Chapter 08

迴圈指令

學習綱要

- 8-1 for 迴圈
- 8-2 while 迴圈
- 8-3 迴圈指令應用實例

學習目標

- 1. 認識 C 語言迴圈指令。
- 2. 能以 for 指令完成重複動作。
- 3. 能以 while 指令完成重複動作。
- 4. 能實作迴圈指令實例。

生活上常會遇到重複執行的事件，例如，一天要上七堂課，連續作答選擇題 10 題等等。所以電腦就有所謂的迴圈，幫您完成指定的重複次數。其次，所有的程式語言都有兩種基本迴圈，那就是 for 與 while，其用法與差別請看本章說明。

8-1 for 迴圈

我們小時候常常被老師罰寫某個句子十遍，這件事若讓電腦做，那可就輕鬆了，因爲電腦有 for 迴圈運算。例如，以下敘述可將『洪焌維』重複寫 10 次。

```
for (int i=1;i<=10;i++)
    printf("洪焌維\n");
```

for 指令的語法如下：

```
for([計數變數=起始值] ;[迴圈運算式];[計數變數的改變量])      {
    [指令區塊1;]
    [break;]
    [continue;]
    [goto (標籤名稱);]
    [指令區塊2;]
}
```

以上語法說明如下：

1. 只要“迴圈運算式”結果爲 1(眞)，則繼續執行迴圈內的指令區塊。
2. for 運算僅預設執行 1 個敘述，若要執行兩個以上敘述，請加上 {}。請鍵入以下程式，寫出執行結果。

```
int main() {
    for (int i=1;i<=10;i++){
        printf("%d\n",i);
        printf("%d\n",i);
    }
}
```

```
int main() {
  int i;
  for ( i=1;i<=10;i++)
        printf("%d\n",i);
        printf("%d\n",i);
}
```

以上變數 i 的有效範圍僅在迴圈內，所以以下程式，第 2 個 printf("%d\n",i); 的 i 無效。

```
int main() {
    for ( int i=1;i<=10;i++){
        printf("%d\n",i);
    }
    printf("%d\n",i);
}
```

3. 計數變數的改變量若爲遞增，請留意終值較大；若爲遞減，則終值較小。請鍵入以下程式，並寫出執行結果。

```
int main() {
    for (int i=1;i<=10;i++)
        printf("%d\n",i);
    for (int i=1;i<=10;i=i+4)
        printf("%d\n",i);
    return 0;
}
```

```
int main() {
    for (int i=10;i>=1;i--)
        printf("%d\n",i);
    for (int i=10;i>=1;i=i-3)
        printf("%d\n",i);
    return 0;
}
```

4. 前面計數變數都是由小而大，使用『<=』。若是由大到小，則要用『>=』，這樣第一筆資料才會得到『true』，才有輸出結果。

```
int main() {
    for( i=10;i>=1;i--) {
        printf("%d",i);//10987654321
    }
}
```

5. 以下程式每次遞減 3。

```
int main() {
    for( i=10;i>=1;i-=3) {
        printf("%d",i);//10741
    }
}
```

6. 關係運算子通常使用『<,<=,>,>=』，若使用『==,!=』，則沒有錯誤訊息，但執行結果不同。請鍵入以下程式，寫出執行結果。

```
int main(){
    for (int i=1;i==5;i++)
        printf("%d",i);
    printf("\n");
    return 0;
}
```

```
int main(){
    for (int i=1;i!=5;i++)
        printf("%d",i);
    printf("\n");
    return 0;
}
```

7. 迴圈運算式也不能寫成：

```
int main() {
    for ( int i=1;i=10;i++)
        printf("%d",i);
}
```

因爲 C 關係運算子沒有「=」，這樣是語法錯誤。其次，若寫成：

```
int main() {
    for ( i=1;i==10;i++)
        printf("%d",i);
}
```

則是語法沒錯，但結果不一樣，因爲一開始 1==10 這件事就 false，所以什麼都沒有作就離開迴圈。

8. 只要迴圈運算式爲 0(false)，就離開迴圈。請鍵入以下程式，並寫出執行結果。

```
int main(){
    for (int i=1;i>=5;i++)
        printf("%d\n",i);
    for (int i=5;i<=1;i++)
        printf("%d\n",i);
    return 0;
}
```

9. 計數變量也可在迴圈內改變。例如：

```
int main(){
    for (int i=1;i<=10 ;i++){
        printf("%d\n",i);
        i++;
    }
    return 0;
}
```

10. 程式若執行到 break，則會提早離開 for 迴圈。（請鍵入以下程式，寫出執行結果）

```
int main(){
    int i, sum=0;
    for(i=1; i<=6; i++) {
        printf("%d\n",i);
        if(i==4)
            break;  //強制離開迴圈
        printf("%d\n",i);
    }
    printf("when out of loop, i=%d",i);
    return 0;
}
```

11. 程式若執行到 continue，則會略過 continue 下面的指令區塊 2，繼續執行下一個計數變量。（請鍵入以下程式，寫出執行結果）

```
int main(){
    int i, sum=0;
    for(i=1; i<=6; i++) {
        printf("%d\n",i);
        if(i==4)
            continue;  //略過下一指令，繼續執行下一個i
        printf("%d\n",i);
    }
    printf("when out of loop, i=%d",i);
    return 0;
}
```

12. 程式若執行到 goto，則會略過 goto 下面的指令區塊 2，跳躍至 (標籤名稱) 所在的位置，繼續執行標籤以下的指令區塊。

```
int main(){
    int i, sum=0;
    for(i=1; i<=6; i++) {
        printf("%d\n",i);
        if(i==4)
            goto aa;  //強制離開迴圈
        printf("%d\n",i);
    }
    aa:
    printf("when out of loop, i=%d",i);
    return 0;
}
```

13. 指令區塊內可以放置任何合法的指令，當然也可含 for。for 內有 for，稱爲巢狀迴圈（Nested Loop）。例如，以下指令可印出 1 至 10 五次。

```
int main(){
    for (int i=1;i<=5;i++){
        for (int j=1;j<=10;j++)
            printf("%d",j);
        printf("\n");
    }
    return 0;
}
```

