

再版序

本書承蒙各校踴躍採用與提供建言，所以修改一些錯誤，且將一些觀念再舉例深入剖析。其次，因為每年都有參加 APCS 考試，雖然主辦單位都沒有公告題目，但我是認為這些題目真的很好，唯有透過實作，才能提高程式設計運算思維，所以這次再版也加入一些學術科最新考題。

關於範例程式

筆者程式設計教學超過 30 年，我是認為學生將範例程式打一次，也是一個很好的實習，所以我們不提供範例程式。因為範例程式使用『複製與貼上』沒有自己打字，這樣學生沒辦法練習除錯，沒辦法發現學習的盲點，最後程式一大，真的不知從何除錯。

關於實作與習題解答

程式設計貴在能『依樣畫葫蘆』與『舉一反三』，因為本書實作題真的都和範例很相近，部分學生可以參考範例很快就做出解答，若無法馬上做出解答，也不要氣餒，不要馬上想看解答，因為只有不斷的揣摩範例，不斷的思考，這樣得到的靈感才會快樂，所以我們也不提供實作解答。其次，習題就有一點進階，這就提供給有意深入程式設計殿堂的禮物，所以也不提供解答，這樣有興趣的學生才有精進的機會，才能從中得到快樂。給您解答，只是一直害您，剝奪您得到快樂開竅的機會。其次，若是老師向出版公司訂書，因為老師日理萬機，還要班級經營與兼任行政，當然分身乏術，所以我們範例程式與實作解答僅提供給老師，敬請讀者見諒。

最後，本人雖然程式教學與著作超過30 年，快樂與熱情不變，儘管不斷的重複讀稿，但還是無法避免一些錯誤，企盼讀者閱讀本書時，將所見的錯誤、疑問、或不容易瞭解的地方郵件給我，如下表，我每年都會選出與抽出回函，贈送我的其他著作，在此先特別致謝。其次，我也會在泉勝網站刊登本書勘誤表。

讀者意見、內文觀念與詞句修正回函表

高中程式設計與 APCS 先修檢測 第二版			
頁數	行數	錯誤或不當詞句	建議修改為

其他批評與建議：			

以上意見、指正回函表，請郵寄至 aa163677@yahoo.com.tw，每年會抽出與選出精彩回函，寄發紀念品。

洪國勝僅識於彰化竹塘
www.goodbooks.com.tw

P2

- ◎ 單選題 (含題組)，以運算思維、問題解決與程式設計概念測試為主。
- ◎ 測驗題型，包括：程式運行追蹤 (code tracing)、程式填空 (code completion)、程式除錯 (code debugging)、程式效能分析 (code performance analysis)、及基礎觀念理解 (basic concepts understanding) 等。
- ◎ 題目若需提供程式片段，則以 C 語言命題。

◆ 命題內容領域

- 程式設計基本觀念 (basic programming concepts)
- 資料型態 (data types)，常數 (constants)，變數 (variables)，視域 (scope)：全域 (global) / 區域 (local)
- 控制結構 (control structures)
- 迴路結構 (loop structures)
- 函式 (functions)
- 遞迴 (recursion)
- 陣列與結構 (arrays and structures)
- 基礎資料結構 (basic data structures)，包括：佇列 (queues) 和堆疊 (stacks)
- 基礎演算法 (basic algorithms)，包括：排序 (sorting) 和搜尋 (searching)

P3

以撰寫完整程式或副程式為主。考試時間為 150 分鐘，題目有四題，可自行選擇以 C, C++, Java, Python 撰寫程式。

◆ 命題內容領域

- 輸入與輸出 (input and output)
- 算術運算 (arithmetic operation)，邏輯運算 (logical operation)，位元運算 (bitwise operation)
- 條件判斷與迴路 (conditional expressions and loop)
- 陣列與結構 (arrays and structures)
- 字元 (character)，字串 (string)
- 函數呼叫與遞迴 (function call and recursion)
- 基礎資料結構 (basic data structures)，包括：佇列 (queues)，堆疊 (stacks)，樹狀圖 (tree)，圖形 (graph)
- 基礎演算法 (basic algorithms)，包括：排序 (sorting)，搜尋 (searching)，貪心法則 (greedy method)，動態規劃 (dynamic programming)

P6

目前 C/C++ 語言常用熱門的開發環境有 CodeBlocks、Dev-C++，因為 APCS 考場採用 CodeBlocks，所以我們介紹 CodeBlocks 如下：

CodeBlocks

CodeBlocks 是目前最熱門的 C/C++ 自由整合式開發環境 (IDE)，APCS 考試也提供此 IDE，CodeBlocks 官網如下圖(<https://www.codeblocks.org/>)：

G201

點選『Download』，畫面如下圖：

G202

點選『Download the binary release』，畫面如下圖：

G203

初學者請直接點選『codeblocks-20.03mingw-setup.exe』，因為它才有內含『C/C++』編譯軟體，不然要自備編譯軟體。

安裝軟體

下載的檔案就是『.exe』執行檔，請在『下載區』按兩下程式，就可自動執行。開啟

『CodeBlocks』，畫面如下圖：

G204

P17

(以上 7、8 兩個步驟，亦可點選工具列的『Build and Run』。)

9、以上 `cout << "Hello world!" << endl;`，就是本例所執行的程式，往後您要寫任何的程式，只要將此敘述刪除，並在此位置寫下您的程式就可以。

17-2

10、如何調整編輯器字型大小。

請點選功能表『Settings』/『Editor』，畫面如下圖，繼續點選『Choose』,就可以依照指示完成字型大小的設定。

G210

11、若出現找不到編譯器？（若之前您有安裝其他 C 語言，例如，DevC++, 有可能系統誤判而找不到編譯器）

(1) 請點選功能表『Setting』/『Compiler』，畫面如下，在這一畫面，您可以指定要使用哪一版本編譯器。(C 語言編譯器版本非常多，分別是 11、14 與 17 等等，往後線上查詢時就會明瞭)

G211

(2) 繼續點選『toolchain executables』，畫面如下圖，請留意您的編譯器路徑是否正確，若不正確請在此修改。

G212

實作

1、以下程式可計算長方形的面積與周長，請練習於 `main()`內輸入程式，並觀察執行結果。

```
int a=3;
int b=4;
int c,d;
c=a*b;
d=2*(a+b);
cout<<"面積="<<c<<endl;
cout<<"周長="<<d<<endl;
```

P20-1

```
#include <iostream>
using namespace std;
int main()
{
```

```
    printf("a\n");
    printf("aa");
    return 0;
}
```

請於開新專案時，一律點選『C++』就可以，因為 C++ 包容 C，所有 C 指令都可在 C++ 執行。

P20-2

常犯的錯誤

以下是一些鍵入程式時常犯的錯誤，整理如下，方便初學者除錯。

1、忘記在敘述行尾端打上分號（；），例如，

```
printf("Hello")
```

且分號『；』是英數下的半形，不是中文狀態下的全形。

2、大小寫未區分。C 語言規定大小寫的字母是不同的字元，應注意字母的輸入。例如，

```
printf("Hello");
```

```
Printf("Hello");
```

就不一樣了。

3、拼錯字。例如將

```
printf("Hello");
```

打成

```
prmtf("Hello");
```

那就錯了，將會出現錯誤信息。

4、括弧未配成對。有一個左大括弧『{』，務必要配一個右大括弧『}』；同樣地，左小括弧『(』務必要配一個右小括弧『)』；左中括弧『[』務必要配右中括弧『]』。初學者經常會忘記打上右括弧，導致語法錯誤。

5、引號未成對。不論雙引號『"』（用於字串），或是單引號『"』（用於字元）的使用，都應該成雙成對出現。

6、不該出現的空白符號。例如，把

```
printf("Hello");
```

打成

```
print f("Hello");
```

也不行。

P22

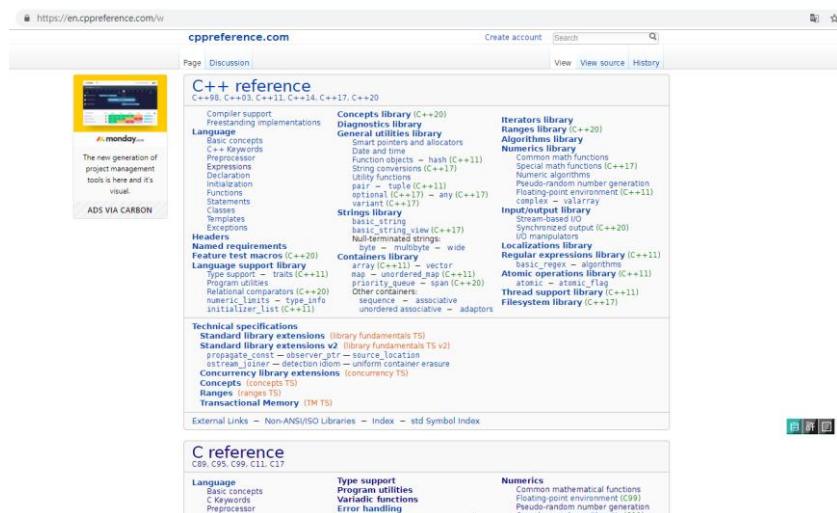
```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int    a;
    long   b;
    float  c;
    printf("Enter an integer:      ");
    scanf("%d", &a);
    printf("Enter a long integer: ");
    scanf("%ld", &b);
    printf("Enter a float:        ");
    scanf("%f", &c);
    printf("\n Your enter..... \n");
    printf("integer      a =%d  \n",  a);
    printf("long integer b =%ld \n", b);
    printf("float        c =%f", c);
    return 0;
}
```

