

版權宣告：此為作者 30 年的教學心得，僅供大家選書參考，更多的內容請購書，這樣作者才有資金繼續研發精彩教材，科技才會進步。

第五章 決策流程指令

5_1 決策流程運算子

5-2 if...else 指令

5_3 switch 指令

5_4 實例探討

5_5 APCS 觀念題

5_6 APCS 實作題

人類的生活必須不斷面對決策問題，連我家一個不到三歲的小孩，也常要思考他手裡的十元是要坐電動車還是買棒棒糖。程式語言是協助解決人類問題的工具，當然也有決策流程敘述，C 語言依決策流程點的多寡，分為以下兩種決策流程敘述，第一是雙向分歧決策流程 if...else...，例如肚子餓了就吃飯，否則繼續前進；第二是多向分歧決策流程的 switch case，例如你身上有 5000 元，走進一家五星級的大飯店用餐，你的分歧點就很多，有自助餐、中式套餐、日本料理、泰國餐點等等分歧點。本章的重點即是探討 C 語言的決策流程敘述。請先鍵入以下程式。

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int a=66;
    char b='P';
    if (a>=60){
        b='P';
    }
    else{
        b='F';
    }
    printf("%c",b);
    return 0;
}
```

以上『if』稱為決策指令，請看 5_2 節，『a>=60』稱為決策運算子，請看 5_1 節。

5_1 決策流程運算子

決策流程運算子需要兩種運算子，分別是關係運算子與邏輯運算子，分別說明如下：

關係運算子(Relational Operators)

關係運算子又稱為比較(Comparison)運算子，用於資料之間的大小比較，比較的結果可得到 int 型態的 1(true)或 0(false)，下表是 C 語言中的關係運算子符號。其中等於與不等於的符號與其它語言有顯然的不同，請讀者留意，例如，下式用來比較 a 與 b 是否相等。

```
if (a == b) ;
```

下面是關係運算子列表：

運算子	定義	優先順序	結合律
<	小於	6	由左至右
>	大於	6	由左至右
<=	小於等於	6	由左至右
>=	大於等於	6	由左至右
==	等於	7	由左至右
!=	不等於	7	由左至右

例如，

```
int a=8,b=5;
printf("%d\n",a>b);//1
printf("%d\n",a==b);//0
printf("%d\n",a>=b);//1
printf("%d\n",a<=b);//0
printf("%d\n",a!=b);//1
```

自我練習

- 1、請將以上運算式於 C 環境編譯、執行，觀察執行結果。
- 2、請鍵入以下程式，編譯與修正程式。

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    int a=8,b=5;
    printf("%d\n",a=b);
    printf("%d\n",a>=b);
    printf("%d\n",a<=b);
    printf("%d\n",a!=b);
    return 0;
}
```

邏輯運算子(Logical Operators)

當同一個運算式要同時存在兩個以上的關係運算子時，每兩個關係運算子之間必須使用邏輯運算子連結，例如，您要找男生且年齡大於 40，此一選擇就同時含有兩個關係運算式，此時就要使用邏輯運算子。C 的邏輯運算子有否定 not 『!』、邏輯 and 『&&』、邏輯 or 『||』，如下表所示：

運算子	定義	優先順序	結合律
!	邏輯否定運算	2	由右至左
&&	邏輯 and 運算	11	由左至右
	邏輯 or 運算	12	由左至右

邏輯否定 not 是將 『true』 轉為 『false』，『false』 轉為 『true』；邏輯 and 是兩件事都 true，結果才是 true，其餘都是 false，其真值表如下：

事件 1	事件 2	結果
true	true	true
true	false	false
false	true	false
false	false	false

邏輯 or 是兩件事只要有一件為 true，那就為 true，只有兩件事全為 false，才為 false，其真值表如下：

事件 1	事件 2	結果
true	true	true
true	false	true
false	true	true
false	false	false

例如：

```
int a=8,b=5,c=3;
printf("%d\n",!(a>b));//0
printf("%d\n",!(a==b));//1
printf("%d\n",a>b && b>c );//1
printf("%d\n",a>b && b==c );//0
printf("%d\n",a>b || b==c );//1
printf("%d\n",a==b || b==c );//0
```

範例 5_1a

若有一數學式，判斷 x 是否滿足 $1 < x \leq 6$ ，請轉換為 C 語言敘述。

提示：判斷 x 是否在 1 到 6 之間，數學語言是

$1 < x \leq 6$

C 語言是

```
int x=3;
printf("%d",x>1 && x<=6);
```

或

```
x=9;printf("%d",x>1 && x<=6); )
```

所以完整 C 語言表示如下：

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    int x=3;
    if(x>1 && x<=6)
        printf("True");
    else
        printf("False");
    return 0;
}
```

範例 5_1b

輸入三角形三邊長 a,b,c，圍成三角形的條件是，任兩邊之和要大於第三邊，那如何以 C 語言表示？

提示：任兩邊之和大於第三邊的數學語言是：

$a+b > c$ and $a+c > b$ and $b+c > a$

C 語言是

```
if ((a+b>c) &&(b+c>a) &&(c+a>b))
```

完整程式如下：

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    //int a=3,b=4,c=5;
    int a=1,b=4,c=6;
    if ((a+b>c) &&(b+c>a) &&(c+a>b))
        printf("True");
    else
        printf("False");
    return 0;
}
```

自我練習

1、若有一數學式，判斷 x 是否滿足 $x > 3$ or $x < -2$ ，請轉換為 C 語言敘述。

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int x=6;
    //int x=1;
    if (x>3 || x<-2)
        printf("True");
    else
        printf("False");
    return 0;
}
```

2、若有一數學式，同時判斷六個變數是否滿足 $\frac{a_1}{a_2} = \frac{b_1}{b_2} = \frac{c_1}{c_2}$ ，請轉換為 C 敘述。

那 $\frac{a_1}{a_2} = \frac{b_1}{b_2} \neq \frac{c_1}{c_2}$ 又如何表示呢？

