

選擇結構

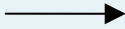

3




程式的三個主要結構為**循序結構**、**選擇結構**與**重複結構**。**循序結構**為程式有從開始逐行執行的特性，第一行執行完畢後執行第二行，第二行執行完畢後執行第三行，直到程式執行結束；**選擇結構**為若條件測試的結果為真，則做條件測試為真的動作，否則執行條件測試為假的動作，例如：若成績大於等於 60 分，則輸出及格，否則輸出不及格。**重複結構**讓電腦重複執行某個區塊的程式多次，電腦適合做重複的工作，例如：求 $1+2+3+\dots+1000$ ，使用重複結構可在很短時間內重複執行加總程式，直到求出結果，善用這三種結構可以寫出解決複雜問題的程式。

日常生活中也有許多選擇結構的對話，「若明天天氣很好的話，我們就去動物園，否則就待在家裡」，程式語言提供選擇結構的程式結構，讓使用者可以於程式中使用，邏輯上的語意為「若測試條件成立，則執行條件成立的動作，否則執行條件不成立的動作」，許多問題的解決過程，都會遇到選擇結構，如登入系統時需要驗證帳號和密碼，正確則可登入系統，否則跳到登入畫面，重新輸入帳號與密碼。選擇結構分成單向選擇結構、雙向選擇結構、多向選擇結構與巢狀選擇結構，以下分別說明敘述。

3-1 ▶ 流程圖簡介

流程圖常用於幫助初學程式設計者寫出問題的解題步驟，若能將解題流程以流程圖表示，就可以轉換成程式語言，所以流程圖也需提供程式語言的三個主要結構**循序結構**、**選擇結構**與**重複結構**，在介紹條件判斷前，我們要先瞭解流程圖的圖示，如下表。

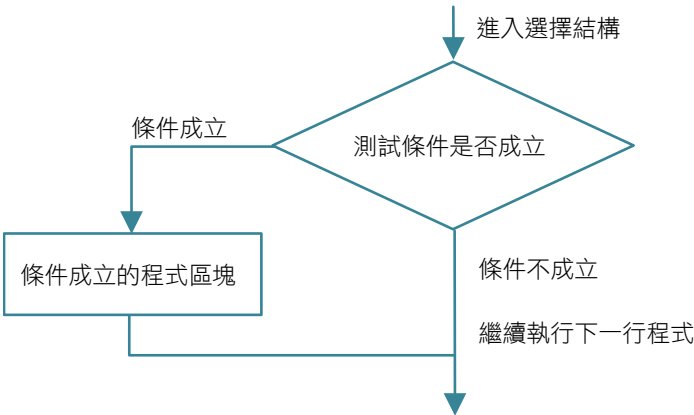
流程圖圖示	意義
	程式流程，表示程式的處理順序，表示 循序結構 。
	條件選擇，於菱形內寫入條件判斷，表示 選擇結構 。

流程圖圖示	意義
	程式敘述區塊，寫出所需完成的功能。
	開始或結束，看到此圖表示程式的開始或結束。
	程式所需的輸入與輸出。

重複結構可由上述元件組合而成，流程圖表示請參閱重複結構章節。

3-2 ▶▶ 單向選擇結構

單向選擇結構是最簡單的選擇結構，日常生活上經常用到，例如：「若週末天氣好的話，我們就去動物園」。單向選擇結構只做測試條件為真時，執行對應的動作，只有一個方向的選擇，因此稱做單向選擇結構。單向選擇結構除了日常生活對話方式表達，還可以使用流程圖與程式語法表達，以下就流程圖與程式語法分別說明。單向選擇使用流程圖表示，如下圖。



單向選擇程式語法	程式範例
<pre>if (條件判斷) { 條件成立的敘述 }</pre>	<pre>if (score >= 60) { cout << "很好，請繼續保持下去"; }</pre>
說明	
若變數 score 大於等於 60，則顯示「很好，請繼續保持下去」。	