以上程式是開新專案時，點選『C』的程式設計類型，還是要不斷提醒，C 環境無法執行 C++指令，但 C++環境可以執行 C 與 C++指令，但是初學者經常分不清楚，所以建議一律開啟 C++專案，這樣就不會有錯誤訊息。

P22_2

網路 C/C++語言使用手冊

網路 C/C++語言的使用手冊很多，但是，<https://en.cppreference.com/w>是我目前找到最齊全的參考手冊，有 C 也有 C++如下圖：



C 語言的詳細資料如下（請點選上圖的下面『C reference』）：上一章曾提到編譯器版本問題，請看下圖就會明瞭，若標示『C99』表示那是 1999 新增的指令；若標示『C11』表示那是 2011 新增的指令，沒標示則是 1990 制訂的標準。

G213

於上圖點選『[Input/output support](#)』，可得所有輸出入函數，請點選『printf』，可得到『printf』的語法，如下圖：

G214

因為 APCS 學科試題以 C 語言命題，所以還是要深入介紹 printf() 函式，printf() 函式的深入語法如下：

```
printf( "[f][w][.p][l]x" ,arg1,arg2)
```

f: 可以是 -、+，輸出預設靠右對齊，若要靠左對齊，請加負號(-)。例如，

```
int a=8;
printf("%5d\n",a);//      8
printf("%-5d\n",a);//8
```

正數輸出預設空白，若一定要加上正號，那就加上正號(+)。例如，

```
printf("%+5d\n",a);//+8
```

w: 用來指定資料的總欄位寬度。例如，

```
printf("%5d\n",a);//      8
```

.p: 用來指定小數點的長度。例如，

```
printf("%5.2f\n",b);// 0.03 共 5 格，. 一格，整數僅剩 2 格
```

l: 用來強調整數是長整數。

x: 常用格式有 d,f,c,s,u,e,E,g,G,o,x,X，說明如下表：

符號	說明
d	整數
f	浮點數
c	字元

s	字串
e,E	以科學符號表示
g,G	取浮點數或科學符號較短者輸出。例如，0.034 會輸出 0.034，但是 0.000000034 則會以科學符號表示。
o	轉為八進位輸出
x,X	轉為十六進位輸出

實作

1、請寫出結果，鍵入程式，執行程式，檢查是否正確。

題號	題目	執行結果
1	<pre>#include <iostream> int main() { int a=8; printf("%5d\n",a);// 8 printf("%-5d\n",a);//8 printf("%+5d\n",a);//+8 float b=0.034; float c=0.0000000034; printf("%5.2f\n",b);// 0.03 共 5 格，.一格，整數僅剩 2 格 printf("%e\n",b); printf("%E\n",b); printf("%g\n",b); printf("%G\n",c); int d=17; printf("%o\n",d);//21 printf("%x\n",d);//11 printf("%X\n",d);//11 return 0; }</pre>	
2	<pre>#include <iostream> int main() { int a=97; printf("%d\n",a); printf("%c\n",a); return 0; }</pre>	

2、請寫一個程式，程式一開始要求使用者輸入人名，有提示字『Please input

your name: 』，接著要求輸入身高,有提示字『input your height 』，最後輸出『Your Name is ...,your height is ... 』

P23

前面C語言使用printf()與scanf()函式作為輸出入，因為輸出入都要指定格式，有點不便，C++則使用cout與cin物件作為輸出入，請看以下說明。

P26

```
char c[5]; //字元陣列，代表字串，C style
cin>>c;
cout<<c<<endl;
```

可輸入字串至變數 c。C 因為要精簡編譯器，所以沒有 string 型態，所有字串處理都要使用字元陣列或指標，C++則彌補此缺憾，新增 string 型態，本書關於字串處理都直接使用 string 型態。但因為 APCS 考試的學科採用 C 語言，所以還是要介紹 C 語言的字串處理方式，這樣才看得懂題目，以下程式可輸入一個字串，並輸出。

```
string d; //字串，C++ style
cin>>d;
cout<<d;
```

若要輸入含空白字串，例如，"This is a book."，則要使用 getline()函式，程式如下：

```
string b;
getline(cin,b); //可輸入含空白字串
cout<<b;
```

以下程式則可連續輸入 3 個變數，輸入時，變數之間請以空白隔開，例如，輸入 3 4 5，則三個變數變數 a,b,c 分別得到 3,4,5。

```
int a,b,c;
cin>>a>>b>>c;
cout<<a<<" "<<b<<" "<<c;
```

實作

- 1、請寫一個程式，程式一開始要求使用者輸入人名，有提示字『Please input your name: 』，接著要求輸入身高,有提示字『input your height 』，最後輸出『Your Name is ...,your height is ... 』
- 2、請於『<https://en.cppreference.com/w/>』搜尋cout與cin的語法。（提示：請點選『[Stream-based I/O](#)』。

2_3 取亂數

程式設計常常需要一些任意整數來測試、或模擬結果，此時就需要請電腦取亂數，下面就介紹C與C++取亂數的方法。

rand()

C語言傳回0到32767的亂數，例如，

```
int a=rand();  
cout<<a;
```

就可傳回0到32767的任意整數。，使用前應先匯入

```
#include <stdlib.h>//使用rand()
```

但電腦不是萬能，要如何於亂數表任意選擇一個起點呢，因為每次執行的系統時間不同，所以就可以以系統時間，當作亂數的起點，全部程式如下：

```
#include <iostream>  
#include <time.h>//使用time()  
#include <stdlib.h>//使用rand()  
using namespace std;  
int main()  
{  
    srand(time(NULL));  
    int a=rand();  
    cout<<a;  
    return 0;  
}
```

以下程式，可取0到5的任意整數。

```
int a=rand()%6;
```

以下程式，可取1到6的任意整數。

```
int a=1+rand()%6;
```

以上可線上查詢如下。(點選『[Pseudo-random number generation](#)』)

G215

C++產生亂數的方法就多如牛毛，如下圖：

G216

以上當然令初學摸不著頭緒，本書當然不可能一一介紹，所以就摘錄可以簡單取亂數的方法如下：例如，以下程式可產生1到6的亂數，請讀者複製此程式，修改您自己的亂數範圍就可以。

```
#include <random>  
#include <iostream>  
using namespace std;  
int main()
```

```

{
    random_device rd; //Will be used to obtain a seed for the random number engine
    mt19937 gen(rd()); //Standard mersenne_twister_engine seeded with rd()
    uniform_int_distribution<> distrib(1, 6); //取1~6的整數亂數
    int a=distrib(gen);
    cout<<a;
    return 0;
}

```

實作

- 1、請分別以C與C++產生10到99的亂數，並輸出此數字。
- 2、請分別以C與C++產生小寫字元的亂數，並輸出此字元。

2-4 線上程式解題系統

目前好用的線上程式解題系統有『高中生程式解題系統』、『台中女中程式解題系統』與高雄市的『飆程式網』，這些都提供一些線上程式設計題目，可貼上您的程式，並由電腦評量是否正確且記錄每一個人的解題狀況。

高中生程式解題系統

高中生程式解題系統 (<https://zerojudge.tw/>) 如下圖：

G219

請自行註冊，以下是第1題的參考解答，請自行複製此程式，練習去解答區貼上此程式，直到系統傳回『AC』（表示通過），若有錯誤，也有一些提示。

```

#include <iostream>
#include <cstring>
using namespace std;
int main() {
    string s;
    cin>> s;
    cout<<"hello, "<<s<<endl;
    return 0;
}

```

台中女中程式解題系統

台中女中程式解題系統 (<http://www.tcgs.tc.edu.tw:1218/>) 如下圖：

G220

請自行註冊，它的第一題如下圖，有程式，請自行複製，點選『解題』，貼上，直到得到『AC』，表示通過。

G221

高雄市翹程式網

高雄市翹程式網 <http://khcode.m-school.tw/>，如下圖：

G222

請自行註冊，並練習解題。

泉勝教學網

泉勝教學網(www.goodbooks.com.tw)，內有很多程式設計教材PDF、自我練習題目，遊戲程式、最新APCS考題等（筆者有去考，所以累積一些較新試題，也在臉書蒐集一些高手所分享的試題與試題解答下載點），歡迎大家瀏覽。

P47

```
#include <iostream>
using namespace std;
int main() {
    char a[]="abcd";//C style
    cout<<a<<endl;//abcd
    char *b="qwer";
    cout<<b<<endl;//qwer
    b=a;
    cout<<b<<endl;//abcd
    b="asdf";//重新指派
    cout<<b<<endl;//asdf
}
```

P48

```
# include <iostream>;
include <cstring>
using namespace std;
int main() {
    int student=0;
    student=student+1;
    cout<<student<<endl;
    int a=3.4;
    cout<<a<<endl;
    int b=34
    cout<<b<<endl;
    char c,d;
```

```

    c="aa";
    cout<<c<<endl;
    d=65;
    cout<<d;
    string e;
    e=55;
    cout<<e<<endl;
    return 0;
}