14. 運算式只要合理就好，不一定要用計數變數。例如：

```
int main(){
    int i=0,s=0;
    for (i=1;s<=10;i++){
        printf("%d",i);
        s=s+i;
    }
    printf("%d",i);
    printf("%d",s);
    return 0;
}
```

```
int main(){
    int i=0,s=0;
    for (i=1;i<=10 && s<=10;i++)
    {
        printf("%d",i);
        s=s+i;
    }
    printf("%d",i);
    printf("%d",s);
    return 0;
}
```

15. 計數變量若用浮點數，因爲浮點數會有誤差，請鍵入以下程式，並寫出執行結果。

```
int main(){
    for (double i=1;i<=2;i=i+0.1){
            printf("%lf",i);
    }
    return 0;
}
```

範例8-1a

認識鍵盤遊戲。請寫一個程式，電腦可以連續出現 10 個字元，每出現一個字元，都由使用者輸入，並判斷是否正確，統計正確與錯誤題數。

程式列印

前面我們已經完成產生 1 個字元，使用者鍵入此字元，電腦判斷是否正確，現在要連續出現 10 個字元，只要加上 1 個 for 迴圈就可以。

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>//本例使用time()函式
int main()
{
    srand(time(NULL));
    int r=0,w=0;
    for (int i=1;i<=10;i++){
        int a=97+rand()%26;
        printf("%c",(char)a);
        char b;
        scanf(" %c",&b);//有點bug,線上查詢後，得知要在控制字元「%」前加空白
        if (b==(char)a){
            printf("right\n");
            r++;
        }
        else{
            printf("wrong\n");
            w++;
        }
```

```
    }
    printf("right=%d\n",r);
    printf("wrong=%d\n",w);
    return 0;
}
```

自我練習

1. 請產生 6 個 -3~3 的亂數，統計正數、0、負數的個數。
2. 請產生 10 個 1~6 的亂數，統計偶數與奇數個數。
3. 請產生 10 組 (x,y) 座標亂數，(x,y) 都介於 -2 與 2 的整數，並統計落在原點、x 軸、y 軸、四個象限的個數。
4. 請產生 10 個 1~6 的亂數，統計數字「3」出現的次數。
5. 請產生 10 個 1~6 的亂數，統計所有數字出現的次數。
6. 請產生 10 個 1~6 的亂數，統計哪一個數字出現的次數最多。
7. 心算練習。請連續產生 8 題兩個 1 位數，由使用者輸入相加結果，電腦回應對或錯，並統計正確與錯誤的題數。
8. 音感練習。請連續發出 10 個 Do 到高音 Do，由使用者輸入 1~8，電腦回應對或錯，並統計正確與錯誤的題數。
9. 請由電腦自動產生 30 個大於等於 0 且小於等於 100 的亂數 x，統計 0~59,60~69,70~79,80~89,90~100 的個數。
10. 請寫一個程式，電腦可以連續出現 50 個小寫字元，電腦統計出現母音的個數。

範例8-1b

試寫一程式，可以輸入 2 至 9 的整數，並輸出此數的九九乘法表。

執行結果

```
input a integer between 2 to 9: 6
6 * 2= 12
6 * 3= 18
6 * 4= 24
6 * 5= 30
6 * 6= 36
6 * 7= 42
6 * 8= 48
6 * 9= 54
```

程式列印

每個數字都要使用 2~9 去乘，這種重複的事件，即爲 for 迴圈的應用，參考程式如下：

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int a;
    printf("input a integer between 2 to 9: ");
    scanf("%d",&a);//&
    for(int i=2; i<=9; i++)
        printf("%d * %d= %d\n",a,i,a*i);
    return 0;
}
```

範例8-1c

求解輸入數值是否質數。

設計步驟

1. 寫出演算法。

   任一整數，除了 1 和本身外，若沒有任何數可以整除此數，則稱此數爲質數。所以，本例使用 2 至該數減一的數一一試除，若無一數可整除，則稱此數爲質數。其次，爲了提高程式執行效率，若找到 1 個可以整除，那此數就不是質數，可以提早離開迴圈。

程式列印

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int a=11;
    _Bool prime=1;//布林型態
    for(int i=2;i<=a-1;i++){
        if (a%i==0) {// 留意==,不是=
            prime=0;
            break;//找到1個就不是了
        }
```

```
    }
    if(prime)
        printf("%d is a prime",a)      ;//1
    else
        printf("%d is not a prime",a);//0
    return 0;
}
```

自我練習

1. 請寫一程式，可以輸入兩數，並判斷其是否互質。(兩數除了 1 以外，沒有任何公因數，稱爲互質。)

8-2 while 迴圈

上一節的 for 是用於程式設計階段已知迴圈次數。但有些情況，我們於程式設計階段並不知迴圈的執行次數，此時即可使用 while 指令。且有些迴圈可能一次都不執行，所以 while 指令又分爲前測試迴圈（Pre-test loop）與後測試迴圈（Post-test loop）。while 的前測試迴圈語法如下圖左，後測試迴圈如下圖右。

```
while(運算式){
    指令區塊;
    break;
    continue;
}
```

```
do{
   指令區塊;
   break;
   continue;
}while (運算式);
```

以上語法說明如下：

1. 不論是前測試迴圈或後測試迴圈，均是運算式值爲 1（眞）時，繼續執行迴圈，運算式爲 0(僞) 時，離開迴圈。
2. 前測試與後測試迴圈的差別爲，前測試迴圈有可能一次均不執行迴圈，但後測試迴圈至少執行一次。
3. 後測試迴圈的 while（運算式）後面要加分號 (;)，而前測試迴圈的 while 不用加分號。
4. while 指令區塊內亦適用 break 與 continue，前者爲強迫提早離開迴圈，後者可略過部份指令，提早進入條件運算式。

5. 以下程式片段可用前測試迴圈解輸出 1 ～ 10。

```
int main(){
    int i=1;
    while ( i<=10){
        printf("%d\n",i);
        i=i+1;
    }
    return 0;
}
```

以下程式片段可用後測試迴圈輸出 1 ～ 10。