```
if ((a1/b1)==(a2/b2) & (a2/b2)==(a3/b3))
    ;
if ((a1/b1)==(a2/b2) & (a2/b2)!= (a3/b3))
    ;
```

5-2 if...else

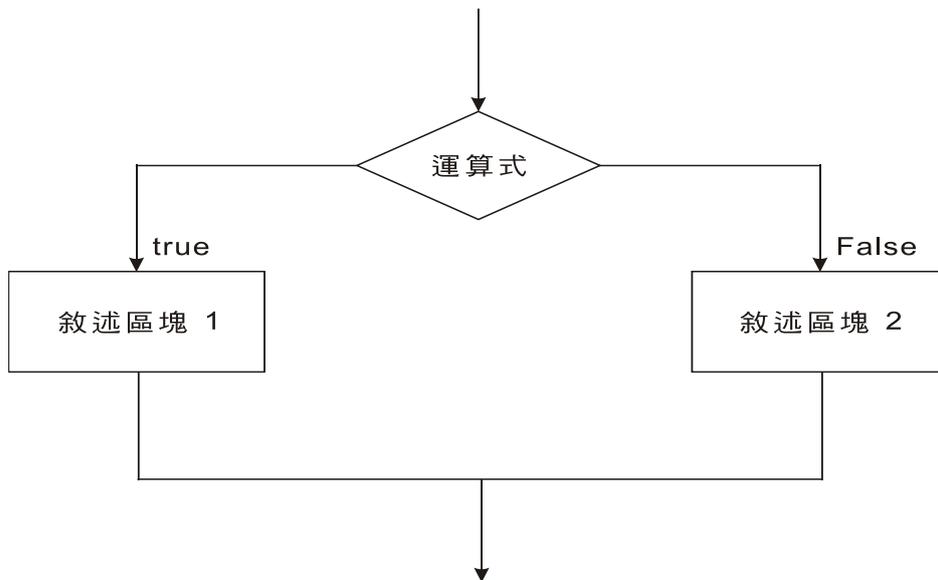
在日常生活領域中，當出現“假如～則～，否則～”時，此種決策流程模式有兩種解決問題的方案，故稱為雙向分歧決策流程，此時可使用 if...else 敘述。

if...else 敘述的語法如下：

```
if (運算式)
{
    敘述區塊 1；
}
[else
{
    敘述區塊 2；
}]
```

以上語法說明如下：

1. 運算式的值若為 1(true)，則執行敘述區塊 1；運算式的值若為 0(false)，則執行敘述區塊 2，其流程圖如下：



2.敘述區塊上下一定要用大括號({})包圍。例如，以下敘述可依 a 的大小評量其及格與否。

```

if (a>=60){
    r='P';
}
else{
    r='F';
}
  
```

3.敘述區塊內的敘述若只有一個，則敘述區塊上下兩個大括號可予省略。例如，以上程式可簡化如下：

```

if (a>=60)
    r='P';
else
    r='F';
  
```

4.有時為了簡化程式的撰寫，可將否則的部分寫在 if 前面，並省略 else。例如，以下程式同義於上面程式。

```

r='F';
if(a>=60) {
    r='P';
}
  
```

5. 若敘述區塊上下的大括號遺漏，則條件成立時，僅執行敘述區塊的第一個敘述，但更嚴重的問題，敘述區快第二個以後的敘述，不論條件成立與否，均會自動執行。例如，以下敘述執行之後，b=0，c=3。

```

a=2;b=0;c=0;
  
```

```

if(a==0)
    b=3;
    c=3;
printf("b=%d,c=%d",b,c);      /* b=0, c=3 */

```

6. if()後面不要緊接分號，例如，

```

a=2;b=0;c=0;
if(a==0);
    b=3;
    c=3;
printf("b=%d,c=%d",b,c);      /* b=3, c=3 */

```

這是初學者常犯的錯誤，因為 if 遇到分號（；），表示此指令已經結束。

7. 敘述區塊內可以放置任何合法敘述，當然也可以再放置 if；if 中有 if，稱為巢狀 if，請看以下範例。

8. C語言為了強調語法的簡潔性，若決策流程的結果，只為求得簡單的運算式，則可用以下敘述的三元運算子（？：），

```

r=(運算式 1)?(運算式 2):(運算式 3)

```

其中『運算式 2』與『運算式 3』的括號只是增加程式的可讀性，此兩括號的有無並不影響程式的執行結果。當運算式 1 的值为 1(true)時，z=運算式 2，當運算式 1 的值为 0(false)時，則 Z=運算式 3，例如，以下敘述亦可評定成績及格與否。

```

r=(a>=60)?('P'):( 'F');

```

自我練習

1、請鍵入以上程式，並觀察執行結果。例如，第 2 點的程式如下，請自行指派 a 與輸出。

```

#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    int a;
    char r;
    //2
    a=56;
    if (a>=60){
        r='P';
    }
    else{
        r='F';
    }
}

```

```

printf("%c\n",r);
return 0;
}

```

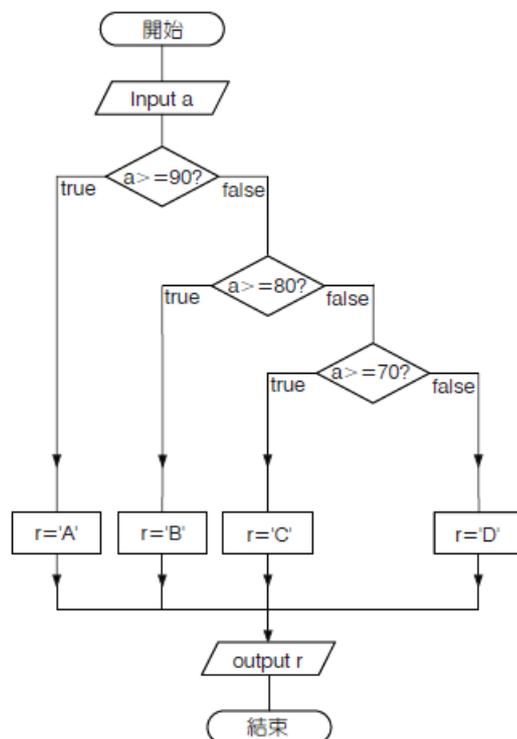
範例 5-2a

請寫一個程式，完成以下要求：

1. 輸入一個 0~100 的分數。
2. 當分數大於 90 分時，輸出 A。
3. 當分數介於 80~89 時，輸出 B。
4. 當分數介於 70~79 時，輸出 C。
5. 當分數介於 0~69 時，輸出 D。

【流程分析】

1. 使用流程圖分析如下：



2. 以上每一個決策流程點，都有兩個分歧點，所以適用 if~else。
3. 每一個 else 後面均需進一步決策流程，所以可在每一 else 後面再放置 if。

【執行結果】

g502

【程式列印】