```

P49

```

#include <iostream>
using namespace std;
int main() {
    bool a;
    cout<<sizeof(a)<<endl;
    char b;
    cout <<sizeof(b)<<endl;
    short c;
    cout<<sizeof(c)<<endl;
    int d;
    cout<<sizeof(d)<<endl;
    float e;
    cout<<sizeof(e)<<endl;
    long double f;
    cout<<sizeof(f)<<endl;
}

```

P50_1

例如，以下程式的變數a，宣告在大括號內，那其有效範圍僅在此區塊，就無法在敘述區塊外輸出。

```

include <iostream>
using namespace std;
int main()
{
    {///內稱為敘述區塊
        int a=2;
    }
}

```

```

        cout <<a;
    }
    a=3;
    cout <<a;
    return 0;
}

```

P50__2

例如，以下程式，main()的變數a，其有效範圍僅在main()函數內，就無法在aa()函數存取。

```

void aa(){
    a=3;#error
}
int main()
{
    int a=2;
    aa();//執行aa()函數
    cout <<a;
    return 0;
}

```

但是，如果把變數a宣告為全域，那就可順利存取，如以下程式：

```

#include <iostream>
using namespace std;
int a;//全域
void aa(){
    a=3;
}
int main()
{
    a=2;
    aa();
    cout <<a;//3
    return 0;
}

```

但是如果有全域宣告，但aa()內又宣告變數a，如以下程式，那aa()函數內變數a就是區域內變數了，那此範圍內的a，也是獨立的，不會回傳，這樣有一個好處，自己要用，就自己宣告，這樣才不會誤用全域變數，除非您知道此為全域變數，且要用此全域變數，那才可不用宣告，如上一個程式。

```

#include <iostream>
using namespace std;
int a;
void aa(){
    int a;
    a=3;
}
int main()
{
    a=2;
    aa();
    cout <<a;//2
    return 0;
}

```

以下變數a，也彼此獨立，不互相干擾，就如同不同的家庭，會有相同姓名相同的情況。這樣有一個好處，彼此互相獨立，才不會誤用別人的變數。

```

#include <iostream>
using namespace std;
void aa(){
    int a=3;
    cout <<"2:"<<a<<endl;//3
}
int main()
{
    int a=2;
    cout <<"1:"<<a<<endl;//2
    aa();
    cout <<"3:"<<a<<endl;//2
    return 0;
}

```

P51

```

#include <iostream>
using namespace std;
const double PI=3.14159;//全域常數
int a;//全域a
void aa();

```

```

void bb();
int main() {
    int b, c;
    a=5;//全域a
    c=5;//main 內 c
    aa();
    cout<<"2.main 的c="<<c<<endl;
    bb();
    cout<<"3.全域a="<<a<<endl;
    for (int a =2 ;a<=3;a++)
        cout<<"4.for 內的 a="<<a<<endl;
    cout<<"5.全域a="<<a<<endl;
    for ( a =2 ;a<=3;a++)//誤用
        cout<<"6.for 內的 a="<<a<<endl;
    cout<<"7.全域a="<<a<<endl;
    if (c>2){
        int a;
        a=1;
        cout<<"8.if 內a="<<a<<endl;
    }
    cout<<"9.全域a="<<a<<endl;
    return 0;
}
void aa()
{
    int c, d;
    c=3;//aa內的c
    cout<<"1.aa內的c="<<c<<endl;
}
void bb()
{
    a=8;//全域a
}

```

P56

在程式設計領域，此運算子稱為『指派』，

P57

```
int x=5, y=4;
cout <<x+y<<endl;    // 9
cout <<x-y<<endl;    // 1
cout <<x*y<<endl;    // 20
cout <<x/y<<endl;    // 1 兩者都是整數，商也是整數
cout <<x%y<<endl;    // 1
```

實作

1、請鍵入以上程式，並觀察執行結果。

整數除法或實數除法

C/C++語言的除法運算，只有被除數與除數的型態均為整數，才是整數除法，得到整數商。其次，只要被除數與除數有一型態為實數即為實數除法，得到實數商。例如，

```
int x=5, y=4;
double z=4;
cout <<x/y<<endl; //1
cout <<x/z<<endl; //1.25
cout <<y/x<<endl; //0
cout <<z/x<<endl; //0.8
z=x/y;
cout<<z<<endl; //1
```

整數除法與取餘的應用

整數除法與取餘可以將一個整數分解為數個數字。例如，若要將一個整數 152 分解為 1,5,2 三個數字，其方法如下：

```
int a=152;
int a3=152%10;//個位數
a=a/10;
int a2=a%10;//十位數
a=a/10;
int a1=a ;//百位數
cout<<a1<<a2<<a3;取餘
```

若要讓一個數字在累加的過程中，保持一定的循環，那也是用取餘。例如，

```
while(1)  { //無窮迴圈，請看第六章
    b=a[i];
    i=(i+1) %4;
}
```

那 i 就永遠在 0,1,2,3 循環。

自我練習

- 1、請指派一個四位數，並將其分解輸出為四個數字。例如，輸入 1234，那可以輸出 4,3,2,1，且輸出其數字和，本例為 10。
 - 2、請指派一個 0 到 86399 的整數，並分解為 HH:MM:SS 等時間格式。
 - 3、請任意指派一個 5 位數，請問如何分解為兩個一組。例如，a=25678，那如何分別得到 78,67,56,25 等 4 個數字；其次，要如何得到 78,56,2 等三組數字呢。
 - 5、請任意指派一個 5 位數，那如何得到最左邊與最右邊數字的和。例如，a=12345，那結果是 6。
 - 6、請任意指派一個 5 位數，那如何得到最左邊兩位數與最右邊兩位數字的和。例如，a=12345，那結果是 12+45=57。
 - 7、請任意指派一個 6 位數，那如何得到最左邊三位數與最右邊三位數字的和。例如，a=123456，那結果是 123+456=579。
 - 8、請任意指派一個 6 位數，並計算其奇位數與偶位數之和。例如，a=123456，那結果是 12 與 9。
- (補充說明：以上方法不只一種，方法沒有優劣，只要自己想出來的都是最好的。其次，以上即為 APCS 考試相近基礎題)

次方與根號

P57_2

但以上數值函數，C 語言是放在 math.h，C++ 是放在 cmath，因為 C++ 包含 C，所以在標頭檔通通使用

```
#include <cmath>
```

載入函式，請留意以上敘述載入標頭檔，不用分號『;』結束。

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。

P59

- 2、二元一次方程式 $ax^2+bx+c=0$ ，有實數解的條件是 $b^2-4ac \geq 0$ ，請將此數學語言轉為 C++ 程式語言。

p60

```
bool a=1;
```

```
int x=3,y=-4;
cout<<a<<endl;//1
cout<<!a<<endl;//0
a=x>0 && y>0;cout <<a<<endl;//0
a=x>0 || y>0;cout <<a<<endl;//1
```

P61

2、請指派1個x，判斷是否 $x < 3$ 或 $x > 7$ 。請以C/C++表示。

3. 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/C++ 語言表示？(提示：任兩邊之和大於第三邊的數學語言是：

$$a+b > c \text{ and } a+c > b \text{ and } b+c > a$$

請將以上數學語言轉為C/C++ 語言)

p63

關於以上運算子的詳細說明可點選<https://en.cppreference.com/w> 裡面的

『Expressions』，如下圖：

G218

繼續點選上圖的『operator precedence』，還可得運算子的優先順序，如下表：

(https://en.cppreference.com/w/cpp/language/operator_precedence)

G217

P64

```
#include <iostream>
using namespace std;
int main()
{
```

```

int a,b,c;
a=1;b=2;b=3;
a=a+a*2/2-3%2;
cout<<a<<endl;
a=5;b=4;
bool d;
d= a>2 && !(b>3) ||(a==3);
cout<<d<<endl;
return 0;
}

```

P71

兩個變數的內容要交換，就如同兩隻手的東西要交換位置。所以假設你有a,b

兩隻手，各拿一樣東西，那要交換其位置的方法如下：

1. 先找來第3 隻手t。
2. 將a 的東西交給t 暫存。

```
t=a;
```

3. 將b 的東西交給a。

```
a=b;
```

4. 將t 的東西交給b，而完成兩隻手東西的交換。

```
b=t;
```

5. 其次，若未先找來第三之手t 幫忙，電腦並沒有這個能力，同時拋出雙手東西，再同時接住另一手的東西，如以下敘述：

```
a=b;
```

```
b=a;
```