```
int main(){
    int i=1;
    do{
        printf("%d\n",i);
        i=i+1;
    } while (i<=10);
    return 0;
}
```

範例8-2a

編譯器的除法運算子。請使用加減法，完成除法運算。

演算法則

兩數相乘時，程式設計階段就知道迴圈執行次數，所以使用 for。但除法就是不知道，只能說，當被除數大於除數時，就連續減去除數，能減去的次數，就是商，剩下的就是餘數。

1. a= 被除數。
2. b= 除數。
3. 商數 q=0。
4. 所謂的商就是被除數 a 共有幾個除數 b，也就是只要 a 大於等於 b，就要執行以下指令：

   a=a-b；

   q=q+1；

5. 本例以 8 除以 3，實際演練如下：
   (1) a=8；b=3
   a>=b 成立，所以執行迴圈指令
   a=a-b=5
   q=q+1=1（商）
   (2) a=5；b=3
   a>=b 成立，所以執行迴圈指令
   a=a-b=2
   q=q+1=2（商）
   (3) a=2；b=3
   a>=b 不成立，所以離開迴圈
   q=2（商）
   a=2（餘數）

程式列印

1. 本例必須重複很多次，所以適用迴圈，程式才能精簡，迴圈有 for 與 while，本例設計階段不知重複幾次，而是一面減、一面判斷，所以適用 while 迴圈。
2. 本程式僅能使用前測試 while 迴圈，而不能使用後測試迴圈，因爲有可能一開始被除數就小於除數。

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int a=8,b=3,q=0;
    //printf("input a dividend: ");          // 輸入被除數
    //scanf("%d",&a);//&
    //printf("\ninput a divisor: ");         //  輸入除數
    //scanf("%d",&b);
    while(a>=b) {
        a=a-b;
        q++;
    }
    printf("quotient = %d\n", q);            // 商數
    printf("remainder= %d\n", a);            // 餘數
    return 0;
}
```

自我練習

1. 同範例，但可以求到小數點 1 位的實數除法。提示：將被除數先乘以 10，再運算，最後將商再除以 10。

範例8-2b

若有一級數 s=3+6+9+…，請問加到第 n 項，其和剛好超過 100。

程式列印

1. 本例於設計階段並不知道迴圈執行次數，所以不適用 for。其次，題目給的條件是累加超過 100，所以可以使用 while (sum <100) 則繼續執行迴圈，繼續累加。

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int i=3,sum=0,n=0;
    while(sum<100){
        printf("%d\n",i);
        sum=sum+i;
        printf("sum=%d\n",sum);
        n=n+1;//次數
        i=i+3;
    }
    printf("\n");
    printf("%d\n",i-3);
    printf("%d\n",n);
    return 0;
}
```

2. 本例若使用 for，則程式如下：

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int i=3,sum=0,n=0;
    for (i=3;sum<100;i=i+3){
        printf("%d\n",i);
```

```
        sum=sum+i;
        printf("sum=%d\n",sum);
        n=n+1;//次數
    }
    printf("\n");
    printf("%d\n",i-3);
    printf("%d\n",n);
    return 0;
}
```

自我練習

1. 請問 3+3+3…共加給次，其和剛好超過 100。
2. 請寫一程式，可以輸入任意個數的整數，當輸入 -1 時結束，計算輸入數字的平均。
3. 請寫一程式，可以輸入一個整數，並將其從個數數逐一輸出，且計算其數字和。例如，輸入 25，輸出 5 2 7。
4. 心算練習。請連續產生兩個 1 位數，由使用者輸入相加結果，電腦回應對或錯，直到錯了才停止，並統計正確題數。
5. 音感練習。請連續發出 10 個 Do 到高音 Do，由使用者輸入 1~8，電腦回應對或錯，直到錯了才停止，並統計正確題數。
6. 密碼。假設密碼爲「1234」的整數，每次程式執行時，至多可以錯 2 次，才可以使用以上範例，密碼錯誤 3 次就離開此程式。
7. 請判斷所輸入任意數是否全偶數。例如，2682 爲全偶數，但 26814 就不是。
8. 請判斷所輸入任意數是否全遞增。例如，1268 爲全遞增，但 12681 就不是。
9. 請寫一個程式，可以連續輸入阿拉伯數字，每次 1 個，當輸入「e」時結束，且組合成一個數字。例如：輸入 4,3,8,e 則輸出 438。

8-3　巢狀迴圈

迴圈中又有迴圈，稱爲巢狀迴圈。巢狀迴圈在程式設計的領域非常重要，這種運算思維是電腦的強項，也是初學者最感頭疼的單元，本節將使用若干範例引領學生征服此運算思維。

範例8-3a

請寫一程式，印出如下的九九乘法表。

執行結果

```
1 * 1 =  1 1 * 2 =  2 1 * 3 =  3 1 * 4 =  4 1 * 5 =  5 1 * 6 =  6 1 * 7 =  7 1 * 8 =  8 1 * 9 =  9
2 * 1 =  2 2 * 2 =  4 2 * 3 =  6 2 * 4 =  8 2 * 5 = 10 2 * 6 = 12 2 * 7 = 14 2 * 8 = 16 2 * 9 = 18
3 * 1 =  3 3 * 2 =  6 3 * 3 =  9 3 * 4 = 12 3 * 5 = 15 3 * 6 = 18 3 * 7 = 21 3 * 8 = 24 3 * 9 = 27
4 * 1 =  4 4 * 2 =  8 4 * 3 = 12 4 * 4 = 16 4 * 5 = 20 4 * 6 = 24 4 * 7 = 28 4 * 8 = 32 4 * 9 = 36
5 * 1 =  5 5 * 2 = 10 5 * 3 = 15 5 * 4 = 20 5 * 5 = 25 5 * 6 = 30 5 * 7 = 35 5 * 8 = 40 5 * 9 = 45
6 * 1 =  6 6 * 2 = 12 6 * 3 = 18 6 * 4 = 24 6 * 5 = 30 6 * 6 = 36 6 * 7 = 42 6 * 8 = 48 6 * 9 = 54
7 * 1 =  7 7 * 2 = 14 7 * 3 = 21 7 * 4 = 28 7 * 5 = 35 7 * 6 = 42 7 * 7 = 49 7 * 8 = 56 7 * 9 = 63
8 * 1 =  8 8 * 2 = 16 8 * 3 = 24 8 * 4 = 32 8 * 5 = 40 8 * 6 = 48 8 * 7 = 56 8 * 8 = 64 8 * 9 = 72
9 * 1 =  9 9 * 2 = 18 9 * 3 = 27 9 * 4 = 36 9 * 5 = 45 9 * 6 = 54 9 * 7 = 63 9 * 8 = 72 9 * 9 = 81
```