```
#include <stdio.h>
```

```

#include <stdlib.h>
int main(int argc, char *argv[]) {
    int a;
    char r;
    printf("input a grade: ");
    scanf("%d",&a);//&
    if(a>=90)          /* 高於 90 分為 A */
        r='A';
    else
        if(a>=80)     /* 介於 80 與 90 分為 B */
            r='B';
        else
            if(a>=70)  /* 介於 70 與 80 分為 C */
                r='C';
            else
                r='D';    /* 不符合上述情況則為 D */
    printf("the grade is %c",r);
    return 0;
}

```

補充說明

- 1、此題有人會寫成 `if(90<a<100)`，但 C 並沒有這種運算式，因為 `90<a<100` 是數學語言，不是程式語言。
- 2、有人會這樣寫，這樣也可以，這樣可以減少一直縮排。

```

#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    int a;
    char r;
    printf("input a grade: ");
    scanf("%d",&a);//&
    if(a>=90)          /* 高於 90 分為 A */
        r='A';
    else if(a>=80)     /* 介於 80 與 90 分為 B */
        r='B';
    else if(a>=70)     /* 介於 70 與 80 分為 C */
        r='C';
    else

```

```

        r='D';          /* 不符合上述情況則為 D */
    printf("the grade is %c",r);
    return 0;
}

```

3、有人會寫成

```

    if (a<100 & a>=90) b='A';
    if (a<90  & a>=80)  b='B';
    if (a<80  & b>=70)  b='C';
    if (a<70  & b>=0)   b='D';

```

這樣雖然也可以，但是其執行效率非常差，因為不管分數為何，僅有一項會成立，但卻都要進行四次判斷，如此會讓電腦不斷進行無效的判斷。而且，電腦的判斷時間遠大於執行計算的時間。

4、也有人會這樣寫，但這是運算思維錯誤，而且結果也不對。

```

    if(a>=60)
        r='D';
    else
        if(a>=70)
            r='C';
        else
            if(a>=80)
                r='B';
            else
                r='A';

```

3、以下是以 C++ 所開發程式，關於 C++ 請看台科大『高中程式設計與 APCS 先修檢測』。

```

#include <iostream>
using namespace std;
int main()
{
    int a;
    char r;
    cout<<"input a grade: ";
    cin>>a;
    if(a>=90)          /* 高於 90 分為 A */
        r='A';
    else
        if(a>=80)     /* 介於 80 與 90 分為 B */

```

```

        r='B';
    else
        if(a>=70)        /* 介於 70 與 80 分為 C */
            r='C';
        else
            r='D';        /* 不符合上述情況則為 D */
    cout<<"the grade is:"<<r;
    return 0;
}

```

自我練習

1. 請寫一程式，判斷所輸入的數是正數、0 或負數。
2. 某一貨品定價 100 元，若購買 500 件，(含) 以上打 7 折，若購買 499~300 件則打 8 折，若購買 299~100 件則打 9 折，購買 100 件以下則不打折，試寫一程式可以輸入購買件數而得總價。

測試資料：

編號	輸入	
1	20	$100 * 20 = 2000$
2	200	$100 * 200 * 0.9 = 18000$
3	400	$100 * 400 * 0.8 = 32000$
4	500	$100 * 500 * 0.7 = 35000$

3. 直線。直線標準式為 $ax+by+c=0$ ，請寫一程式，可以指派一直線係數 a,b,c。其次，可再輸入任一點座標，並判斷所輸入點式否在直線上。例如，指派 a,b,c 分別是 1,2,-3，那方程式就是 $x+2y-3=0$ ，其次，輸入點若是(1,1)，那就是在直線上，若是(1,2)，那就不在直線上。

```

#include <iostream>
using namespace std;
int main(int argc, char** argv) {
    int a=2,b=-3,c=5;
    int x=-1,y=1;
    char ans='n';
    if (a*x+b*y+c==0)
        ans='y';
    cout<<ans;
    return 0;
}

```

4. 請寫一程式，可以輸入一個小寫字元，請判斷其是否為母音。說明，字元

a, e, i, o, u 稱為母音，其餘為子音。

5. 請寫一個程式，可以產生一個 0 到 25 的亂數，且依以下分數顯示燈號

21~25：五個燈。

16~20：四個燈。

11~15：三個燈。

6~10：兩個燈。

1~5：一個燈。

0：零個燈。

範例 5-2b

請寫一個程式，可以判斷所輸入座標的所在象限。

【執行結果】

g503

【程式列印】

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    int x,y;
    char r;
    printf("input x,y:");
    scanf("%d,%d",&x,&y);//留意&
    if(x>0)
        if(y>0)
            r='1';        /* 第一象限 */
        else
            r='4';        /* 第四象限 */
    else
        if(y>0)
            r='2';        /* 第二象限 */
        else
            r='3';        /* 第三象限 */
    printf("x =%d,y=%d is 第 %c 象限",x,y,r);
    return 0;
}
```

補充說明

1、此題目有人會寫成

```
if (x>0 & y>0) b="I";
if (x<0 & y>0) b="II";
if (x<0 & y<0) b="III";
if (x>0 & y<0 ) b="IV";
```

這樣雖然沒有錯，但是執行效率非常差，因為電腦要不斷的比較。

自我練習

- 1、同上範例，但增加先判斷是否在原點或 x、y 軸上。
- 2、同上題，但程式一執行，要求先輸入密碼，密碼若為數字 1234，那才可執行本程式。

範例 5_2c

請寫一程式，滿足以下條件。

- 1、輸入兩個數。
- 2、求輸入兩數極大值。
- 3、輸出極大值。

【演算法則】

- 1、輸入第一數，本例以變數 a 儲存。
- 2、輸入第二數，本例以變數 b 儲存。
- 3、設定較大值(max)為第一數。max=a
- 4、當第二數(b)大於極大值時，極大值即以 b 取代。

```
if (b>max)
    max=b;
```

- 5、輸出極大值(max)即為所求。

【執行結果】

g504

【程式列印】

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    int a,b,max;
    printf("input a,b:");
    scanf("%d,%d",&a,&b);//留意&
    max=a;
    if (b>max)
        max=b;
    printf("The max is %d",max);
    return 0;
}
```

自我練習

1、請寫一程式，滿足以下條件。

- (1) 輸入三個數。
- (2) 求此三個數的極大值。
- (3) 輸出極大值。

[演算法則]

- (1)輸入第一個數，本例以變數 a 儲存。
- (2)輸入第二個數，本例以變數 b 儲存。
- (3)輸入第三個數，本例以變數 c 儲存。
- (4)設定極大值(max)為第一數 a。
- (5)當第二數(b)大於極大值時，極大值即以 b 取代。