6. 這就是一種運算思維，電腦運算要很明確，全部程式如下：

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int a=3,b=4;
```

```
    int t;
```

```
    t=a;a=b;b=t;
```

```
    cout<<"a="<<a<<endl;
```

```
    cout<<"b="<<b<<endl;
```

```
    return 0;
```

```
}
```

P73

要電腦計算總分與平均，就要先將這些資料以變數儲存到電腦記憶體，才能進

一步計算，本例以name儲存姓名，a1,a2,a3,a4儲存成績，sum儲存總和，avg

儲存平均。所以程式如下：

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
string name;//儲存姓名

int a1,a2,a3,a4;//本例使用4個int變數

int sum;//本例使用1個int變數

float avg;

cout<<"Input Name:";

cin>>name;

cout<<"Input a1:";

cin>>a1;

cout<<"Input a2:";

cin>>a2;

cout<<"Input a3:";

cin>>a3;

cout<<"Input a4:";

cin>>a4;

sum=a1+a2+a3+a4;

avg=sum/4;

cout<<"Name is :"<<name<<endl;

cout<<"a1="<<a1<<endl;

cout<<"a2="<<a2<<endl;
```

```

    cout<<"a3="<<a3<<endl;

    cout<<"a4="<<a4<<endl;

    cout<<"sum="<<sum<<endl;

    cout<<"average="<<avg<<endl;

    return 0;

}

```

P74

(本習題請先不要用陣列與迴圈，這樣才能體會陣列與迴圈的優點)

資料的抽象化

若有一直線方程式是 $ax+by+c=0$ ，點 $p(m,n)$ 到直線的距離是

$$d = \frac{|am+bn+c|}{\sqrt{a^2+b^2}}$$

(提示：傳回整數絕對值函式是abs()，傳回實數是fabs())

這件事要由電腦作，那就是要將資料從人類約定的書寫習慣 $ax+by+c=0$ 獨立抽

象出來。例如，直線方程式是 $3x+4y+5=0$ ，求點 $p(1,2)$ 到直線的距離，首先，以

```
a=3;b=4;c=5;
```

約定此為直線

```
3x+4y+5=0
```

然後以

```
m=1;n=2;
```

約定此為 $p(1,2)$ ，所以全部程式如下：

```

#include <iostream>
#include <cmath>
using namespace std;
int main()
{

```

```

int a,b,c,m,n;
double d;
a=3;b=4;c=5;
m=1;n=2;
d=fabs(a*m+b*n+c)/(sqrt(a*a+b*b));
cout<<d;//3.2
return 0;
}

```

又例如，您要判斷 $q(1,-2)$ 是否在直線上，也是將資料抽象出來，程式如下：

(if 請看下一章)

```

a=3;b=4;c=5;
m=1;n=-2;
if ((a*m+b*n+c)==0)
    printf("yes");
else
    printf("no");

```

以上 $3x+4y+5=0$ 是人類在數學上書寫的約定，電腦只要給 a,b,c ，我們就約定此

為 $ax+by+c=0$ ，此稱為資料的抽象化或資料的數位化。

P75

實作

- 1、同上範例，可繼續輸入一個點，求此點與直線的距離。
- 2、點斜式。給定一個點，與斜率，求其直線方程式。例如，給定 $(1,2),m=3$ ，則此直線方程式 $3x-y-1=0$ 。
- 3、斜截式。給定兩個截距，求出此直線方程式。例如，某一直線， x 截距是3， y 截距是4，則其直線方程式是 $4x+3y=12$ 。

範例 4_1e 時間差的計算。假設某一汽車從 $h1=3,m1=20$ 進入， $h2=5,m2=10$ 離開，請寫一程式可以計算其停留時間。

執行結果

```
t=110 分,等於 1:50
```

演算法

關於兩個時間差的計算，小學是用直式算法，若分鐘不夠要向『時』借位，但

電腦就強了，直接通通化為分鐘，就可相減。若是兩個日期差，也是通通轉為日，再直接相減。關於兩個日期差請看第 7 章。所以學電腦程式設計，就要慢慢調整成電腦的運算思維。

程式列印

```
#include <iostream>
using namespace std;
int main() {
    int h1=3,m1=20;
    int h2=5,m2=10;
    int t1,t2,t;
    int h,m;
    t1=h1*60+m1;
    t2=h2*60+m2;
    t=t2-t1;
    h=t/60;
    m=t %60;
    cout<<"停留時間="<<t<<endl;
    cout<<h<<":"<<m<<endl;
    return 0;
}
```

實作

1、找零問題。請寫一程式，可以協助付款 1000 元時，計算所需找回的錢，並以最節省鈔票與硬幣數量的方式找錢。(APCS105 試題)

p76

一、程式執行結果

本章每次必考3~5題，且一定要拿到分數。以下是一些模擬試題，其實寫程式沒人這樣寫，但這樣才能測驗學生對本章的瞭解程度。請先紙上寫出答案，再在電腦執行比對答案，這樣才會進步。

題號	題目	結果
1	//Arithmetic Operator //The Precedence of Arithmetic //The Associativity of Arithmetic #include <iostream> using namespace std; int main() {	

	<pre> int a=1,b=2,c=3,d=4,e=5; cout<<a*b-c*-d<<endl; cout<<a/b*c+d+ ++e<<endl; a=(b=2)+(c=3); cout<<a<<endl; //上式括號可否去掉 a=1;b=2;c=3; a+=b+=c+=5; cout<<"a"<<a<<endl; cout<<"b"<<b<<endl; cout<<"c"<<c<<endl; a=1;b=2;c=3; a=++b + ++c; cout<<a<<","<<b<<","<<c<<endl; a=b++ +c++; cout<<a<<","<<b<<","<<c<<endl; a=++b + c++; cout<<a<<","<<b<<","<<c<<endl; a=b-- + --c; cout<<a<<","<<b<<","<<c<<endl; a=1;b=2;c=3; d+=b+c; cout<<"d"<<d<<endl; d*=a+b+5; cout<<"d"<<d<<endl; return 0; } </pre>	
2	<pre> //Relational Operator #include <iostream> using namespace std; int main() { int a=1; cout<<(a>4)<<endl; cout<<(a>=0)<<endl; cout<<(a==1)<<endl; cout<<(a=5)<<endl; cout<<(a!=1)<<endl; } </pre>	

	<pre> cout<<(a=>4)<<endl;//錯在哪 cout<<(a!=5)<<endl; //錯在哪 int a=1,b=2,c=3,d=4,e=5; c=(a+3)==b; cout <<c<<endl; c=(a+1)==b; cout <<c<<endl; return 0; } </pre>	
3	<pre> #include <iostream> using namespace std; //Logital Operator //The Precedence of Logical //The Associativity of Logical int main() { int a=5,b=4,c=3,d=2,e=1; cout<<((a!=b) & (b>=3))<<endl; cout<<((a!=b) ^ !(b>=3))<<endl; cout<<(!(a!=b) !(b>=3))<<endl; cout<<((a==1)& (a>0) && (b>=2))<<endl; cout<<((a==1) (a>0) & (b>2) ^(c==3))<<endl; cout<<((a==1) (a>0) && (b>2) ^(c==3))<<endl; return 0; } </pre>	
5	<pre> #include <iostream> using namespace std; int main() { int a=4;b=5,t; t=a;a=b;b=t; cout<<a<<endl; cout<<b<<endl; return 0; } </pre>	

P83-1

int main() {	int main() {
--------------	--------------

<pre> int a=2,b=0,c=0; if (a==0) b=3; c=3;//縮排，不改其效果 cout<<b<<endl;//0 cout<<c<<endl;//3 } </pre>	<pre> int a=2,b=0,c=0; if (a==0) b=3; c=3; cout<<b<<endl;//0 cout<<c<<endl;//3 } </pre>
---	---

P83-2

<pre> int main() { int a=2,b=0; if (a==0); b=3; cout<<b<<endl;//3 } </pre>	<pre> int main() { int a=2,b=0; if (a==0); b=3; cout<<b<<endl;//3 } </pre>
--	--

83_3

7、也有人這樣寫。

```

int a=2,b=0;
if (a>0) b=3;

```

P87_1

```

#include <iostream>
using namespace std;
int main() {
    int x,y;
    char b=' ';
    cout <<"input x:";
    cin >>x;
    cout <<"input y:";
    cin >>y;
    if(x>0)
        if(y>0)
            b='1';    // 第一象限
        else

```

```

        b='3';        //第四象限
    else
        if(y>0)
            b='2';        //第二象限
        else
            b='4';        //第三象限
    cout<<x<<" "<<y<<" is ";
    cout <<" 第 "<<b<<"象限";
    return 0;
}

```

P87-2

2. 表決器。假設有一項評審工作，有3 位評審，當其中兩人或以上同意，則表示過關，請設計此程式。（請共用鍵盤就好）

3. 表決器。假設有一項評審工作，有5 位評審，當A 或B 任一個同意或B、C、D 全同意，則表示過關，請設計此程式。（請共用鍵盤就好）

4、請寫一個程式，電腦可以產生一個小寫字元，並由使用者輸入此字元，電腦評定是否正確。

5、請寫一個程式，電腦可以產生兩個 0 到 9 的亂數，並由使用者輸入相加結果，電腦評判是否正確。

6、請寫一程式，電腦可以產生一個小寫字元，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。

7、請寫一個程式，電腦可以產生一個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。

P93

```
#include <iostream>
using namespace std;
int main() {
    int a=2;
    string b=""; //季節 C++Style
    //char *b;//C Style
    switch(a)
    {
        case 1:
            b="Spring";    //春
            break;
        case 2:
            b="Summer";    //夏
            break;
        case 3:
            b="Fall";      //秋
            break;
        case 4:
            b="Winter";    //冬
            break;
        default :
            b="input error";
    }
    cout <<b;
    return 0;
}
```