有了這樣的概念後，我們就舉個實例說明。

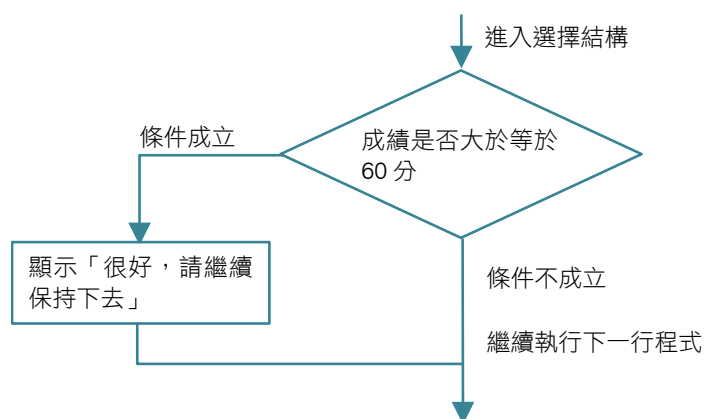
3-2-1 判斷及格(ch3\判斷及格.cpp)

寫一個程式判斷所輸入成績是否及格，成績及格則顯示「很好，請繼續保持下去」。

(a) 解題想法

可以使用單向選擇結構撰寫程式，判斷成績是否及格，及格就顯示「很好，請繼續保持下去」。

流程圖表示如下。



(b) 程式碼與解說

行數	程式碼
1	<code>#include <iostream></code>
2	<code>using namespace std;</code>
3	<code>int main(){</code>
4	<code> int score=98;</code>
5	<code> if (score > 60){</code>
6	<code> cout << "很好，請繼續保持下去" << endl;</code>
7	<code> }</code>
8	<code>}</code>

解說

- 第 4 行：宣告整數變數 `score`，並初始化為 98。

- 第 5 到 7 行：條件判斷(if)對 `score` 做判斷，大於等於 60 分就輸出「很好，請繼續保持下去」。

(c) 預覽結果

按下「執行 → 編譯並執行」，結果顯示在螢幕。

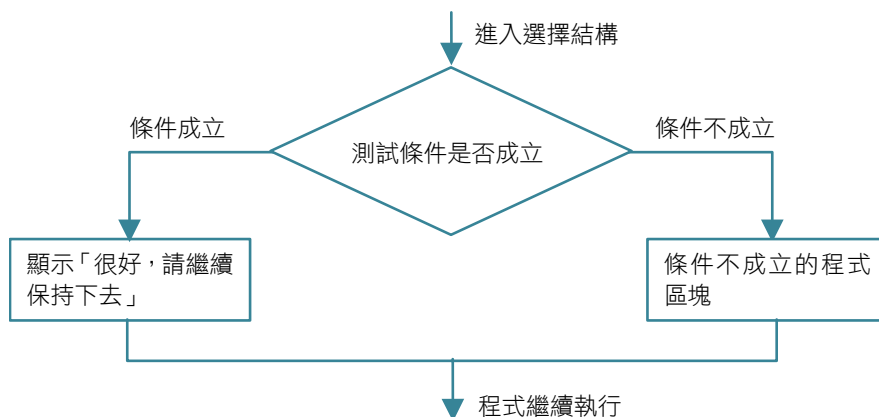
```

F:\C++程式設計入門\ch3\判斷及格.exe
很好，請繼續保持下去
-----
Process exited after 0.01436 seconds with return value 0
請按任意鍵繼續 . . .

```

3-3 ▶▶ 雙向選擇結構

雙向選擇結構比起單向選擇結構更複雜一些，日常生活上屬於雙向選擇的對話，例如：「若週末天氣好的話，我們就出去參觀動物園，否則去看電影」。雙向選擇結構為當測試條件為真時，執行測試條件為真的動作，否則做測試條件為假的動作，有兩個方向的選擇，因此稱做雙向選擇結構。雙向選擇結構也可使用流程圖與程式語法表達，雙向選擇使用流程圖表示，如下圖。



雙向選擇程式語法	程式範例(滿 2000 打九折)
<pre> if (條件判斷) { 條件成立的敘述 }else{ 條件不成立的敘述 } </pre>	<pre> if (cost >= 2000) { cout << cost * 0.9; } else { cout << cost; } </pre>
說明	
若 cost 大於等於 2000，則顯示為 cost 的值打九折，否則顯示 cost 的值。	