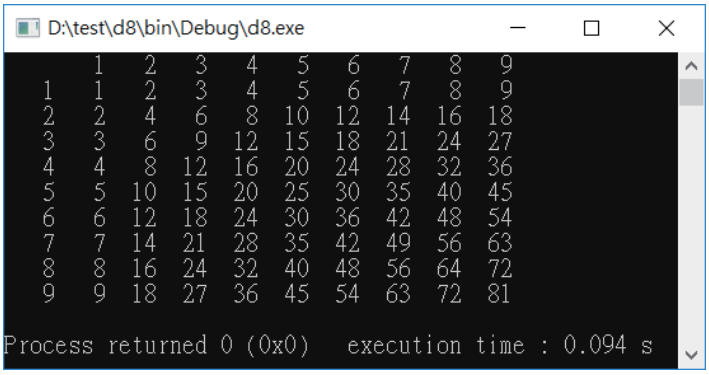
程式列印

前面單元已經介紹如何輸出 1 個數字的九九乘法表，現在只是此一數字也要以迴圈表示，所以雙迴圈程式如下：

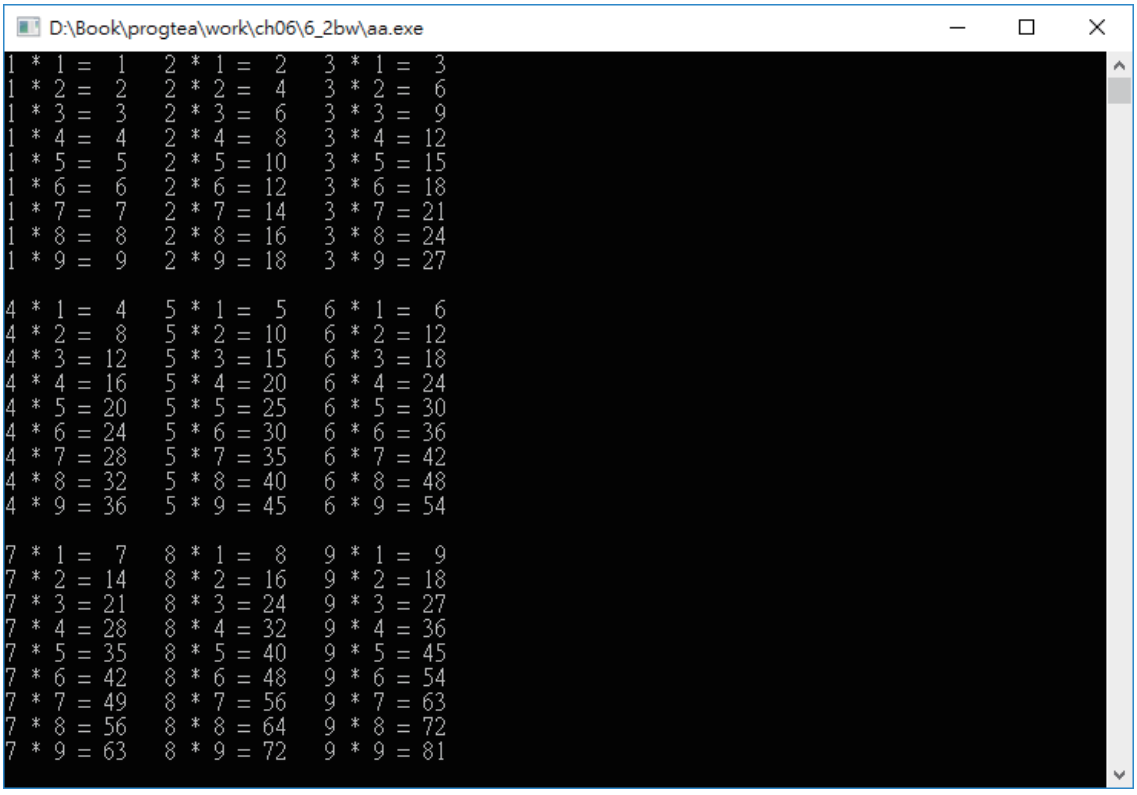
```
#include <stdio.h>
#include <stdlib.h>
int main() {
    for (int i=1;i<=9;i++){
        for (int j=1;j<=9;j++){
            printf("%d * %d = %2d ",i,j,i*j);
        }
        printf("\n");
    }
    return 0;
}
```

自我練習

1. 請寫一程式，嘗試使用兩層迴圈印出如下的九九乘法表。



2. 請寫一程式，嘗試使用三層迴圈印出如下的九九乘法表。



範例8-3b

請寫一程式，印出 2 至 1000 的所有質數。

運算思維

前面我們已經說明如何判別一個整數是否質數，現在要找 2 至 1000 的所有質數，所以只要再使用一個迴圈就好，參考程式如下：

程式列印

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    _Bool prime=1;//布林型態 ，有些編譯器不行
    for(int a=2 ;a<=1000;a++){
        prime=1;//每一次都要預設初值
        for(int i=2;i<=a-1;i++){
            if (a%i==0) {// 留意==,不是=
                prime=0;
                break;//找到1個就不是了
            }
        }
        if(prime)
            printf("%5d",a);//每個數字佔5格，靠右對齊
    }
    return 0;
}
```

自我練習

1. 同上範例，但請加上讓每列僅輸出 10 個數字。
2. 請寫一程式，可以計算 1+(1+2)+(1+2+3)+(1+2+3+4)+..+(1+2+..+n)，n 可指派爲 1 到 100 的自然數。

※ 3. 質因數連乘積。請寫一個程式，輸入一正整數，將其質因數分解後印出其式子，例如：