P94

```
#include <iostream>
using namespace std;
int main() {
    int a;
    string b="";    /* 季節 */
    cout <<"Input a score:";
    cin >>a;
```

```

if(a>100 || a<0) { /* 如果輸入的成績大於 100 或小於 0，則列印錯誤訊息 */
    b="input error";
    return 0;      /* 提早離開 */
}
switch(a/10) {
    case 10:
    case 9:
        b="A";
        break;
    case 8:
        b="B";
        break;
    case 7:
        b="C";
        break;
    case 6:   case 5:   case 4:
    case 3:   case 2:   case 1:   case 0:
        b="D";
        break;
}
cout<<"The grade is  "<<b;
return 0;
}

```

P95

2、請將範例 5_1b 實作第 6、7 兩題，以 switch 重做。

P95-2

```

#include <iostream>
#include <stdlib.h> //使用 rand()
#include <time.h> //使用 time()
using namespace std;
int main() {
    int a,b;
    string a1="",b1="",r="";//C++ style

```

```

//char *a1="",*b1="",*r="";//C style
//time() 需要宣告#include <time.h>
srand(time(NULL));;//隨機選取亂數起點， 若未使用此指令， 每次亂數起點都相同
cout<< "input 0(剪刀) 1(石頭) 2(布) :";
cin>>a;

b=rand()%3;//產生 0， 1， 2 的亂數， 需要#include <stdlib.h>
switch (a){
    case 0:
        a1="剪刀";
        switch(b){
            case 0:
                b1="剪刀";
                r ="平手" ;
                break;
            case 1:
                b1="石頭";
                r ="電腦贏" ;
                break;
            case 2:
                b1="布";
                r ="你贏";
                break;
        }
        break;
    case 1:
        a1="石頭";
        switch(b){
            case 0:
                b1="剪刀";
                r ="你贏" ;
                break;
            case 1:
                b1="石頭";
                r ="平手" ;
                break;
            case 2:
                b1="布";
                r ="電腦贏";

```



```

        break;
    }
    break;
case 2:
    a1="布";
    switch(b){
        case 0:
            b1="剪刀";
            r="電腦贏";
            break;
        case 1:
            b1="石頭";
            r="你贏";
            break;
        case 2:
            b1="布";
            r="平手";
            break;
    }
    break;
}

cout<<"你出:"<<a<<a1<<"，電腦出："<<b<<b1<<"，結果："<<r;
return 0;
}

```

P97

(提示：以上 2 個人，兩層 switch，3 個人就 3 層 switch)

範例5_2c 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：前面的scanf() 輸入字元還要按

『Enter』，但是getche() (放在conio.h) 不用，可使用此函式輸入字元。

程式列印

```
#include <iostream>
```

```
#include <conio.h>//本例使用getche()
```

```

using namespace std;
int main()
{
    char a;
    while(1){//無限迴圈，請看下一章
        a= getch();//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;
}

```

實作

1. 請以if else 重做本範例。

2. 作出有升Key (黑鍵) 的電子琴鍵盤。
3. 還可先輸出按鍵值，提醒使用者該按哪一鍵，那就是電子琴教學機了。

5-3 位元運算子

相信大家都有作過複選題，請您思考一下複選題答案應該如何數位化。例如，答案有可能ABC，也有可能BE，那如何將這些資料數位化，並用適當的變數型態處理呢，是不是用一個字串表示呢？例如，以a="ABD"表示，這樣當然也可以，但每個答案就要占1 ~5 byte(1 byte=8bit)，但"ABD" 若是以11010 表示呢，"A" 就以10000 表示，這樣就可以用26(11010=24+23+21=26) 與16(10000=24=16) 表示答案，這樣竟然只要1byte 就可以記錄所有答案。好的，以上已經完成非常精簡的資料數位化方式，那電腦如何評分呢，答案就是電腦也備有位元運算子，可以錙銖必較的處理每個位元(bit)。

位元 (Bitwise) 運算子

位元 (Bitwise) 運算子是將一個8 位元的整數逐位元進行運算，此類運算子還可以分為兩類：位移 (Shift) 運算子與布林運算子。位移運算子可以用來將各個位元向左或是向右移；布林運算子則可以逐位元進行布林運算。下面是C++ 語言位元操作運算子的列表：

運算子	定義	結合律
<<	逐位元向左位移	由左至右
>>	逐位元向右位移	由左至右
~	逐位元 not 運算	由右至左
&	逐位元 and 運算	由左至右
^	逐位元 xor 運算	由左至右
	逐位元 or 運算	由左至右

<<

『<<』是左移運算子，例如，

```
a=1;
cout<<(a<<1);//2  1 向左移位 1 位元，結果是 2
a=1;
cout<<(a<<2);//4  1向左移位2位元，結果是4
```

>>

『>>』是右移運算子，例如，

```
a=4;
a=a>>1;
cout<<a;//2  4向右移位1位元，結果是2
a=7;
a=a>>2;
cout<<a;//1
```

~

『~』是逐位元not 運算，not 是將0 變1，1 變0，運算結果如下表：

c=~a	a
0	1
1	0

例如，

```
a=1;
printf("%d",~a);//-2
```

為什麼答案是-2 呢？這有點難懂，筆者簡單說明如下，因為電腦是以2 補數表示負數，a=1，分解為2 進位是00000001

$\sim a = \text{not } 00000001 = 11111110$ //逐位元反相

首位元是1 表示負數，那到底負多少，就取2 補數。2 補數的步驟是先取1 補數再加1。其次，所謂1 補數是『逐位元將1 變0，0 變1』，所以上面11111110 的1 補數是

00000001

將1 補數再加1，就是2 補數，上面00000001 加1 就是00000010，其大小是2。上面首位元符號既然是『1』，那就表示其為負數，所以11111110 代表此數是-2，所以輸出-2，此即為『2補數』的觀念。

補充說明

1. 正數的數位化。正數就直接以2 進位編碼，本例以8 位元為例，如下表：

正數	二進位編碼
0	00000000
1	00000001
2	00000010
3	00000011
4	00000100
5	00000101
6	00000110
7	00000111
254	11111110
255	11111111

2. 正負數的數位化：正負數的數位化，直覺的作法，就將最高位元放0 代表正數，放1 代表負數，那直覺的編法如下：

二進位編碼	數字
0000000	0
0000001	1
01111111	127
10000000	-0
10000001	-1
10000010	-2
11111110	-126
11111111	-127

由上表發現，『00000000』與『10000000』都對應到0，這樣不是一對一函數，這樣求反函數太複雜而浪費時間，所以科學家就不斷討論與修正，發現負數的轉換需要兩個步驟，第一，求其1 補數（將0變1，1 變0），第二，求2 補數（1 補數加1 稱為2 補數。）

2. 將負數部分取2 補數的過程如下：

數字	取正數	1補數	2補數
-0	00000000	11111111	00000000
-1	00000001	11111110	11111111
-2	00000010	11111101	11111110
-3	00000011	11111100	11111101
-126	01111110	10000001	10000010
-127	01111111	10000000	10000001
-128	10000000	01111111	10000000

4. 將上表合併正數，如下表：（請留意正數直接編碼，負數取2補數編碼）

數字	取正數	1補數	編碼
127			01111111
126			01111110
2			00000010
1			00000001
0			00000000
-0	00000000	11111111	00000000
-1	00000001	11111110	11111111
-2	00000010	11111101	11111110
-3	00000011	11111100	11111101
-126	01111110	10000001	10000010
-127	01111111	10000000	10000001
-128	10000000	01111111	10000000

5. 由上表可知，若負數先取2 補數再編碼，竟然就解決正負0的問題。這樣就1
對1 了，這樣函數、反函數通通有解，這就是為什麼負數要取2 補數了。而且
這種轉換竟然一石二鳥，也解決減法問題，因為

5-2

可以看成

5+(-2)

6. 由上表可知，剛剛

```
a=1; //a=00000001
cout<<~a; //~a=11111110 11111110 is -2
```

為什麼是『-2』了，若您還是沒有轉過來，麻煩您去買一個Arduino 實驗板，
就可以看到記憶體的內容是『11111110』，請看拙著『中小學自造與程式設
計，泉勝』。

實作

1. 請輸入以下程式，並觀察執行結果，且自行使用以上原理推敲，看答案是否
一致。

```

a=-3;

cout<<~a;

a=-127;

cout<<~a;

```

&

『&』是逐位元and 運算，and 運算真值表如下，當a 與b 同時為1，c 才得到

1：

c= a & b	a	b
0	0	0
0	0	1
0	1	0
1	1	1

例如，

```

a=1;b=5;

cout<<(a&b);//00000001 & 00000101=00000001=1

```

^

『^』是逐位元xor 運算，xor 運算真值表如下，當a 與b 不同時，c才得到1。

c=a^b	a	b
0	0	0
1	0	1
1	1	0
0	1	1

例如，

```

a=1;b=5 ;

cout<<(a^b);//00000001 ^00000101=00000100=4

```

|

『|』是逐位元or 運算，or 運算真值表如下，當a 與b 有一為1，c 就得到1。

c=a b	a	b
0	0	0
1	0	1
1	1	0
1	1	1

例如，

```
a=1;b=5;
cout<<(a|b); //00000001 | 00000101 =00000101=5
```

範例3_5a複選題評分。大學學測考試的複選題共五個選項，五個選項全答對得5分，錯一個選項得3分，錯兩個選項得1分，錯超過兩個選項得0分，未作答也是0分，且答錯不倒扣，請寫一程式，可以協助評分，請問您的資料結構為何？