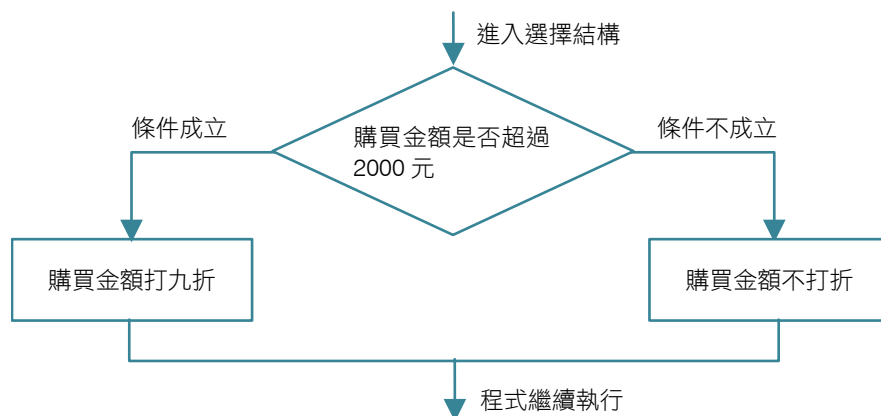
3-3-1 滿 2000 打九折(ch3\滿 2000 打九折.cpp)

採買物品時，有時會遇到店家為了刺激消費，會使用滿額折扣，如：滿 2000 打九折，未滿 2000 則不打折，請寫一個程式幫助店家計算顧客所需付出的金額。

(a) 解題想法

可以使用雙向選擇結構撰寫程式，判斷購買金額是否在 2000 元以上，若購買金額在 2000 元以上，輸出購買金額乘以 0.9；否則依照原價輸出。

流程圖表示如下。



(b) 程式碼與解說

行數	程式碼
1	#include <iostream>
2	using namespace std;
3	int main(){

```

4      int cost=3000;
5      if (cost >= 2000){
6          cout << cost*0.9 << endl;
7      }else{
8          cout << cost << endl;
9      }
10     }

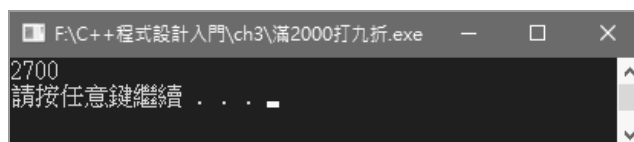
```

解說

- 第 4 行：宣告整數變數 `cost`，並初始化為 3000。
- 第 5 到 9 行：條件判斷(`if`)對 `cost` 做判斷，大於等於 2000 就將該數值打九折(第 5 到 6 行)，否則該數值不打折(第 7 到 9 行)。

(c) 預覽結果

按下「執行 → 編譯並執行」，結果顯示在螢幕。



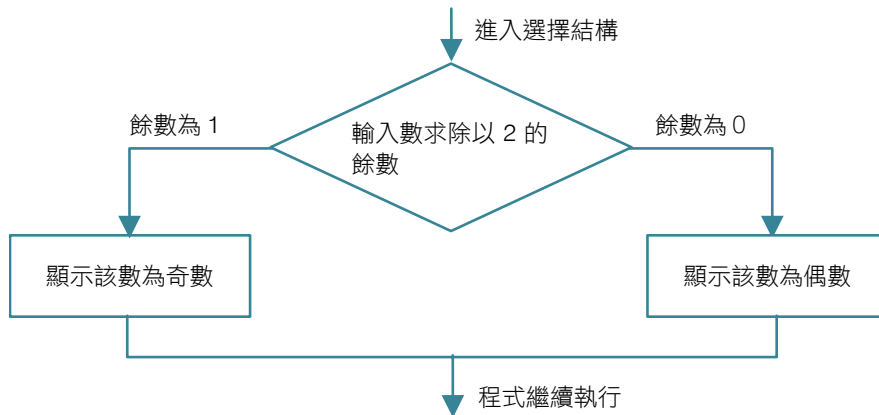
3-3-2 判斷奇偶數(ch3\判斷奇偶數.cpp)

