊。有時也可以不寫 default 區塊，直接跳出 switch 區塊。

我們先來看一個模擬計算機程式，我們先讓使用者輸入想要執行的運算（以一個字元表示），如果輸入『+』符號，程式會計算將變數 a 的值加上變數 b 的值，並將結果印出；如果輸入『-』符號，程式會計算將變數 a 減去 b 的值，並將結果印出；如果輸入的不是『+』也不是『-』符號，程式印出『I don't know how to do that.』字串，參考程式碼如下。

```

char ch;
scanf("%c",&ch);
switch(ch) {
    case '+':
        printf("a + b = %d",a+b);
        break;
    case '-':
        printf("a - b = %d",a-b);
        break;
    default:
        printf("I don't know how to do that.");
}

```

小提示 使用 switch 敘述，最常犯的錯誤是在一個 case 結束時，忘了加上 break 敘述，這時候程式會繼續往下一個 case 執行，直到遇見 break 敘述為止。造成我們只想執行某一個 case 內的程式區塊，卻不小心多執行了其他區塊。初學 switch 敘述時，最好是養成一個 case 對應一個 break 的好習慣。

延伸上一個程式，我們設計一個程式，此程式當使用者輸入『p』或是『+』，都可以將變數 a 和變數 b 相加。當 ch 為 + 時，會進入 case '+' 區塊之中，因為該區塊沒有 break 敘述，就繼續往下執行，接著會進入 case 'p' 區塊之中，印出 a+b 的結果之後，才遇到 break 敘述來跳出 switch 程式區塊，結束 switch 敘述的執行。

```

switch(ch) {
    case '+':
    case 'p': //plus
        printf("a + b = %d",a+b);
        break;
    case '-':
    case 'm': //minus
        printf("a - b = %d",a-b);
        break;
}

```



程式範例：運用 switch 敘述設計四則計算機程式

學習重點：熟悉 switch 敘述的使用／參考檔案：4-6-1.c

程式設計目標

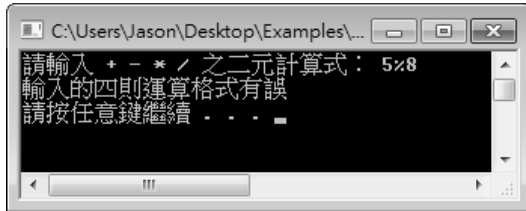
用 Switch 敘述撰寫一個做四則計算的程式，可以處理兩個運算元和一個加、減、乘、除運算子的計算。例如：輸入『5+8』，程式會回應『13』，執行結果如下圖所示。



輸入『5/8』，程式會回應『0.625000』，執行結果如下圖所示。



輸入『5%8』，程式會回應『輸入的四則運算格式有誤』，執行結果如下圖所示。



參考程式碼

```
01: /* 用 Switch 敘述做四則計算程式 */
02: #include <stdio.h>
03: #include <stdlib.h>
04: int main(void)
05: {
06:     int a,b;
07:     char oper;
08:     printf("請輸入 + - * / 之二元計算式: ");
```

```
09:     scanf("%d %c %d",&a ,&oper, &b);
10:     switch(oper)
11:     {
12:         case '+':
13:             printf("計算結果為%d\n", a+b);
14:             break;
15:         case '-':
16:             printf("計算結果為%d\n", a-b);
17:             break;
18:         case '*':
19:             printf("計算結果為%d\n", a*b);
20:             break;
21:         case '/':
22:             printf("計算結果為%f\n", (float)a/b);
23:             break;
24:         default:
25:             printf("輸入的四則運算格式有誤\n");
26:     }
27:     system("pause");
28:     return 0;
29: }
```