資料結構

1. 答案要如何表示呢？若答案是ABE，那用ABE表示嗎？這樣雖然可以，但需要5個bytes，非常浪費記憶體，若將其看為二進位，那只要以 $(11001)_2 = (25)_{10}$ 表示，也就是1個byte就足夠。

執行結果

(下圖是b=17，表示AE，答對4個，錯1個，所以得3分)

G501

程式列印

```
#include <iostream>
using namespace std;
int main()
{
    int a=25; //標準答案ABE
```

```

int b=25;//答對5個5分
b=17; // AE答對4個，錯1個3分
//b=1; // E答對3個，錯2個1分
//b=2;//D 0分
//b=0;//未作答
int c=0;
int d=0;//答對幾個
int e=0;//得分
if (b==0)
e=0;
else {
c=a^b;//逐位元xor，位元相同是0，不同是1
d=0;
cout<<c<<endl;
c=~c;//取補數，1變0，0變1
cout<<c<<endl;
d=d+(c & 1);//計算1的個數
//d=d+c & 1 // 錯，因為+的優先順序大於&
c=c>>1;//右移
d=d+(c & 1);
c=c>>1;
d=d+(c & 1);
c=c>>1;
d=d+(c & 1);
c=c>>1;
d=d+(c & 1);
}
if (d==5)
e=5;
else if(d==4)
e=3;
else if (d==3)
e=1;
else
e=0;
cout<<d<<endl;
cout<<e;
return 0;

```

```
}
```

P102

這題後來成為APCS考題，但題目強調『輸入包含若干筆測是資料』，所以要將輸入資料放在while迴圈，所以程式如下：(while迴圈請看下一章)

```
#include <iostream>
#include <cmath>
using namespace std;
int main() {
    int a, b, c, t;
    int ab, cc;
    string style=" ";
    while (cin>>a>>b>>c){
        /* 三邊長比較與交換，將 a 換為最小，b 其次，c 為最大 */
        if(a>b)
            { t=a; a=b; b=t; }    /* a, b交換 */
        if(b>c)
            { t=b; b=c; c=t; }    /* b, c交換 */
        if(a>b)
            { t=a; a=b; b=t; }    /* a, b交換 */
        cout<<a<<" "<<b<<" "<<c<<endl;
        if(a+b<=c)                /* 確定最小的兩邊之合是否小於第三邊 */
            style="No";           /* 無法構成三角形 */
        else{
            ab=pow(a,2)+pow(b,2);  /* 最小的兩邊長之平方合值 */
            cc=pow(c,2);           /* 第三邊長之平方值 */
            if(ab>cc)              /* 確認三角形種類 */
                style="Actue";    /* 銳角三角形 */
            else
                if(ab==cc)
                    style="Right"; /* 直角三角形 */
                else
                    style="Obtuse"; /* 鈍角三角形 */
        }
    }
}
```

```

    }
    /* 輸出三角形之種類與面積 */
    cout << style<<endl;
}
return 0;
}

```

P104

範例5-4d 紅綠燈程式。請觀察路口的紅綠燈，請設計程式完成其功能。

運算思維

- 工業控制領域，通常將輸出結果和時間建立一個關係，此稱為
時序圖，本例先以單向為例，所以時序圖如下：

時間	輸出
0	綠燈
1	綠燈
2	綠燈
3	綠燈
4	綠燈
5	黃燈
6	黃燈
7	紅燈
8	紅燈
9	紅燈
10	紅燈

- 您要讓程式重覆執行，那就要一個無窮迴圈如下：（下一章會介紹）

```

while (1){
}

```

- 要讓程式依照時序且無限循環不會爆掉，就要聯想取餘運算『%』。

```

while (1){
    i=(i+1)%11;
}

```

```
}
```

3. 全部程式如下：system("cls");可讓輸出固定不捲動，請自行去掉，並比較輸

出結果。

```
#include <iostream>
using namespace std;
int main()
{
    string g="Green";
    string y="Yellow";
    string r="Red";
    string out;
    int i=0;
    while (1){
        system("cls");
        switch(i){
            case 0:
            case 1:
            case 2:
            case 3:
            case 4:
                out=g;
                break;
            case 5:case 6:
                out=y;
                break;
            case 7:case 8:case 9:case 10:
                out=r;
                break;
        }
        cout<<out<<endl;
        _sleep(1000);//wait a second
        i=(i+1)%11;//let i must in 0,1,2..10
    }
    return 0;
}
```

自我練習

1. 觀察十字路口雙向紅綠燈，規劃其時序，寫出其程式。

範例5-4e

時鐘程式。請寫一個可以顯示『hh:mm:ss』的24 小時時鐘程式。

執行結果

1. 以下輸出結果未使用system("cls"); 若使用system("cls");，則第一次執行要等一會，不是當掉，但是輸出位置會固定。

資料結構與演算法則

1. 時鐘就是每秒秒數加1，所以資料結構就是指派秒數的變數為t，程式如下：

```
int t=0;
while(1) {
    t=(t+1);
    _sleep(1000); //delay one second,延遲1秒
}
```

2. 考慮資料範圍與型態。

1天24 小時，每1 小時60 分，每1 分60 秒， $24*60*60=86400$ ，short 的範圍是-32768~32767，86400已經出範圍，所以要取int 或 long。

```
int t;
```

3. 時鐘就是每秒秒數加1，秒數到深夜12 點要歸零，所以程式如下：

```
while(1) {
    t=(t+1) % 86400; //60 * 60 * 24=86400
    _sleep(1000); //delay one second,延遲1秒
}
```

4. 將秒數使用整數除法與取餘，就可以到時，分、秒。（人類要看時、分、秒，電腦不用，電腦僅用0~86399表示時間，此即為電腦運算思維）

```

h=t/3600;

m=(t-h*3600L)/60;

s=t%60;

```

5. 全部程式如下：

```

#include <iostream>

#include <cstring>

using namespace std;

int main(){

    int t=23*60*60+58*60+55;

    short h,m,s;

    while (1){

        system("cls");

        h=t/3600L;

        m=(t-h*3600L)/60;

        s=t%60;

        cout<<h<<":"<<m<<":"<<s<<endl;

        _sleep(100);

        //_sleep(1000); delay one second

        t=(t+1)%86400;

    }

    return 0;

}

```

6. 本例有人會將資料結構先指派時、分、秒等三個變數

```

short h=23,m=58,s=50;

```

所以程式會這樣寫

```

#include <iostream>

```

```

using namespace std;

```

```

int main(){

```

```

    short  h=23,m=58,s=50;

```

```

    while (1){

```

```

s=s+1;

if (s>=60){

    s=0;

    m=m+1;

    if (m>=60){

        m=0;

        h=h+1;

        if (h>=24)

            h=0;

    }

}

cout<<h<<":"<<m<<":"<<s<<endl;

_sleep(100);

//_sleep(1000); delay one second

}

return 0;

}

```

7. 這樣雖然也可以，但是這樣對於往後時間的長度計算與時間的前後判斷會很麻煩，例如，於停車場的時間計算，前者是


```
t1=22000
```

```
t2=52000
```

兩個t 直接相減就可得到停車時間，但是本例將會有

```
t1=08:20:30
```

```
t2=09:30:40
```

那這兩個時間的差額計算就有點麻煩了。

範例5_4f 停車場計費。請寫一個程式，程式一執行，使用者按『1』鍵

時，表示開始計時，輸出進場時間，當使用者按『2』鍵時，表示結束計

時，輸出離場時間，與停車時間。

執行結果

G504

程式列印

```
#include <ctime>
#include <conio.h>
using namespace std;
int main(){
    time_t t,t1,t2;
    char a;
    int h,m,s;
    while (1){
        a=getche();
        if (a=='1'){
            t1=time(0);
            cout<<endl;
            cout<<t1<<endl;
            cout<<ctime(&t1);
        }
        if (a=='2'){
            t2=time(0);
            cout<<endl;
            cout<<t2<<endl;
```

```

        cout<<ctime(&t2);
        t=t2-t1;
        s=t %60;
        m=(t/60)%60;
        h=t/3600;
        cout<<"Total Time:"<<h<<":"<<m<<":"<<s<<endl;
        //s=t2 %60;
        //m=((t2%38400)/60)%60;
        //h=(t2%38400)/3600;
        //cout<<"Current Time:"<<h<<":"<<m<<":"<<s<<endl;
    }
}
return 0;
}

```

5-5 APCS試題

二、邏輯運算子(Logic Operators) (APCS10610試題)

此題題目請自行至APCS官網下載，參考程式如下：

```

#include <iostream>
using namespace std;
int main() {
    int a, b, c;
    bool found=false;
    cin>>a>>b>>c;
    if (a>0)
        a=1;
    if (b>0)
        b=1;
    if ((a&b) == c) {
        cout<<"AND"<<endl;
        found = true;
    }
    if ((a|b) == c) {
        cout<<"OR"<<endl;
        found = true;
    }
}