請寫一個程式判斷輸入的值是奇數還是偶數，通常會以求除以 2 的餘數，若餘數為 0 表示輸入的數為偶數，否則輸入的數為奇數。

(a) 解題想法

可以使用雙向選擇結構撰寫程式，判斷輸入值除以 2 的餘數，若餘數為 0，則輸出該數為偶數；否則輸出該數為奇數。

流程圖表示如下。



(b) 程式碼與解說

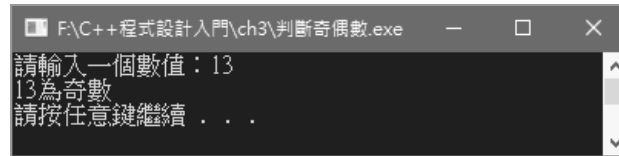
行數	程式碼
1	#include <iostream>
2	using namespace std;
3	int main(){
4	int num;
5	cout << "請輸入一個數值：";
6	cin >> num;
7	if ((num%2) == 0){
8	cout << num <<"為偶數" << endl;
9	}else{
10	cout << num <<"為奇數" << endl;;
11	}
12	}

解說

- 第 4 行：宣告 `num` 為整數變數。
- 第 5 行：顯示「請輸入一個數值：」於螢幕。
- 第 6 行：由鍵盤輸入一整數，儲存到變數 `num`。
- 第 7 到 11 行：利用條件判斷(`if`)對 `num` 做判斷，取 2 的餘數，若餘數為 0 則顯示為偶數(第 7 到 8 行)，否則餘數不為 0，則顯示為奇數(第 9 到 11 行)。

(c) 預覽結果

按下「執行 → 編譯並執行」，輸入一個數字，例如：13，顯示結果在螢幕上。



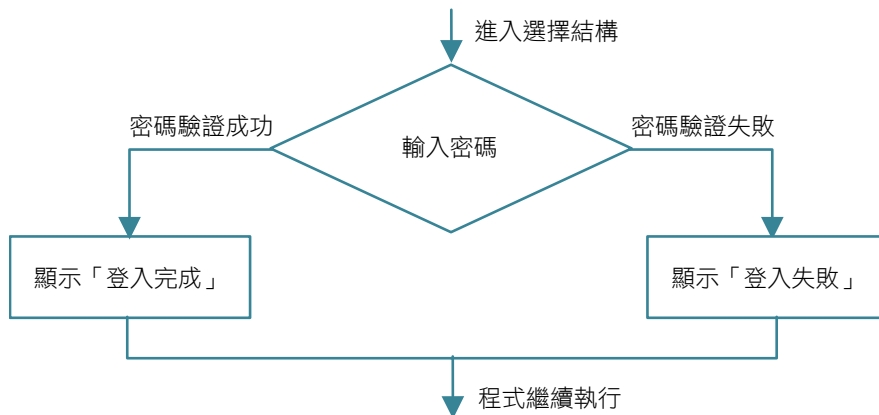
3-3-3 密碼驗證(ch3\密碼驗證.cpp)

請寫一個程式模擬密碼登入，若密碼正確則顯示登入完成，否則顯示登入失敗。

(a) 解題想法

可以使用雙向選擇結構撰寫程式，判斷輸入的密碼是否正確，若密碼正確，則顯示「登入完成」；否則顯示「登入失敗」。

流程圖表示如下。



(b) 程式碼與解說

行數	程式碼
1	#include <iostream>
2	using namespace std;
3	int main(){
4	int num;
5	cout << "請輸入數值密碼：";
6	cin >> num;

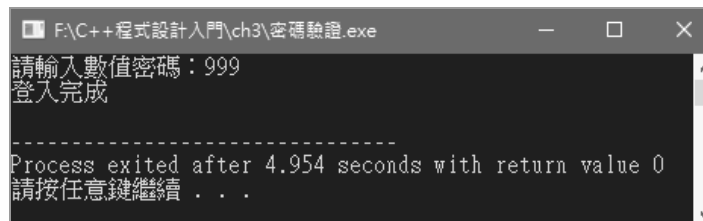

```
7      if (num == 999){
8          cout << "登入完成" <<endl;
9      }else{
10         cout << "登入失敗" <<endl;
11     }
12 }
```

解說

- 第 4 行：宣告整數變數 `num`。
- 第 5 行：螢幕顯示「請輸入數值密碼：」。
- 第 6 行：鍵盤輸入密碼儲存入變數 `num`。
- 第 7 到 8 行：判斷輸入的值是否為「999」，若是則顯示「登入完成」。
- 第 9 到 11 行：否則顯示「登入失敗」。