```
if (b>max)
    max = b;
```

- (6)當第三數(c)大於極大值時，極大值即以 c 取代。

```
if (c>max)
    max = c;
```

- (7)輸出極大值 max。

2、請寫一程式，可以輸入 5 個數值，請去掉最大值與最小值，再求其平均。

範例 5_2d

請寫一程式，滿足以下條件。

- 1、輸入三個數。
- 2、將此三個數由小而大輸出。

【演算法則】

1. 分別以 a、b 及 c 表示欲排序的資料。
2. 假如 a 大於 b，則 a 與 b 交換，如下圖的(1)。
3. 假如 b 大於 c，則 b 與 c 交換，如下圖的(2)。
4. 假如 a 大於 b，則 a 與 b 交換，如下圖的(3)排序已完成，共需進行 3 次的比較與交換，如下圖所示。

a	b	c
(1)	(2)	
(3)		

【執行結果】

G505

【程式列印】

```

#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    int a,b,c,t;
    printf("input a,b,c :");
    scanf("%d,%d,%d",&a,&b,&c);//留意&
    if(a>b)
        { t=a; a=b; b=t; }//兩數交換
    if(b>c)
        { t=b; b=c; c=t; }
    if(a>b)
        { t=a; a=b; b=t; }
    //output
    printf("a,b,c 由小而大排列是 %d,%d,%d ",a,b,c);
    return 0;
}

```

[補充說明]

1. 若有 4 筆資料需要排序，則共需進行 6 次比較與交換，如下圖所示。

a	b	C	D
(1)	(2)	(3)	
(4)	(5)		
(6)			

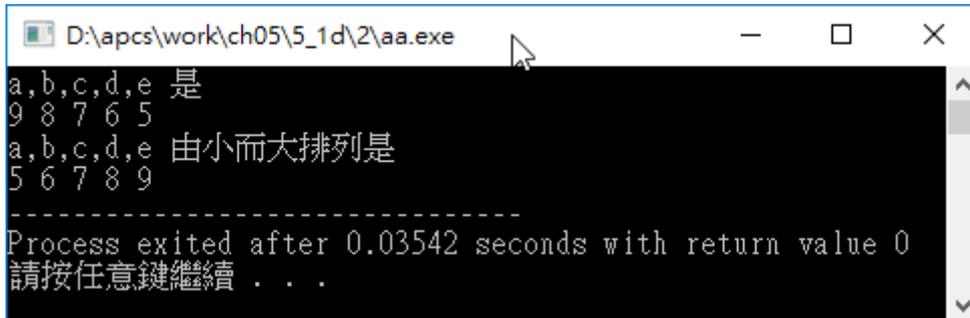
2. 若有 5 筆資料需排序，則共需進行 10 次比較與交換，如下圖所示，此即為氣泡排序法。

a	b	c	d	e
(1)	(2)	(3)	(4)	
(5)	(6)	(7)		
(8)	(9)			
(10)				

3. 以上氣泡排序法，處理 3、4 或 5 筆資料的比較與排序，其比較與交換的次數尚可克服。但是，若欲排序的資料超過 5 個，例如 20 筆資料欲排序，那要如何呢？請大家不用擔心，待迴圈與陣列資料結構介紹以後，程式就有精簡的寫法。
4. 測試排序程式是否正確，測試資料通通要給最惡劣的情況。例如，資料要改由由小而大，那測試資料就要由大而小給，例如，9,8,7,6,5，這樣才能測出程式是否正確。也就是最惡劣都能解了，那就沒有不能完成的資料。反之，

您沒給最惡劣的資料，那程式是否能克服最糟的情況，就不得而知。例如，本例測試資料若是 1,3,2,5,8，這樣就不行，因為這不是最糟的情境。

1、將5個數由小而大排序輸出。



```
D:\apcs\work\ch05\5_1d\2\aa.exe
a,b,c,d,e 是
9 8 7 6 5
a,b,c,d,e 由小而大排列是
5 6 7 8 9
-----
Process exited after 0.03542 seconds with return value 0
請按任意鍵繼續 . . .
```

2、

```
#include <iostream>
using namespace std;
int main(int argc, char** argv) {
    int a=9,b=8,c=7,d=6,e=5,t;
    cout<<"a,b,c,d,e 是 " <<endl;
    cout<<a<<" ";
    cout<<b<<" ";
    cout<<c<<" ";
    cout<<d<<" ";
    cout<<e<<" " <<endl;
    //I
    if(a>b) {
        t=a; a=b; b=t;
    }
    if(b>c) {
        t=b; b=c; c=t;
    }
    if(c>d) {
        t=c; c=d; d=t;
    }
    if(d>e) {
        t=d; d=e; e=t;
    }
    //II
    if(a>b) {
        t=a; a=b; b=t;
    }
    if(b>c) {
        t=b; b=c; c=t;
    }
    if(c>d) {
        t=c; c=d; d=t;
    }
    //III
    if(a>b) {
        t=a; a=b; b=t;
    }
    if(b>c) {
```

```

        t=b; b=c; c=t;
    }
    //IV
    if(a>b) {
        t=a; a=b; b=t;
    }
    //output
    cout<<"a,b,c,d,e 由小而大排列是 "<<endl;
    cout<<a<<" ";
    cout<<b<<" ";
    cout<<c<<" ";
    cout<<d<<" ";
    cout<<e<<" ";
    return 0;
}

```

- 3、請寫一程式，可以輸入四個人名與其分數，並依照分數由小而大輸出，輸出應含人名與分數。（因為字串還沒介紹，本例假設人名僅為一個字元）
本例的重點要讓學生練習，分數交換時，人名也要交換。

```

D:\apcs\work\ch05\5_1d\aa.exe
a,b,c,d 是
aa, 9
bb, 8
cc, 7
dd, 6
a,b,c,d 由小而大排列是
dd, 6
cc, 7
bb, 8
aa, 9

```

4、

```

#include <iostream>
#include <cstring> //字串較字元陣列好用
using namespace std;
int main(int argc, char** argv) {
    int a=9,b=8,c=7,d=6,t;
    string an="aa",bn="bb",cn="cc",dn="dd",tn;
    cout<<"a,b,c,d 是 "<<endl;
    cout<<an<<" , "<<a<<endl;
    cout<<bn<<" , "<<b<<endl;
    cout<<cn<<" , "<<c<<endl;
    cout<<dn<<" , "<<d<<endl;
    //I
    if(a>b) {
        t=a; a=b; b=t;

```