程式碼解說

第 7 行：宣告 `oper` 為字元變數，為運算子 `operator` 的縮寫，用來儲存使用者輸入的運算子，也就是『+』、『-』、『*』、『/』四個運算子。

第 9 行：以『`a+b`』、『`a-b`』、『`a*b`』或『`a/b`』的格式讀入運算式。

第 10~26 行：使用 `switch` 敘述來做運算，依據字元變數『`oper`』的值，來決定要執行哪一個對應的 `case`。

第 12~14 行：處理『+』的運算。

第 15~17 行：處理『-』的運算。

第 18~20 行：處理『*』的運算。

第 21~23 行：處理『/』的運算，此處還有加上強制型態轉換指令（`float`）。

第 24、25 行：處理其他的運算子情況，會告訴使用者『輸入的四則運算格式有誤』。

4-7 | 程式練習



程式範例 1：單位轉換程式

學習重點：熟悉 if...else if...else...敘述的使用／參考檔案：4-7-1.c

📁 程式設計目標

1 公尺 = 3.28 英呎，1 公斤 = 2.2 英鎊，請寫出一個可讓使用者自由選擇要轉換『公尺->英呎』或『公斤->英鎊』的程式。下圖為輸入『1』後，選擇公尺轉英呎運算，再輸入『50』的執行結果。



📁 參考程式碼

```

01: /*單位轉換程式 */
02: #include <stdio.h>
03: #include <stdlib.h>
04: int main(void)
05: {
06:     char ch;
07:     double input;
08:     printf("您要轉換 1)公尺 -> 英呎 2)公斤 -> 英鎊");
09:     scanf("%c",&ch);          /* 輸入選擇 */
10:     scanf("%lf",&input);      /* 輸入欲轉換的數 */
11:     if(ch == '1')
12:         printf("%lf 公尺 = %lf 英呎\n",input,input*3.28);
13:     else if(ch == '2')
14:         printf("%lf 公斤 = %lf 英鎊\n",input,input*2.2);
15:     else
16:         printf("沒有這個選項\n");
17:     system("pause");          //使程式暫停在執行畫面讓我們看到結果
18:     return 0;
19: }
    
```

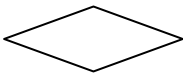
4-8 | 習題

選擇題：

() 1. 使用 switch 多重選擇結構，我們通常會在一個 case 結束時，加上何種指令？

- (a) more
- (b) break
- (c) then
- (d) continue

() 2. 請問在流程圖中，「決策」所使用的符號為下列何者？

- (a) 
- (b) 
- (c) 
- (d) 

() 3. 請問下列哪一種並非 C 的判斷流程語法？

- (a) switch...case
- (b) select...case
- (c) if...else
- (d) if...else if...else

() 4. 如果 a 的值為 2，則執行下列程式後會輸出何值？

```
if(a==3)
a=3;
a=4;
```

- (a) 2
- (b) 3
- (c) 4
- (d) 5

() 5. 如果 a 的值為 2，則執行下列程式後 a 為多少？

```
if(a==3)
a=3;
else
a=4;
```

- (a) 2
- (b) 3
- (c) 4
- (d) 5

() 6. 下列何者不是流程圖的優點？

- (a) 易於程式的除錯
- (b) 讓人較容易了解整個作業流程
- (c) 有助於程式的修改與維護
- (d) 流程圖符號沒有標準較彈性

- () 7. 下列何者適合用來描述演算法的執行過程？
- (a) 折線圖 (b) 圓餅圖
(c) 魚骨圖 (d) 流程圖
- () 8. 使用電腦解題，用來描述問題的解決方法稱為？
- (a) 敘述法 (b) 圖解法
(c) 演算法 (d) 模擬法
- () 9. 下列何者「不是」演算法的特性？
- (a) 演算法描述的執行過程，可以用人工的方式推算出同樣的結果
(b) 執行過程需要在有限步驟結束
(c) 不需要產生輸出結果
(d) 執行步驟一定要明確清楚
- () 10. 下列何者不是變數的資料型態？
- (a) 網址 (b) 整數
(c) 浮點數 (d) 字串