(c) 預覽程式執行結果

按下「執行 → 編譯並執行」，結果顯示在螢幕。



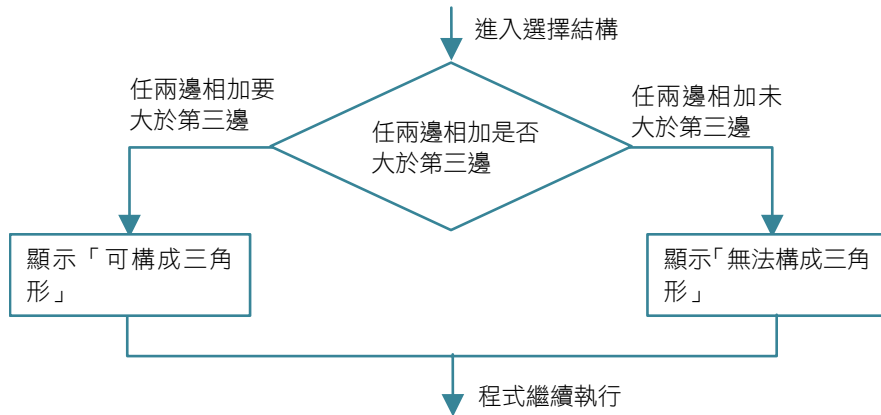
3-3-4 三角形判斷(ch3\三角形判斷.cpp)

設計一個程式允許輸入三角形三邊長，分別為 `a`、`b` 與 `c`，根據三角形中任兩邊相加要大於第三邊，判斷是否為三角形。視窗元件配置如下圖。

(a) 解題想法

可以使用雙向選擇結構撰寫程式，判斷任兩邊相加是否大於第三邊，若任兩邊相加要大於第三邊，則顯示「可構成三角形」；否則顯示「無法構成三角形」。任兩邊相加是否大於第三邊，可以結合關係運算子的大於運算子(`>`)與邏輯運算子的且運算子(`&&`)完成任兩邊相加是否大於第三邊的判斷。

流程圖表示如下。



(b) 程式碼與解說

行數	程式碼
1	#include <iostream>
2	using namespace std;
3	int main(){
4	int a,b,c;
5	cout << "請輸入三角形 a 邊長為：";
6	cin >> a;
7	cout << "請輸入三角形 b 邊長為：";
8	cin >> b;
9	cout << "請輸入三角形 c 邊長為：";
10	cin >> c;
11	if ((a+b>c)&&(a+c>b)&&(b+c>a)){
12	cout << "可構成三角形" <<endl;
13	}else{
14	cout << "無法構成三角形" <<endl;
15	}
16	}

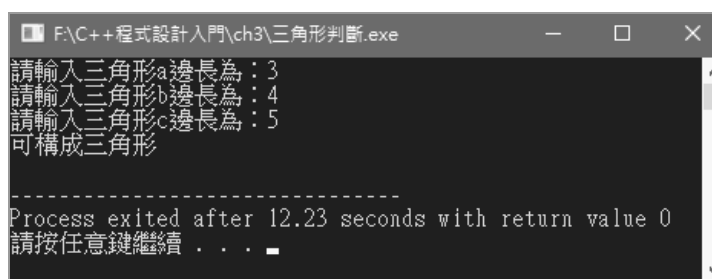
解說

- 第 4 行：宣告整數變數 a、b 與 c。
- 第 5 行：螢幕顯示「請輸入三角形 a 邊長為：」。
- 第 6 行：鍵盤輸入密碼儲存入變數 a。
- 第 7 行：螢幕顯示「請輸入三角形 b 邊長為：」。

- 第 8 行：鍵盤輸入密碼儲存入變數 b 。
- 第 9 行：螢幕顯示「請輸入三角形 c 邊長為：」。
- 第 10 行：鍵盤輸入密碼儲存入變數 c 。
- 第 11 到 12 行：判斷輸入的值是否為 $(a+b>c)$ 且 $(a+c>b)$ 且 $(b+c>a)$ ，若是則顯示「可構成三角形」。
- 第 13 到 15 行：否則顯示「無法構成三角形」。