```
輸入：319
輸出：319=11*29
輸入：19
```

```
輸出：19 =質數
輸入：521752
輸出：521752 = 2^3*7^2*11^3
```

範例8-3c

試寫一程式，找出三位數“阿姆斯壯數”。所謂阿姆斯壯數是指一數等於各個位數的立方和。例如，$153=1^3+5^3+3^3$。

程式列印

1. 本例可以使用三個迴圈，程式如下：

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>//使用pow()
int main() {
    int i, j, k;
    int n=0;              /* 個數 */
    int sum1, sum2;
    for(i=1; i<=9; i++){
        for(j=0; j<=9; j++){
            for(k=0; k<=9; k++) {
                sum1=100*i+10*j+k;
                sum2=(int)(pow(i,3)+pow(j,3)+pow(k,3));
                //pow傳回double，所以要轉型
                if(sum1==sum2){
                    n++;
                    printf("%d : %4d\n",n,sum1);
                }
            }
        }
    }
    return 0;
}
```

2. 也可以使用 1 個迴圈，然後分解數字，程式如下：

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main() {
```

```
    for (int i=100;i<=999;i++){
        int a=i/100;
        int b=(i-a*100)/10;
        int c=i%10;
        int d=pow(a,3)+pow(b,3)+pow(c,3);
        if (i==d )
            printf("%d\n",i);
    }
    return 0;
}
```

自我練習

1. 試寫一程式，找出四位數“阿姆斯壯數”abcd=a^4+b^4+c^4+d^4。提示：同範例，但使用四層迴圈。

範例8-3d

請嘗試使用雙迴圈，寫一程式輸出如下：

```
*
**
***
****
*****
******
```

程式列印

1. 本例的列數、每一列的星星個數，列表解析如下：

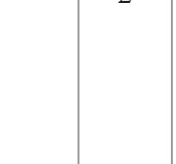
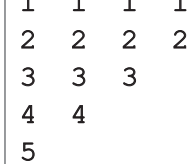
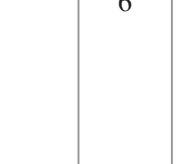
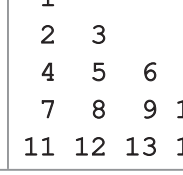
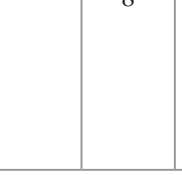
列編號	星星個數
1	1
2	2
3	3
4	4
5	5
6	6

2. 由上表可知，共 6 列，所以 i 從 1 到 6。

3. 共有兩個變數，我們就是要找關係，這樣才能簡化問題，找出解答，本例內迴圈 j 與 i 的關係爲 i=j ，所以 j 從 1 到 i。（此一變數找關係的運算思維，往後會持續用到）

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int i, j;
    for(i=1; i<=6; i++) {
        for(j=1; j<=i; j++)   {
            printf("*");
        }
        printf("\n");//跳列且歸位
    }
    return 0;
}
```

自我練習

題號	題目	題號	題目
1		2	
3	 （105APCS試題）	4	 （105APCS試題）
5		6	
7		8	

8-4　迴圈應用實例

循序法

循序法故名思義就是將所有的值一一列出或一一嘗試，程式語言裡的 for 即可實現此演算法。請看以下範例。

範例8-4a

電腦的乘法運算。

運算思維

1. 人類計算乘法大致如下：

$$\begin{array}{rcccc} & & & 8 & 2 \\ \times & & & 3 & 6 \\ \hline & & 4 & 9 & 2 \\ & 2 & 4 & 6 & \\ \hline & 2 & 9 & 5 & 2 \end{array}$$

2. 但是電腦可不用如此麻煩，不用任何記憶，不用任何學習，電腦的強項是計算能力強，所以就用 for 迴圈實現累加如下：（電腦內部只有累加器與比較器）

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int a=82,b=36,sum=0;
    for(int i=1; i<=b; i++)
        sum=sum+a;//累加a
    printf("sum=%d",sum);
    return 0;
}
```

範例 8-4b

等差數列的求和。例如：寫一個程式計算 1+2+3+…+100 的和，或計算 3+6+9+…+93 的和。

運算思維

本例人類通常使用等差級數求和的公式（首項 + 末項）* 項數 /2，但電腦可不用如此麻煩、不用額外學習，電腦就直接一個一個逐一累加，程式如下：

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int s=0;
    for (int i=1;i<=100;i++) {
        printf("%d\n",i);
        s=s+i;//逐一將i累加
    }
    printf("\n");
    printf("%d",s);
    return 0;
}
```

自我練習

1. 請寫一個程式計算 5+11+17+…53 的和。
2. 請寫一個程式計算 1+1/2+1/3+…+1/20 的和。

循序搜尋法（Sequential search）

所謂循序搜尋法，就是將所有可能的解一一循序代入，又稱爲暴力猜值法。例如，您要求任一數的開平方，因爲開平方的結果一定在 0 與此數之間，所以我們就從 0 開始，每次遞增 1、0.1、0.01 或 0.001…，至於是要遞增多少，那就依您要的精密度了。例如，要求整數解，那就遞增 1，要求到小數 1 位，那就遞增 0.1，要求到小數兩位，就遞增 0.01 等等等，請看以下範例說明。

範例8-4c

請用循序猜值法求任意數的平方根。

運算思維

1. 人類求開根號的方法，大致如下：

		3　　7　　2
1	9*9=81(>13 不行) 8*8=64(>13 不行) 7*7=47(>13 不行) 6*6=36(>13 不行) 5*5=25(>13 不行) 4*4=16(>13 不行) 3*3=9(沒有 >13，可以)	1　3　8　3　8　4 　9
2	9,8 太大 3×20+7=67 67×7=469	4　8　3 4　6　9
3	9,8,7,6,5,4,3 太大 37×20+2=742 742×2=1484	1　4　8　4 1　4　8　4
4		0

以上演算法則在第 9 章習題 4(p9-63) 有詳細說明，可以減少計算，但是電腦計算能力強，所以可使用循序搜尋法或二分搜尋法求解，此即爲電腦的運算思維。（二分搜尋法請看範例 8-4e）

程式列印

1. 以下程式每次遞增 1。

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int i, y;
    int a=9;
    for ( i=0;i<=a;i=i+1){
        printf("%d\n",i);
```

```
        y=i*i;
        if (y>=a){
            break;
        }
    }
    printf("%d",i);
    return 0;
}
```

2. 以下程式每次遞增 0.1。

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    double i, y;
    double a=10;
    double d=0.1;
    for ( i=0;i<=a;i=i+d){
        printf("%lf\n",i);
        y=i*i;
        if (y>=a){
            break;
        }
    }
    printf("%lf",i);
    return 0;
}
```