```

        tn=an; an=bn; bn=tn;
    }
    if(b>c)    {
        t=b; b=c; c=t;
        tn=bn; bn=cn; cn=tn;
    }
    if(c>d)    {
        t=c; c=d; d=t;
        tn=cn;cn=dn; dn=tn;
    }
    //II
    if(a>b) {
        t=a; a=b; b=t;
        tn=an; an=bn; bn=tn;
    }
    if(b>c)    {
        t=b; b=c; c=t;
        tn=bn; bn=cn; cn=tn;
    }
    //III
    if(a>b) {
        t=a; a=b; b=t;
        tn=an; an=bn; bn=tn;
    }
    //output
    cout<<"a,b,c,d 由小而大排列是 "<<endl;
    cout<<an<<" , "<<a<<endl;
    cout<<bn<<" , "<<b<<endl;
    cout<<cn<<" , "<<c<<endl;
    cout<<dn<<" , "<<d<<endl;
    return 0;
}

```

5-3 switch...case

一個決策流程點若同時擁有三個或三個以上的解決方案，則稱此為多向分歧決策流程。多向分歧決策流程雖也可使用 5-1 的巢狀 if else 解決，但卻增加程式的複雜度及降低程式可讀性，若此一決策流程點能找到適當的運算式，能使問題同時找到分歧點，則可使用 switch case 敘述。switch case 語法如下：

```

switch (運算式)
{
    case 常數 1 :
        敘述區塊 1 ;
        break;
    case 常數 2 :
        敘述區塊 2 ;
        break;
    case 常數 3 :
        敘述區塊 3 ;
        break;

    [ default :
        敘述區塊 n ; ]
}

```

以上語法說明如下：

1. switch 的運算式值僅能為『**整數**』或『**字元**』。
2. case 的常數僅能『**整數**』或『**字元**』，且其型態應與上面的 switch 運算式相同。
3. 電腦將會依 switch 的運算式值，逐一至常數 1、常數 2 尋找合乎條件的 case，並執行相對應的敘述區塊，直到遇到 **break** 敘述，才能離開 switch。
4. default 可放置特殊情況，也就是沒有適當的 case，則執行 default。若省略 default，且若沒有任何 case 滿足 switch 運算式，則程式會默默離開 switch 敘述。語法中，兩旁加中括號表示此敘述可省略。
5. 敘述區塊可放置任何合法的敘述，當然也可放置 switch 或 if。
6. 以下敘述，可將 1、2、3、4 轉為字元 'A'、'B'、'C'、'D'。

```

int main(int argc, char *argv[]) {
    int a=1;
    char r;
    switch(a)
    {
        case 1:
            r='A';
            break;
        case 2:
            r='B';
            break;
        case 3:
            r='C';
            break;
        case 4:

```

```

        r='D';
        break;
    default :
        r=' ';
    }
    printf("%c",r);
    return 0;
}

```

7. 有些語言可用逗號將兩種 case 放在一起，但在 C/C++ 語言中每一 case 僅能放置一個常數，所以若兩個或兩個以上 case，有相同的處理方法，則應將兩個 case 分成兩個敘述

```

    case 10:

```

```

    case 9:

```

```

        r='A';

```

```

        break;

```

請看以下範例。

範例 5-3a

試以 switch case 重作範例 5-1a。

【執行結果】

s18

【程式列印】

2. 使用 return 可提早離開 main() 函式。

```

#include <stdio.h>

```

```

#include <stdlib.h>

```

```

int main(int argc, char *argv[]) {

```

```

    int a;

```

```

    char r;

```

```

    printf("input a grade: ");

```

```

    scanf("%d",&a);//&

```

```

    if(a>100 || a<0) { // 如果輸入的成績大於 100 或小於 0，則 提早離開

```

```

        return 0;

```

```
}  
switch(a/10) {  
    case 10:  
    case 9:  
        r='A';  
        break;  
    case 8:  
        r='B';  
        break;  
    case 7:  
        r='C';  
        break;  
    case 6:  
    case 5:  
    case 4:  
    case 3:  
    case 2:  
    case 1:  
    case 0:  
        r='D';  
        break;  
}  
printf("the grade is %c",r);  
return 0;  
}
```

自我練習

1、請寫一程式，將所輸入的 0、1、2...6，轉為'S'、'M'、'T'、'W'、'T'、'F'、'S'等字元。

2. 請寫一個程式，可以產生一個 0 到 25 的亂數，且依以下分數顯示燈號

21~25：五個燈。

16~20：四個燈。

11~15：三個燈。

6~10：兩個燈。

1~5：一個燈。

0：零個燈。

範例 5_3b 電子琴。請設計一個電子琴程式，分別使用數字鍵的 1,2,3,4,5,6,7,8 等鍵，代表 DO,Re,Mi ,Fa,So,La,Si ,Lo。

提示：scanf()輸入字元還要按『Enter』，這樣不像電子琴，所以線上查尋，發現 getche()（放在 conio.h）不用，所以本例使用此函式輸入字元。

程式列印

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h> //本例使用 getche
int main(int argc, char *argv[]) {
    char a;
    while(1){ //無限迴圈，請看下一章
        a= getche(); //輸入後，不用按 Enter
        switch(a) {
            case('1'):
                _beep(523,500); //Do
                break;
            case ('2'):
                _beep(587,500); //Le
                break;
            case ('3'):
                _beep(659,500); //Mi
                break;
            case ('4'):
                _beep(698,500); //Fa
                break;
            case ('5'):
                _beep(784,500);
                break;
            case ('6'):
                _beep(880,500);
```

```
        break;
    case ('7'):
        _beep(988,500);
        break;
    case ('8'):
        _beep(1046,500);
        break;
    }
}
return 0;
}
```

自我練習

- 1、自己找一個簡譜來彈奏。
- 2、作出有升 Key（黑鍵）的電子琴鍵盤。
- 3、電子琴教學機。KTV 的影片就是先輸出節奏與歌詞，這樣就是歌唱教學機。本例若先輸出簡譜，提醒使用者該按哪一鍵，那就是電子琴教學機了，請自行練習。