(c) 預覽程式執行結果

按下「執行 → 編譯並執行」，輸入三邊長分別為 3，4 與 5，結果顯示在螢幕。



```
F:\C++\程式設計入門\ch3\三角形判斷.exe
請輸入三角形a邊長為:3
請輸入三角形b邊長為:4
請輸入三角形c邊長為:5
可構成三角形

-----
Process exited after 12.23 seconds with return value 0
請按任意鍵繼續 . . . -
```

解析 APCS 程式設計觀念題

- (D) 1. 給定右側函式 $F()$ ，已知 $F(7)$ 回傳值為 17，且 $F(8)$ 回傳值為 25，請問 if 的條件判斷式應為何？

(106 年 3 月 APCS 第 16 題)

- (A) $a \% 2 != 1$
- (B) $a * 2 > 16$
- (C) $a + 3 < 12$
- (D) $a * a < 50$

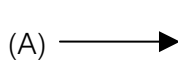

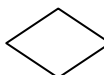

```
int F(int a){
    if(____?____)
        return a * 2 + 3;
    else
        return a * 3 + 1;
}
```

解析

$F(7)$ 回傳值為 17，因此判斷是執行「 $a*2+3$ 」獲得，且 $F(8)$ 回傳值為 25，因此判斷是執行「 $a*3+1$ 」獲得，選項(D) $a*a < 50$ ，是唯一符合此結果的判斷式。

選擇題

- () 1. 下列何者不是程式設計中三個主要結構
(A) 循序結構 (B) 排序結構 (C) 選擇結構 (D) 重複結構
- () 2. 程式中若要寫出「若成績大於 60 分則顯示及格，否則顯示不及格」這樣的結構為
(A) 循序結構 (B) 排序結構 (C) 選擇結構 (D) 重複結構
- () 3. 程式第一行執行完後執行第二行，程式第二行執行完後執行第三行，這樣的結構為
(A) 循序結構 (B) 排序結構 (C) 選擇結構 (D) 重複結構
- () 4. 程式中重複執行相同的動作，這樣的結構為
(A) 循序結構 (B) 排序結構 (C) 選擇結構 (D) 重複結構
- () 5. 流程圖中選擇結構為以下何圖？

(A)  (B)  (C)  (D) 

- () 6. 以下程式執行後，a 等於多少？

```
int a = 3000;
if (a > 2000) {
    a = a * 0.5;
}else{
    a = a * 0.6;
}
```

(A) 1500 (B) 1800 (C) 3000 (D) 2000

- () 7. 以下程式執行後，a 等於多少？

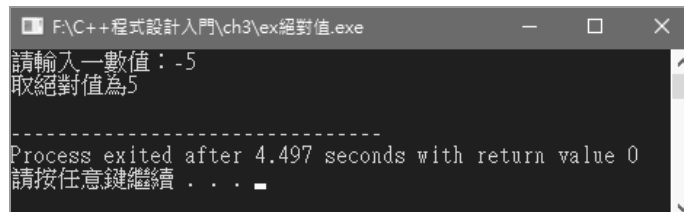
```
int a = 2000;
if (a > 2000) {
    a = a * 0.5;
}else{
    a = a * 0.6;
}
```

(A) 1000 (B) 1800 (C) 1200 (D) 2000

程式實作

1. 絕對值(ch3\ex 絕對值.cpp)

設計程式允許輸入一個數值，求出該數的絕對值，絕對值表示，若該數小於 0，則取該數加上負號，否則該數維持不變。執行結果，如下圖。

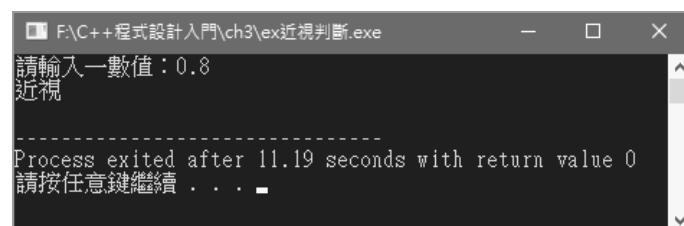


```
F:\C++\程式設計入門\ch3\ex絕對值.exe
請輸入一數值: -5
取絕對值為5

-----
Process exited after 4.497 seconds with return value 0
請按任意鍵繼續 . . .
```