自我練習

1. 請用循序法求解某一正數的立方根。例如，輸入 27 可得到 3.0。本例小數點取 1 位。
2. 假設有一函式 $y=f(x)=x^2-4x-5$ 請分別印出 x 從 -10 到 10 的值。
3. 同上題，請用循序猜值找出其整數解。提示：x 從 -10 到 10 一一代入，找出使函數爲零的值，此即爲暴力猜值法解題。
4. 同上題，請找出極小值。

5. 函數極值。請寫一程式，可以輸入一個一元二次函式，並求其極大或極小值。例如，輸入 $y=f(x)=x^2-2x+2$ 有極小值 1，輸入 $y=f(x)=-x^2-2x+2$ 有極大值 -1，
6. 假設一個一元多次方程式含有實數解，請寫一程式，可求其解。例如，$y=f(x)= x^2+x-0.75=0$ 的解是 0.5 和 -1.5。提示：浮點運算時，若無法得到 0，此時要使用接近 0 的判斷。例如，abs(y)<0.0001，即可視爲成立，0.0001 即是其精密度，請換成自己想要的精密度 0.1、0.01 或 0.001。
7. 請用循序法求解任意整數的所有因數。
8. 請寫一個程式，找出 1 至 200 的整數中，找出含有 2 的數字，且統計共含有多少個 2。例如，122 有兩個 2。

輾轉相除法

以輾轉相除法，求兩數的最大公因數的演算法如下：

1. 輸入 a、b 兩數。
2. r= 餘數。
3. 將 a 除以 b，得餘數 r，假如餘數不爲 0，則以原除數爲被除數，原餘數爲除數，重複執行 a 除以 b，直到餘數爲 0，促使餘數爲 0 的除數，即爲最大公因數。
4. 本例以 a=21，b=9，實際演練如下：

   (1) a=21；b=9；

   r=a%b=3

   a=b=9

   b=r=3

   r>0 成立，所以繼續執行迴圈。

   (2) a=9；b=3；

   r=a%b=0

   a=b=3

   b=r=0

   r>0 不成立，所以離開迴圈，最大公因數爲 a=3。

範例8-4d

以輾轉相除法，求兩數的最大公因數。

程式列印

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int a=9,b=21,t,r;
    do {
        r=a%b;          // 餘數
        a=b;            // 以原除數爲被除數
        b=r;            // 以原餘數爲除數
    }
    while (r>0);
    //離開迴圈時，最大公因數已經轉交給a
    printf(" The highest common factor:%d",a);  // 輸出最大公因數
    return 0;
}
```

程式說明

1. 輾轉相除法的執行之初並無法預估執行次數，所以較適用 while，且本例迴圈至少執行一次，所以用後測試迴圈。
2. 後測試迴圈的 while 之後必須加分號 (;)，以免產生錯誤的訊息。
3. 此題也可寫成前測試迴圈，程式如下，此爲 APCS105 試題。

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int a=76,b=48,r;
    while ((a% b) !=0){
        r=a%b;
        a=b;
        b=r;
    }
    printf("%d",b);
    return 0;
}
```

4. 因爲是前測試，所以離開迴圈時，當時的 b，就是餘數，就是最大因數。

自我練習

1. 請寫一個程式，滿足以下條件：
   (1) 可以產生 0 至 5 的亂數。
   (2) 累加以上亂數。
   (3) 輸出此亂數與統計其和。
   (4) 若亂數不爲 0，則重複（1）～（3），否則輸出其和並結束程式。
2. 請寫一個程式，電腦一直產生 1 到 6 的亂數，直到有連續兩個相等爲止，並輸出其和。例如，1,3,4,4，答案是 12。
3. 請寫一個程式，電腦一直產生 1 到 6 的亂數，直到有連續三個亂數相同爲止，並輸出其和。例如 :1,5,3,2,2,2，答案是 9; 又例如 :4,5,2,2,3,3,3 答案是 13。
4. 請寫一個程式，滿足以下條件：(擲骰子遊戲)
   (1) 可以產生兩個 1 至 6 的亂數。（提示：1 ＋ rand()%6）
   (2) 累加以上亂數。
   (3) 輸出此亂數與其和。
   (4) 若亂數和大於 8，則重複（1）～（3），直到亂數和小於等於 8，則程式結束。
5. 請寫一程式，可以連續產生兩個 1 ～ 6 的亂數，並輸出此兩個亂數，直到後面的亂數大於前面的亂數。
6. 請寫一程式，可以連續產生 3 個 1 ～ 6 的亂數，並輸出此 3 個亂數，直到有其中兩個亂數相等爲止。
7. 請寫一程式，可以連續產生 4 個 1 ～ 6 的亂數，並輸出此 4 個亂數，直到有其中兩個亂數相等爲止，並輸出此不相等的數字與和。例如，產生 6,4,5,1 則繼續產生亂數，若亂數爲 6,2,1,6 則其和爲 3。

※ 8. 請寫一猜拳遊戲程式，可以讓人與電腦猜拳，並輸出結果，直到任一方贏 3 次爲止。

二分搜尋法

前面的循序法是循序一個一個搜尋，若沒搜尋中，每次僅減少一筆資料，這裡要介紹一個較有效率的搜尋法，稱爲二分搜尋法。二分搜尋法是每次搜尋其可能範圍的中間值，若搜尋的太大，則調整搜尋上限爲此中間值，表示後半部都刪除；若搜尋的太小，則調整搜尋下限爲此中間值，表示前半部都刪除，每次都縮小範圍爲一半，所以可提升搜尋效率，請看一下範例。

範例8-4e

猜數字。請您先默想一個 1 ～ 100 的數字，讓電腦猜，但每次要回應太大 '3' 或猜中 '2' 或太小 '1'。

執行結果

以下是我默想 40，逐次回答電腦，電腦搜尋的過程。

```
 調整 x1=1,x2=100

I guess 50,please enter 1(太小),2(正確) or 3(太大):3
a=3 ,太大 調整 x1=1,x2=49

I guess 25,please enter 1(太小),2(正確) or 3(太大):1
a=1 ,太小 調整 x1=26,x2=49

I guess 37,please enter 1(太小),2(正確) or 3(太大):1
a=1 ,太小 調整 x1=38,x2=49

I guess 43,please enter 1(太小),2(正確) or 3(太大):3
a=3 ,太大 調整 x1=38,x2=42

I guess 40,please enter 1(太小),2(正確) or 3(太大):
```