5_4 實例探討

更多的內容，請購買書籍，作者才有資金繼續研發精彩教材。

4_4 APCS 觀念題

- 1、

右側程式執行過後所輸出數值為何？

- (A) 11
- (B) 13
- (C) 15
- (D) 16

```
void main () {
    int count = 10;
    if (count > 0) {
        count = 11;
    }
    if (count > 10) {
        count = 12;
        if (count % 3 == 4) {
            count = 1;
        }
        else {
            count = 0;
        }
    }
    else if (count > 11) {
        count = 13;
    }
    else {
        count = 14;
    }
    if (count) {
        count = 15;
    }
    else {
        count = 16;
    }

    printf ("%d\n", count);
}
```

105/03/16

參考解答

1、本題第一個重點

```
int count;
if(count) {
    count = 15;
}
```

count 若為 0 則為『False』,否則全為『True』

2、第二個重點，本題共三個 if 指令，考試沒電腦，用人腦追蹤如下，斜粗體是執行的指令。

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int count = 10;
    //1
```

```
if (count > 0) {
    count = 11; //11
}
printf ("1.count=%d\n", count); //11
//2
if (count > 10) {
    count = 12;
    if (count % 3 == 4) {
        count = 1;
    }
    else {
        count = 0; //0
    }
}
else if (count > 11) {
    count = 13;
}
else {
    count = 14;
}
printf ("2.count=%d\n", count); //0
//3
if (count) { //0
    count = 15;
}
else {
    count = 16; //16
}
printf ("3.count=%d\n", count); //16
}
```

所以答案是 (D)。

2、

右側 switch 敘述程式碼可以如何以 if-else 改寫？

```
switch (x) {
    case 10: y = 'a'; break;
    case 20:
    case 30: y = 'b'; break;
    default: y = 'c';
}
```

- (A) `if (x==10) y = 'a';`
`if (x==20 || x==30) y = 'b';`
`y = 'c';`
- (B) `if (x==10) y = 'a';`
`else if (x==20 || x==30) y = 'b';`
`else y = 'c';`
- (C) `if (x==10) y = 'a';`
`if (x>=20 && x<=30) y = 'b';`
`y = 'c';`
- (D) `if (x==10) y = 'a';`
`else if(x>=20 && x<=30) y = 'b';`
`else y = 'c';`

105/10/02

參考解答

本例 case 20:沒有 break，所以與 case 30:共用程式，所以答案選 (B)。

3、

右側是依據分數 *s* 評定等第的程式碼片段，

正確的等第公式應為：

90~100 判為 A 等

80~89 判為 B 等

70~79 判為 C 等

60~69 判為 D 等

0~59 判為 F 等

這段程式碼在處理 0~100 的分數時，有幾個分數的等第是錯的？

```
if (s>=90) {
    printf ("A \n");
}
else if (s>=80) {
    printf ("B \n");
}
else if (s>60) {
    printf ("D \n");
}
else if (s>70) {
    printf ("C \n");
}
else {
    printf ("F\n");
}
```

(A) 20

(B) 11

(C) 2

(D) 10

105/10/09

參考解答

- 1、本例 `if(s>60)` 順序錯誤，太早出現，所以 79,78,77,76,75,74,73,72,71,70 會誤判。
- 2、其次 60 被誤判，`if(s>60)` 應該調整為 `if(s>=60)`。
- 3、`if(s>70)` 應該調整為 `if(s>=70)`。
- 4、全部程式與修改後的程式如下，請自行鍵入。

```

#include <stdio.h>
#include <stdlib.h>
int main()
{
    for (int s=100 ;s>=0;s--){
        printf("s=%d: ",s);
        if (s>=90) {
            printf ("A \n");
        }
        else if (s>=80) {
            printf ("B \n");
        }
        else if (s>60) {
            printf ("D \n");
        }
        else if (s>70) {
            printf ("C \n");
        }
        else {
            printf ("F\n");
        }
    }
    printf("\n");
    for (int s=100 ;s>=0;s--){
        printf("s=%d: ",s);
        if (s>=90) {
            printf ("A \n");
        }
        else if (s>=80) {
            printf ("B \n");
        }
        else if (s>=70) {
            printf ("D \n");
        }
        else if (s>=60) {
            printf ("C \n");
        }
        else {

```