2. 近視判斷(ch3\ex 近視判斷.cpp)

設計程式允許輸入視力測量值，根據測量值判斷是否有近視，若測量值小於 0.9，顯示有近視，否則顯示視力正常。執行結果，如下圖。

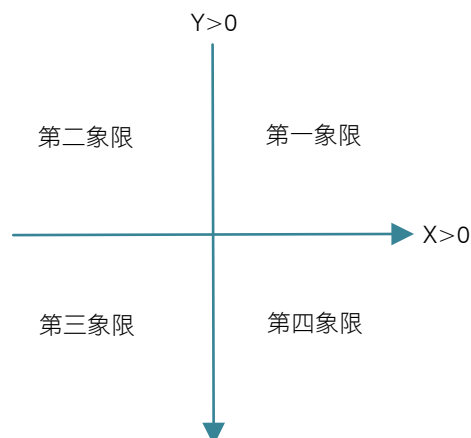


```
F:\C++\程式設計入門\ch3\ex近視判斷.exe
請輸入一數值: 0.8
近視

-----
Process exited after 11.19 seconds with return value 0
請按任意鍵繼續 . . .
```

3. 象限判斷(ch3\ex 象限判斷.cpp)

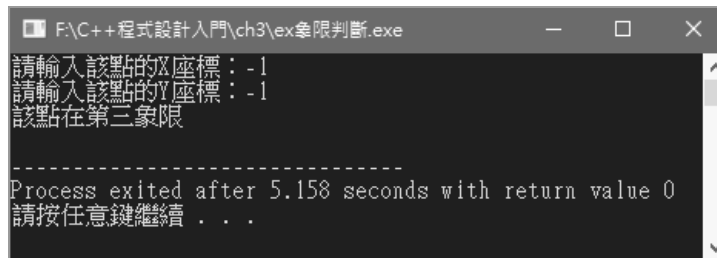
數學將平面象限分成四個象限，平面分成 X 軸與 Y 軸，由 X 軸與 Y 軸切割成四個象限如下圖所示，請寫一個程式輸入平面中某點的 X 值與 Y 值，輸出該點所在象限。



歸納出 X 值與 Y 值與各象限的定義如下。

- (a) 若 $X > 0$ 且 $Y > 0$ ，則在第一象限
- (b) 若 $X < 0$ 且 $Y > 0$ ，則在第二象限
- (c) 若 $X < 0$ 且 $Y < 0$ ，則在第三象限
- (d) 若 $X > 0$ 且 $Y < 0$ ，則在第四象限
- (e) 若 $X = 0$ 或 $Y = 0$ ，則在座標軸上

執行結果，如下圖。

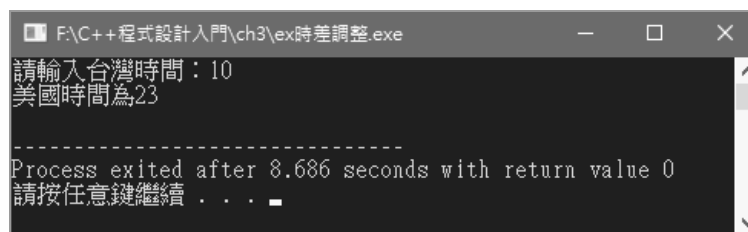


```
F:\C++\程式設計入門\ch3\ex象限判斷.exe
請輸入該點的X座標：-1
請輸入該點的Y座標：-1
該點在第三象限

-----
Process exited after 5.158 seconds with return value 0
請按任意鍵繼續 . . .
```

4. 時差調整(ch3\ex 時差調整.cpp)

小明在美國，而小華在台灣，因為時差的關係假設美國比台灣晚 11 小時，也就是台灣早上 10 點，美國就晚上 23 點，而台灣早上 11 點，美國就晚上 0 點。請以 24 小時表示時間，晚上 24 點請以 0 點表示，也就是只能輸入與輸出都是介於 0 與 23 的整數，請設計一個程式輸入台灣時間，輸出對應的美國時間，時間輸入為整點。執行結果，如下圖。



```
F:\C++\程式設計入門\ch3\ex時差調整.exe
請輸入台灣時間：10
美國時間為23

-----
Process exited after 8.686 seconds with return value 0
請按任意鍵繼續 . . .
```