演算法

1. 讓電腦搜尋，電腦可從 1,2,3 開時搜尋，此稱爲循序搜尋法，若數字是 1，那運氣好，1 次就搜尋到；若數字是 100，那就要搜尋 100 次，所以平均是 50 次才可搜尋到。這就由讀者練習。
2. 本範例要介紹二分搜尋法，那就是每次搜尋其一半。例如，本例下限 x1=1，上限 x2=100，每次就搜尋一半 x=(x1+x2)/2=50，然後您會告知搜尋中、太大或太小。

3. 若是太大，調整上限 x2=x-1=49; 若是太小，則調整下限 x1=x+1=51，然後重複步驟 2 與 3。每次範圍縮小一半，很快就會搜尋到，程式如下：

程式列印

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    int x1=1,x2=100,x;
    int a;
    _Bool right=0;//舊版編譯器不行，請改爲int right=0;
    do{
        printf(" 調整 x1=%d,x2=%d\n",x1,x2);
        x=(x1+x2)/2;
        printf("\nI guess %d,please enter 1(太小),2(正確) or 3(太大):",x);
        scanf("%d",&a);
        printf("a=%d ,",a);
        if (a==2){
            right=1;
            break;
        }
        else {
            if (a==3){
                printf("太大");
                x2=x-1;//調整上限爲x-1
            }
            else{//a=1
                printf("太小");
                x1=x+1;//調整下限
            }
        }
    }while( right==0);
    printf(" Bingo, The number is %d",x);
    return 0;
}
```

範例8-4f

請以二分搜法求解一正數的平分根。本例要精密度到小數點以下第二位。

執行結果

```
4.500000
2.250000
3.375000
2.812500
3.093750
2.953125
3.023438
2.988281
3.005859
2.997070
The sqrt of 9.000000 is 2.997070
```

演算法

1. 本例以求 9 的平分根爲例。
2. 求解 9 的平方根，其解必在 0 到 9 之間，所以設定下界爲 0，上界爲 9。首先，先搜尋 4.5，如下圖步驟 1，但 4.5 的平方爲 20.25，大於 9，表示搜尋的太大，那就縮小範圍，將上界調爲 4.5，持續在 0 與 4.5 之間搜尋值，如下圖步驟 2。第二次就搜尋 2.25，但是 2.25 的平方爲 5.025，小於 9，表示搜尋的太小，那就調整下界爲 2.25，且持續在 2.25 與 4.5 之間搜尋，如下圖步驟 3，那要搜尋到何時呢？答案是設定一個精密度，例如，您要小數一位，那就是下界與上界之間的距離小於 0.1，若是要小數兩位，那就是下界與上界之間的距離小於 0.01，此時下界或上界的值，就是所要求的答案了。

搜尋步驟	搜尋值內容
1	x1 x x2 0 4.5（太大） 9
2	x1 x x2 0 2.25（太小） 4.5
3	x1 x x2 3.374（太大）

3. 以上敘述的演算法如下：
   (1) 設求解的正數爲 y，則其平方根必在 x1=0（下界）與 x2=y（上界）之間，搜尋範圍爲 [x1,x2]。
   (2) 首先猜 x1+x2 之和的一半 x。
   (3) 若所搜尋 x 的平方小於 y，表示搜尋的太小，並縮小搜尋範圍爲〔x，x2〕。
   (4) 若所搜尋的 x 的平方大於 y，表示搜尋的太大，並縮小搜尋範圍爲〔x1，x〕。
   (5) 重覆步驟 (2)、(3)、(4)，直到〔x1，x2〕的範圍小於所要求的精密度（小數兩位 0.01 或小數三位 0.001），此時的 x1 或 x2 即爲平方根。

程式列印

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>//使用fabs()
int main() {
    double y,x,x1,x2,t;
    y=9;x1=0;x2=y;
    do {
        x=(x1+x2)/2;
        t=x*x;
        if (t<y)
            x1=x;//搜尋的太小 ，調整下限爲x
        else
            x2=x; //搜尋的太大 ，調整上限爲x
    }
    while (fabs(x2-x1) >0.01)    ; //只要x2-x1大於0.01就要繼續搜尋
    //fabs()絕對值
    printf("The sqrt of %f is %f ",y,x);
    return 0;
}
```

自我練習

1. 請將 fabs() 改爲 abs()，並觀察執行結果。
2. 請以二分搜尋法，求解兩數相除的結果。(本例假設除數大於 1 的整數)

3. 請以二分搜尋法，求解一正數的立方根。
4. 請寫一程式，由電腦產生一個 1 到 100 的亂數，由使用者用二分搜尋法猜，電腦應逐次回答太大、太小、或猜中，且回應幾次猜中。

※ 5. 請將 10 進位數轉爲 N 進位數（本例暫討論 N<=9，其餘 N>=11 時，待串列介紹之後再討論）。提示：演算法如下：

(1) 若要將十進位的 a 轉爲 2 進位，則以數學式表示如下：

$(a)_{10}=a_0+a_1 2^1+a_2 2^2+a_3 2^3+a_4 2^4$

$=a_0+2(a^1+a_2 2^1+a_3 2^2+a_4 2^3)$　　#a1 以後，提出 2

(2) 上式的 a0 爲 a 除以 2 的餘數，$a_1+a_2 2^1+a_3 2^2+a_4 2^3\cdots\cdots$則爲 a 除以 2 的整數商，重複上式，直至整數商等於零爲止。

(3) 最先出爐的餘數應放在 2 進位的最右邊。$(a)_{10}=(a_n\cdots a_3 a_2 a_1 a_0)_2$

課後習題

一、填充題

（請先追蹤程式寫出執行結果，然後鍵入以下程式，核對自己答案。#include <stdio.h> #include <stdlib.h> 請自己補上）

題號	題目	輸出結果

1	`int i, total = 0;`
	`for( i = 1; i < 8; i += 2 )`
	`    total += i;`
	`printf("%d",total);`	