```

```

    }
    if ((a^b) == c) {
        cout<<"XOR"<<endl;
        found = true;
    }
    if (!found) { //若AND OR XOR都沒有
        cout<<"IMPOSSIBLE"<<endl;
    }
    return 0;
}

```

題目強調，『輸入包含若干筆測試資料』，請自行修改以上程式。

p108

11	<pre> //The Bitwise Operator #include <iostream> using namespace std; int main() { int a=3,b=-2,c=3,d=4; cout<<~a<<endl; cout<<~b<<endl; cout<<(a<<2)<<endl; cout<<(a>>2)<<endl; cout<<(b<<2)<<endl; cout<<(b>>2)<<endl; cout<<(a & b)<<endl; cout<<(a b)<<endl; cout<<(a ^ b)<<endl; cout<<(a & b c ^ d)<<endl; a<<=3; cout<<a<<endl; b>>=2; cout<<a<<endl; return 0; </pre>	
----	---	--

	}	
12	<pre> int main() { int a=5,b=4,t; if (a>b) t=a;a=b;b=t; cout<<a; return 0; } </pre>	
13	<pre> int main() { //The else default to the nearest a=-5;b=4;t=0; if (a>0) if (b>0) t++; else t--; cout<<t<<endl; a=-5;b=4;t=0; if (a>0) if (b>0) t++; else t--; cout<<t<<endl; //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) t++; } else t--; cout<<t<<endl; return 0; } </pre>	

P110

提示：S點若在三角形ABC內，則

（SAB 面積+SBA 面積+SCA 面積）一定等於 ABC 的面積。

P111

例如，要輸出『Hello』十次，那要將cout<<"Hello";寫十次嗎？

p113

9、計數變數的改變量可為正或負的整數或實數，上面程式都是正，迴圈運算式要用『小於等於』結束。以下程式則是負，改變量為負，就如同倒數，迴圈運算式要用『大於等於』結束。例如，以下程式是倒數輸出 654321。

```
for (int i=6;i>=1;i--)  
    cout <<i;//654321
```

也就是箭號的方向就代表數字變化的趨勢，箭號方向錯了，結果就錯。以下程式，就沒有錯誤信息，也沒有任何輸出，因為一開始 6 就沒有小於 1，就結束了，當然沒輸出。

```
for (int i=6;i<=1;i--)  
    cout <<i;//沒有輸出
```

以下程式計數變量為 2。

```
for (int i=1;i<=10;i=i+2)  
    cout <<i;//13579
```

以下程式計數變量為-3。

```
for (int i=9;i>=1;i=i-3)  
    cout <<i;//963
```

P114-1

1. 題目要印出1,2,3...10，這可以cout<<1;cout<<2;cout<<3...，但是電腦有迴圈指令，當然用一個迴圈就可以解決。

2、要求1+2+3...+10的和，此一數學問題，國小、國中學生就直接一個一個累加，高中數學稱此為等差級數，求和可用公式法，此公式為『((首數+ 末數)* 項數)/2』。

3. 電腦的特性因為速度快，所以可不用學習、不用學習先判斷此為等差級數，

再套用等差級數的公式，而是直接一個一個累加，此為電腦的強項，稱為『循序暴力法』，程式如下：

P114_2

- 4、 請讓電腦自動產生10題心算題，被加數與加數都僅有1位數，且讓使用者回答，電腦並評判對錯與累加正確與錯誤題數。
- 5、 請讓電腦自動產生10個小寫字元，讓使用者鍵入此字元，電腦並評判此字元是否正確，並累加正確與錯誤題數。
- 6、 請讓電腦10個小寫字元，讓電腦統計母音與子音的個數，並輸出結果。
- 7、 電腦自動產生10個1到6的亂數，請統計1到6各出現幾次。
- 8、 電腦自動產生50個0到100的亂數，統計0~59,60~69,70~79,80~89,90~100等區間人數。

P117

這題第五章決策指令已經說明如何判斷任意年是否閏年，現在只要多一個迴圈，讓年數從1到2000自動跑一遍就好，這就是電腦快速神奇的運算思維，所以程式如下：

```
#include <iostream>
using namespace std;
int main() {
    int a, i, c=0, a4, a100, a400;          /* c=閏年個數 */
    for(a=1; a<=2000; a++) {
        a4 = a % 4;                          /* 取餘數 */
        a100 = a % 100;
        a400 = a % 400;
```

```

        if(a4==0){
            if(a100==0){
                if(a400==0) {
                    c++;
                    cout<< c<<" "<< a<<endl;    /* 閏年    printf("a leap
year(4)\n"); */
                }
            }
            else {
                c++;
                cout<< c<<" "<< a<<endl;    /* 閏年    printf("a leap
year(2)\n"); */
            }
        }
    }
    return 0;
}

```

4、

P119

這一題，每一個數都要乘以2，乘以3，...到乘以9，且運算規則都一樣，所以適用迴圈，迴圈就從2到9，所以程式如下：

P121-1

本例輸入a，要檢驗2到a-1是否能整除，這就要靠迴圈了，這樣才會省力，所以程式如下：

```

#include <iostream>
using namespace std;
int main() {
    int a, i;
    bool prime=true; //先假設每一個數都是prime
    cout<<"input a integer: ";
    cin>>a;
    for(i=2; i<=a-1; i++)//試驗2到該數減1
        if(a%i==0)    {

```

```

        prime=false;//找到1個整除,就不是
        break;//找到1個整除，就提早離開
    }
    cout <<a;
    if(prime)
        cout<<" is an prime number";        /* 質數 */
    else
        cout<<" is not a prime number";      /* 不是質數 */
}

```

P122

前面我們已經寫出判斷一個整數是否質數，現在只要在外面多一個迴圈，將2到1000的整數都跑一遍，所以程式如下，但請留意，每個人還沒試驗以前，都要先假設其為質數。

```

#include <iostream>
using namespace std;
int main() {
    int i, j, k=0;
    bool prime;
    cout<<"prime numbers between 2 to 1000\n";
    for(i=2; i<=1000; i++) {
        prime=true;//初學者很容易忘這一行
        for(j=2; j<i-1; j++)    {
            if(i%j==0)        {
                prime=false;
                break;
            }
        }
        if(prime)    {
            cout.width(5);
            cout<< i;
            k++;
            if(k%10==0)
                cout<<"\n";
        }
    }
    cout<<"\n Total prime numbers is "<< k;    /* 總計2至1000間的質數總數 */
    return 0;
}

```



```
}
```

P123

前面已經介紹作出一個整數的九九乘法，現在只要2到9通通套用上一程式，所以程式如下：

P125

這一題，以前沒有電腦，那教授的門生就要從1000到9999的整數，一個一個計算，現在有電腦了，這些人就失業了，直覺的方法就是千位數1到9用一個迴圈，百位數0到9一個迴圈，個位數0到9一個迴圈，這樣就可以列出1000到9999的所有整數，所以程式如下：

P125_2

- 1、本例也可以用一個迴圈，列出1000～9999的所有整數，然後分解、判斷是否符合條件，請自己完成此程式。

P131

本範例介紹另一種播放亂數的方法，程式如下：

P132

- 4、請寫一程式，讓電腦每次產生3個亂數，直到此3個亂數完全遞增。例如，出現1,3,6稱為完全遞增，出現2,5,1就沒有。

P135

範例6_3b 請寫一程式，可以將所指派的整數（整數範圍），以反向的方式輸出。例如，a=2345，則輸出5432。

程式列印

```
#include <iostream>
using namespace std;
int main() {
```

```

int a=34567;
do{
    int b=a%10;
    cout<<b;
    a=a/10;

} while (a!=0);
return 0;
}

```

自我練習

- 1、 同上範例，可以求得幾位數與其數字和。例如， $a=2345$ ，則輸出4位數與14。
- 2、 同上範例，可以求得奇數位與偶數位的和。例如，上例奇數位和是8，偶數位的和是6。
- 3、 同上範例，那如何從個位數分割為兩個兩個1組，不重複，並求其和。例如，上例是 $45+23=68$ 。
- 4、 同上範例，那如何從個位數分割為兩個兩個1組，數字重複，並求其和。例如，上例是 $45+34+23=102$ 。
- 5、 英打練習。請由電腦自動產生小寫字元，並由使用者輸入，直到輸入錯誤才停止，並統計正確的字元數。
- 6、 同上題，直到3分鐘到為止，並輸出打對的字元。
- 7、 心算練習。電腦自動產生兩個1位數，並由使用者回答，直到答錯為止，並輸出答對題數。
- 8、 同上題，直到3分鐘到為止，並輸出答對與答錯題數。
- 9、 若有一級數 $s=3+6+9+...$ ，請問加到何數字？共幾個數字？其和剛好超過100。
- 10、讓電腦一直產生小寫字元，直到出現母音為止。並統計共產生幾個亂數。