```

        printf ("F\n");
    }
}
return 0;
}

```

5_6 APCS 實作題

更多的內容，請購買書籍，作者才有資金繼續研發精彩教材。

自我練習

題號	題目	
1	<pre> int main(int argc, char** argv) { int a=5,b=4,t; if (a>b) t=a;a=b;b=t; printf("%d",a); return 0; } </pre>	
2	<pre> //請留意else 是預設最接近的if int main(int argc, char** argv) { //The else default to the nearest int a=-5,b=4,t=0; if (a>0) if (b>0) t++; else t--; printf("%d\n",t); a=-5;b=4;t=0; if (a>0) if (b>0) t++; else t--; printf("%d\n",t); //If you want to change it,you need to use the curly brackets. a=-5;b=4;t=0; if (a>0){ if (b>0) </pre>	

	<pre> t++; } else t--; printf("%d\n",t); return 0; } </pre>	
--	--	--

習題

1. 假設所得稅稅率累進法則如下：

- (1) 淨所得 30 萬以下繳納 6%。
- (2) 淨所得 30~80 萬之間，則前面 30 萬繳納 6%，超過 30 萬的部分繳納 13%。
- (3) 淨所得在 80~200 萬之間繳納 21%。(前面 30 萬繳 6%，30~80 萬之間繳 13%)
- (4) 淨所得超過 200 萬，超過的部分繳納 30%。

試寫一程式可以輸入淨所得，並計算應繳納稅額。例如淨所得若為 40 萬，則其納稅金額計算如下：

$$300000 \times 6\% + 100000 \times 13\% = 31000$$

若淨所得為 100 萬，則納稅金額如下：

$$300000 \times 6\% + 500000 \times 13\% + 200000 \times 21\% = 125000$$

2. 假設自來水費率如下：

- A、100 度以下，每度 3 元。
- B、100~300 度，超過 100 度的部分，每度 5 元。
- C、300 度以上，超過 300 度的部分，每度 6 元。

根據以上條件，寫一程式，可輸入用水度數，得到水費。

3. 假設計程車的計費方式如下：

1000 公尺以內 60 元。超過 1000 公尺時，

- (1) 日間以每 500 公尺加收 6 元，不足 500 公尺時，以 500 公尺計算。
 - (2) 夜間以每 300 公尺加收 6 元，不足 300 公尺，以 300 公尺計算
- 請根據以上條件，寫一程式完成車資的計算。

4. 寫一程式輸入 x 值，並印出其所對應的值，其函數對應如下：

$$y=f(x)= \begin{cases} x+3 & x>3 \\ x^2 & 1 \leq x \leq 3 \\ \sqrt{x} & 0 < x < 1 \\ 0 & x \leq 0 \end{cases}$$

5.請寫一程式，可以輸入三點座標，輸出其面積。提示:三角形面積公式如下：

 提示：三角形面積公式如下：若面積為零，則表示共線。

$$\frac{1}{2} \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{vmatrix} = \frac{1}{2} (x_1y_2 + x_2y_3 + x_3y_1 - x_2y_1 - x_3y_2 - x_1y_3)$$

6、同上題，任意輸入 3 點，檢查是否共線，若未共線，輸出其面積。

提示：任 3 點面積若為 0，則共線。

7、判斷任意點 D 是否在三角形 ABC 內或外。已知 ABC 三點座標，請寫一程式，可以輸入 D 點座標，並判斷任一點 D 是否在三角形 ABC 內或外。

提示：這一題目，幾何有幾何的解法，但若將點的位置座標化，那就更簡易了，所以您將發現，有了電腦，將資料數位化，很多事就有不同的解法。本例，D 若在三角形內，則三角形 ABC 面積 = DAB + DBC + DAC 的面積)

8、凸或凹四邊形。已知 ABCD 四點座標，請寫一程式，可以判斷四邊形 ABCD 是否為凸或凹四邊形。

(提示：若為凸四邊形，則任一點，均不得在任三點所圍成的三角形內)

9、同上題，若為凸四邊形，並計算其面積。提示：以下是 3 邊形公式，任意凸多邊行，都可以將以下公式推廣，並且代入。

 提示：三角形面積公式如下：若面積為零，則表示共線。

$$\frac{1}{2} \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{vmatrix} = \frac{1}{2} (x_1y_2 + x_2y_3 + x_3y_1 - x_2y_1 - x_3y_2 - x_1y_3)$$

9.於二維平面中，請寫一程式，可以判斷兩直線平行、或相交或重合，若平行計算其距離，相交計算其交點。

10.寫一程式，可以任意輸入凸四邊形的四個端點座標，並可將其按順時針方向排列此四個點。

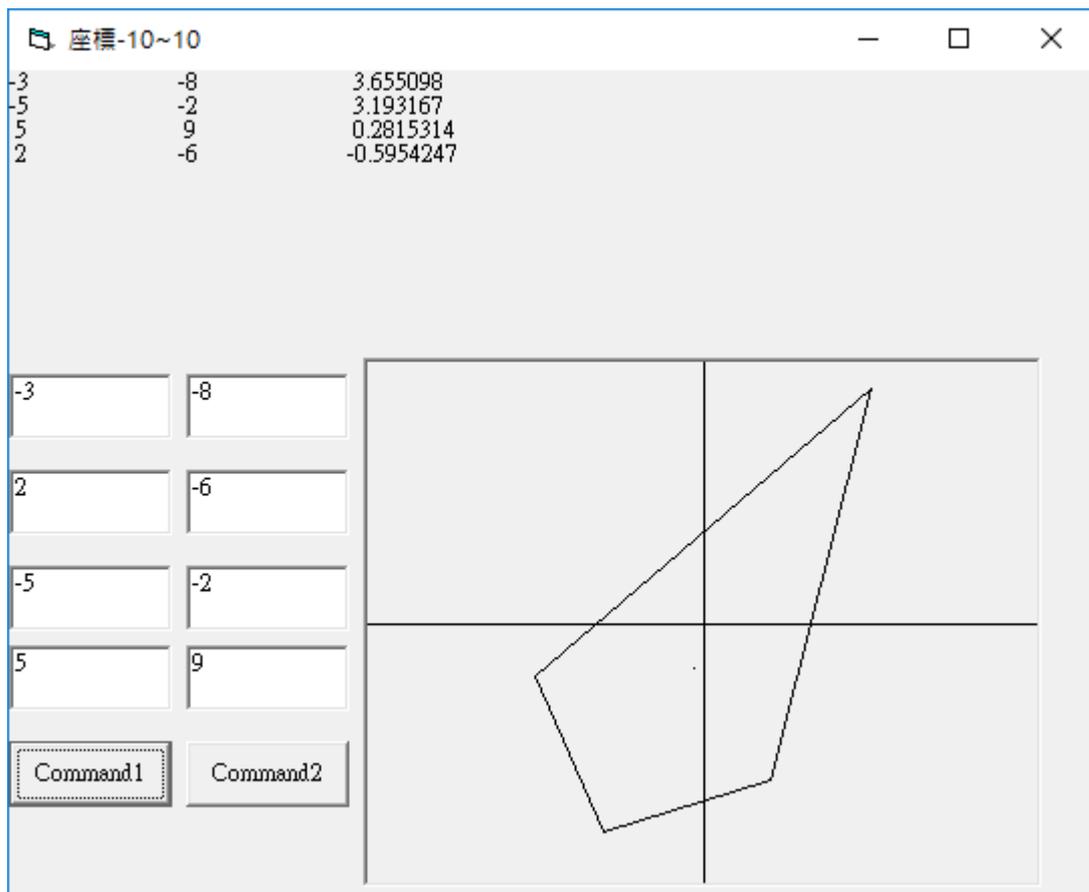
解

1、求出四边形的中心点坐标。

2、以中心点为原点，求出四个点的極座標的角度 s。

3、以極座標的角度 s 排序，即可從新按照順序排列。

3、VB 可方便繪圖，所以以 VB 實現如下：