2	`int y, r, a = 30, b = 42;`
	` r = a%b;`
	` while( r != 0 ) {`
	`    a = b;`
	`    b = r;`
	`    r = a%b;`
	` }`
	` y = b;`
	`printf("%d",y);`	

3	int main()
	{
	　　int k=4;
	　　int m=1;
	　　for (int i=1;i<=5;i++){
	　　　　//cout<<k<<" "<<m;
	　　　　for (int j=1;j<=k;j++){
	　　　　　　printf(" ");
	　　　　}
	　　　　for (int j=1;j<=m;j++){
	　　　　　　printf("*");
	　　　　}
	　　　　printf("\n");
	　　　　k=k-1;
	　　　　//m=m+1;
	　　　　m=2*i+1;//修改處
	　　}
	　　return 0;
	}	

4	void main () {//APCS
	　　int a=0;
	　　int b=1;
	　　int i, temp, N=12;
	　　for (i=2; i<=N; i=i+1) {
	　　　　temp = b;
	　　　　b=a+b ;
	　　　　a = temp;
	　　　　printf ("%d\n", (b));
	　　}
	}	

5	void main () {//APCS
	　　int i = 76;
	　　int j = 48,k;
	　　while ((i % j) != 0) {
	　　　　k=i%j;
	　　　　i=j;
	　　　　j=k;
	　　}
	　　printf ("%d\n", j);
	}	

6	void main() {//APCS
	for (int i=0; i<=10; i=i+1) {
	printf ("%d ", i);
	i = i + 1;
	}
	printf ("\n");
	}	

7	void main() {//APCS
	int a=0, n=1;
	for (int i=1; i<=n; i=i+1)
	for (int j=i; j<=n; j=j+1)
	for (int k=1; k<=n; k=k+1)
	a = a + 1;
	printf("%d",a);
	}//請分別寫出n=1~6,10,100的輸出值。	

8	void main() {//APCS105/10
	for (int i=0; i<=3; i=i+1) {
	for (int j=0; j<i; j=j+1)
	printf(" ");
	for (int k=6-2*i;k>0 ; k=k-1)
	printf("*");
	printf("\n");
	}
	}	

9	void main() {//APCS10510
	int i=2, x=3;
	int N=65536;
	while (i <= N) {
	i = i * i * i;
	x = x + 1;
	}
	printf ("%d %d \n", i, x);
	}	

10	以下程式片段無法正確列印 20 次的 "Hi!"，請問下 列哪一個修正方式仍無法正確列印 20 次的 "Hi!" ？ // APCS10603
	`for (int i=0; i<=100; i=i+5) {`
	`  printf ("%s\n", "Hi!"); }`
	(A) 需要將 i<=100 和 i=i+5 分別修正為 i<20 和 i=i+1
	(B) 需要將 i=0 修正為 i=5
	(C) 需要將 i<=100 修正為 i<100;
	(D) 需要將 i=0 和 i<=100 分別修正為 i=5 和 i<100	

11	`int main() {//APCS10603`
	`  int x = 0, n = 5;`
	`  for (int i=1; i<=n; i=i+1)`
	`  for (int j=1; j<=n; j=j+1) {`
	`  if ((i+j)==2)`
	`  x = x + 2;`
	`  if ((i+j)==3)`
	`  x = x + 3;`
	`  if ((i+j)==4)`
	`  x = x + 4;`
	`  }`
	`  printf ("%d\n", x);`
	`  return 0;`
	`}`	

二、程式設計

1. 秘密差（106/10 APCS 試題）

   將一個十進位正整數的奇數位數的和稱為 A，偶數位數的和稱為 B，則 A 與 B 的絕對差值 |A − B| 稱為這個正整數的秘密差。例如：263541 的奇數位數的和 A = 6+5+1 = 12，偶數位數的和 B= 2+3+4 = 9，所以 263541 的秘密差是 |12 − 9|= 3。給定一個十進位正整數 X，請找出 X 的秘密差。

2. 完全奇數（107/10 APCS 試題）

   13311,13199 稱為完全奇數，13021 就不是。請寫一程式，可以輸入一個整數 N，並求出比此數大的完全奇數 M 與比此數小的完全奇數 P，並求其差的最小值，也就是 | N-M | 與 | N-P | 的最小值。例如，輸入 N=13256，則 M=13311，P=13199，而 | 13256-13311 | =55 , | 13256-13199 | =57，所以輸出 55。

※ 3. 遞增數。APCS 107 試題

345 等稱為遞增數，53、55 等就不是，請寫一個程式，可以求出 1 到指定輸入數字的遞增數。（APCS107 試題）

※ 4. 請用迴圈解 $\int_0^{10} xdx$ ，dx 分別取 1、0.1 與 0.01。

※ 5. 請使用迴圈求解 sin(x) 正半週面積。sin(x) 正半週面積 = $\int_0^{\pi} \sin x dx = 2$ 。（sinx 請先查詢 11-1 公用函式，此為高職基本電學一年級內容）

6. e= 2.718 的由來

工程與科學的指數與對數計算，通常不是以 2 或 10 為底，而是以 e 為底，我簡單闡述如下：

(1) 假設借款金額為一元，年利率為 100%，每年複利一次，則一年後本利和為 2 元。

$$1\times(1+1)^{1}=2$$

(2) 假設借款金額為一元，年利率為 100%，每月複利一次，則一年後本利和為 2.613。

$$1\times\left(1+\frac{1}{12}\right)^{12}=2.613$$

(3) 假設借款金額為一元，年利率為 100%，每日複利一次，則一年後本利和為 2.714。

$$1\times\left(1+\frac{1}{365}\right)^{365}=2.714$$

(4) 假設借款金額為一元，年利率為 100%，每秒複利一次，則一年後本利和為 2.718。

$$1\times\left(1+\frac{1}{365\times24\times60\times60}\right)^{365\times24\times60\times60}=2.718$$

(5) 細菌的繁殖、電容的充放電等，這些都是數量非常龐大的成長現象，都要用 e 來解釋才能理解。也就是數量非常龐大的物件，若其一年「平均」成長 2 倍，則一年後總數量會成長為 2.718 倍，而不是 2 倍。

(6) 請根據以上演算法，求出 e 值。