```
Private Sub Command1_Click()
```

```
    PI = 4 * Atn(1)
```

```
    Dim a(4, 2) As Integer
```

```
    Dim b(4, 2) As Single
```

```
    Dim n As Integer
```

```
    n = 4
```

```
    pic.Cls
```

```
    Cls
```

```
    Dim x, y As Single
```

```
    Dim s(10) As Single '斜率
```

```
    '設定座標系統
```

```
    pic.Scale (-10, 10)-(10, -10)
```

```
    '畫出 x 軸
```

```
    pic.Line (-10, 0)-(10, 0)
```

```
    '畫出 y 軸
```

```
    pic.Line (0, 10)-(0, -10)
```

```
    a(1, 1) = Val(Text1.Text)
```

```
    a(1, 2) = Val(Text2.Text)
```

```

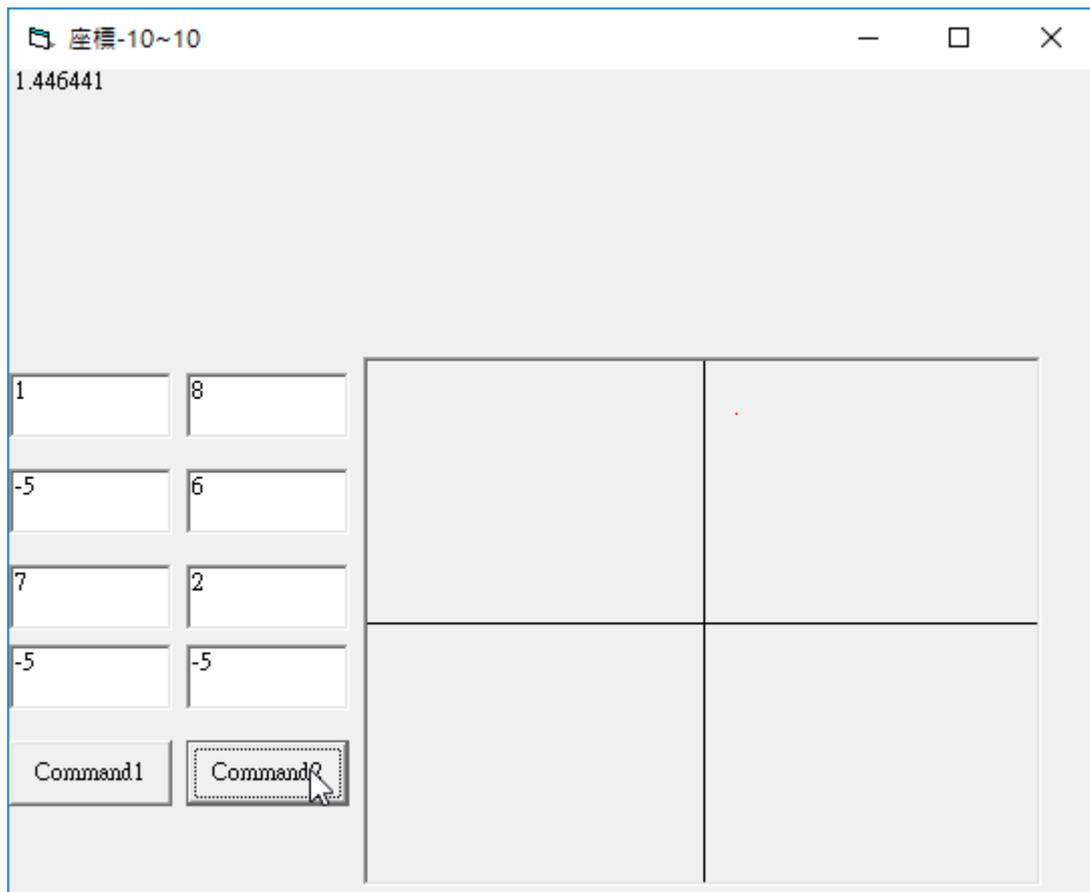
a(2, 1) = Val(Text3.Text)
a(2, 2) = Val(Text4.Text)
a(3, 1) = Val(Text5.Text)
a(3, 2) = Val(Text6.Text)
a(4, 1) = Val(Text7.Text)
a(4, 2) = Val(Text8.Text)
For i = 1 To n
    pic.PSet (a(i, 1), a(i, 2))
Next
'求中心點座標
x = 0: y = 0
For i = 1 To n
    x = x + a(i, 1)
    y = y + a(i, 2)
Next
x = x / n
y = y / n
pic.PSet (x, y)
'求對中心點的斜率，但回傳值僅 PI/2~-PI/2
For i = 1 To n
    b(i, 1) = a(i, 1) - x: b(i, 2) = a(i, 2) - y
    If b(i, 1) = 0 And b(i, 2) >= 0 Then
        s(i) = PI / 2
    Else
        If b(i, 1) = 0 And b(i, 2) < 0 Then
            s(i) = 3 * PI / 2
        Else
            If b(i, 1) > 0 Then
                s(i) = Atn(b(i, 2)) / (b(i, 1))
            Else
                s(i) = Atn(b(i, 2)) / (b(i, 1)) + PI
            End If
        End If
    End If
End For
'End If

```

```

    'If b(i, 1) < 0 And b(i, 2) < 0 Then '第四象線
        's(i) = s(i) + PI
    'End If
Next
'排序
For i = 1 To n - 1
    For j = 1 To n - i
        If s(j) < s(j + 1) Then
            t = a(j, 1): a(j, 1) = a(j + 1, 1): a(j + 1, 1) = t
            t = a(j, 2): a(j, 2) = a(j + 1, 2): a(j + 1, 2) = t
            t = s(j): s(j) = s(j + 1): s(j + 1) = t
        End If
    Next
Next
Next
'畫線
For i = 1 To 4
    Print a(i, 1), a(i, 2), s(i)
Next
For i = 1 To n - 1
    pic.Line (a(i, 1), a(i, 2))-(a(i + 1, 1), a(i + 1, 2))
Next
pic.Line (a(1, 1), a(1, 2))-(a(n, 1), a(n, 2))
End Sub
4、VB6 沒有好用方便的極座標轉換函式 Atn2，所以必須自行撰寫如下：(C 語言有 atan2)

```



```

Private Sub Command2_Click()
    Dim x, y As Integer
    PI = 4 * Atn(1)
    Print PI
    pic.Cls
    Cls
    Dim s As Single '斜率
    設定座標系統
    pic.Scale (-10, 10)-(10, -10)
    '畫出 x 軸
    pic.Line (-10, 0)-(10, 0)
    '畫出 y 軸
    pic.Line (0, 10)-(0, -10)
    x = Val(Text1.Text)
    y = Val(Text2.Text)
    pic.PSet (x, y), QBColor(12)
    If x = 0 And y >= 0 Then
        s = PI / 2 '正 y
    Else

```

```
If x = 0 And y < 0 Then
    s = 3 * PI / 2 ' minus y
Else
    If x > 0 Then
        s = Atn(y / x) 'I,IV
    Else
        s = Atn(y / x) + PI 'II,III
    End If
End If
End If
Print s
End Sub
```

此為作者 30 年的教學心得，僅供大家選書參考，更多的精彩內容請購書，這樣作者才有資金繼續研發精彩教材，科技才會進步。