範例 6_3c 請寫一程式，可以累加所輸入的數字，設計階段並不知數字個數，直

到輸入值是 0 時，印出所輸入數字的和及平均。

執行結果

輸入資料時，可以連續輸入，資料之間以空白隔開，也可以每輸入一個資料，就按一下『Enter』鍵。

G602

程式列印

```
#include <iostream>
using namespace std;
int main() {
    int a,n=0,s=0;
    cin>>a;
    while (a!=0){
        s=s+a;
        n++;
        cin>>a;
    }
    cout<<s<<endl;
    cout<<s/n<<endl;
    return 0;
}
```

這種方式給測試資料，也可以直接把輸入資料放在 `while()` 括號內，程式如下。

```
#include <iostream>
using namespace std;
int main() {
    int a,n=0,s=0;
    while (cin>>a && a!=0){
        s=s+a;
        n++;
    }
    cout<<s<<endl;
    cout<<s/n<<endl;
    return 0;
}
```

```

#include <iostream>
#include <cstring>//使用to_string()
using namespace std;
int main() {
    int a=13,b=2,r;
    string d="";
    while (a>0){
        r=a%b;
        d=to_string(r)+d;//數值轉字串,請加入#include <cstring>
        a=a/b;
    }
    cout<<d<<endl;
    return 0;
}

```

P140_1

若要實數解，那就每次遞增0.1或0.01或0.001，請依照題目所要求的精密度。其次，因為實數運算結果不會等於0，所以要改為非常接近0就好，所以程式如下：

```

#include <iostream>
#include <cmath>
using namespace std;
int main() {
    double s=0,y=0;
    for (double x=-10 ;x<=10;x=x+0.1){
        y=x*x-3*x-10;
        if (fabs(y)<=0.001)
            cout<<x<<endl;
    }
    return 0;
}

```

實作

1、請用暴力法，求解一元三次方程式。例如， $x^3+4x^2+x-6=0$ 的解是1,-2,-3。

141-1

前面的循序法是一個一個的循序猜，每次猜值範圍僅減少一個，本單元的二分猜值逼近法，則是每次都是猜一半，若沒猜到，那猜值範圍就減少一半，但前提是要將所有答案排序。

P141-2

5、只要 $[x1, x2]$ 的範圍大於所需要的精密度（小數1位是0.1,小數兩位是0.01），那就重複步驟2,3,4。

6、跳出迴圈時， $x1, x2$ 已經靠的很近，此時 $x1$ 或 $x2$ 即為所求。

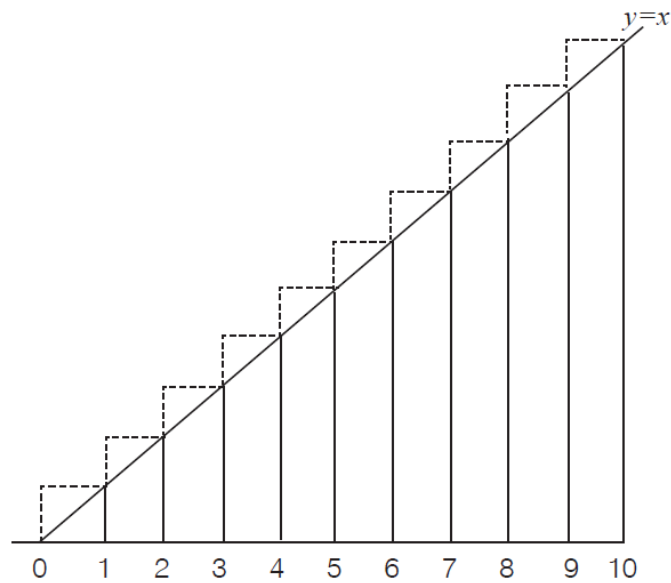
P142_1

```
#include <iostream>
#include <cmath>
using namespace std;
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;//猜的太小
        else
            x2=x;
    }
    while (fabs(x2-x1) >0.001) ; //只要x2-x1大於0.001就要繼續猜
    cout <<x1<<endl;
    return 0;
}
```

P142_2

2、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'，猜中時，也要告知共猜幾次。

143-0



143-1

```
#include <iostream>
using namespace std;
int main() {
    int sum=0;
    int dx=1;
    int y;
    for(int x=0; x<=10; x=x+dx){
        y=x;//y=x
        sum=sum+y*dx;//每一個矩形面積=高度*切割的寬度
    }
    cout<< sum;
    return 0;
}
```

143-2

```
#include <iostream>
using namespace std;
int main() {
    double sum=0;
    double dx=0.1;
    double y;
    for(double x=0; x<=10; x=x+dx){
        y=x;//y=x
        sum=sum+y*dx;//每一個矩形面積=高度*切割的寬度
    }
    cout<< sum;
    return 0;
}
```

P144

sin30等於0.5，那是你用背的，那電腦怎知到任何角度的sin(x)值呢，答案就是要靠級數。

P146

五、圓面積的剖析

圓面積大家都知道是 $r*r*\pi$ ，此公式已經有很多數學家用不同方法證明，本例則用電腦愚公移山的迴圈精神求解，我將一個半徑為 1 的圓，圓心放在(0,0),然後在 (-1,-1)，(1,1) 矩形之間，以 0.1 的距離畫線，將此矩形分成每塊 0.1*0.1 的正方形，那這一正方形便有 200*200 的小正方形，每一小正方形座標是(x,y)，面積為 0.01，若此小正方形與原點的距離小於等於 1，那此塊小正方形的集合便是屬於圓，我們累加這些滿足圓的小正方形個數再乘以 0.01，就可以得到圓面積，此演算法的 C++語言程式如下：

```
#include <iostream>
using namespace std;
int main() {
    double dx=0.1;
    int sum=0;
    for (double i=-1;i<=1;i=i+dx){
        for (double j=-1;j<=1;j=j+dx){
            if (i*i+j*j<=1)//每一個小正方形，都有座標(i,j)
                sum=sum+1;
        }
    }
    cout<< sum*0.01;
}
```

```

    }
}
cout<<sum*(dx*dx);//3.15
return 0;
}

```

同理，若將此矩形以 0.01 的大小切割，那精密度就會提高，請讀者自己將以上 dx 改為 0.01。

圓交集與聯集面積

兩個圓交集或聯集面積若要用幾何的方法求解，那就有點難了，但若將幾何問題數位化，用以上愚公移山的暴力法，方法就簡單了，以下是兩個圓面積的交集與聯集寫法。也就是檢查每一個小方塊是否同時在兩個圓內，就是交集面積。

```

#include <iostream>
using namespace std;
int main(int argc, char** argv) {
    int x1=18,y1=22,r1=7;//圓心(18,22),半徑 7
    int x2=17,y2=28,r2=8;//圓心(17,28),半徑 8
    double dx=0.1;
    int xmin=min(x1-r1,x2-r2);//找出左邊的 x
    int xmax=max(x1+r1,x2+r2);// 找出右邊的 x
    int ymin=min(y1-r1,y2-r2);// 找出下邊的 y
    int ymax=max(y1+r1,y2+r2);// 找出上邊的 y
    int a=0;
    for (double i=xmin;i<=xmax;i=i+dx){
        for (double j=ymin;j<=ymax;j=j+dx){
            if ( (((i-x1)*(i-x1)+(j-y1)*(j-y1))<=7*7)&& (((i-x2)*(i-x2)+(j-y2)*(j-y2))<=8*8))
                a=a+1;//交集個數
        }
    }
    cout<<a*dx*dx<<endl;//87.32 交集面積
    cout<<8*8*3.14+7*7*3.14-a*dx*dx;//267.5，聯集面積
    return 0;
}

```

多個圓的交集與聯集

以上只有兩個圓的交集，還可以直接比對每一小塊是否同時在兩個圓內，但若

是三個或四個圓的交集與聯集呢，或程式執行階段才知道圓的個數，那就只好先將每一個圓的小方塊以其座標編號，然後蒐集這些編號放在陣列，最後求這些陣列的交集或聯集，一樣可以得到交集或聯集面積，程式如下：

```
#include <iostream>
using namespace std;
int main() {
    int x1=18,y1=22,r1=7;//圓心(18,22),半徑 7
    int x2=17,y2=28,r2=8;//圓心(17,28),半徑 8
    int a[100000][2];
    int b[100000][2];
    //蒐集第 1 個圓的小正方形編號
    double dx=0.1;
    int s=0;
    for (double i=x1-r1;i<=x1+r1;i=i+dx){
        for (double j=y1-r1;j<=y1+r1;j=j+dx){
            if ((i-x1)*(i-x1)+(j-y1)*(j-y1)<=(r1*r1)){
                a[s][0]=(int)(10*i);
                a[s][1]=(int)(10*j);
                s=s+1;
            }
        }
    }
    int n1=s;
    cout<<"n1="<<n1<<endl;
    //蒐集第 2 個圓的小正方形編號
    s=0;
    for (double i=x2-r2;i<=x2+r2;i=i+dx){
        for (double j=y2-r2;j<=y2+r2;j=j+dx){
            if ((i-x2)*(i-x2)+(j-y2)*(j-y2)<=(r2*r2)){
                b[s][0]=(int)(10*i);
                b[s][1]=(int)(10*j);
                s=s+1;
            }
        }
    }
    int n2=s;
    cout<<"n2="<<n2<<endl;
    //求這些編號的交集，就是重疊的面積
```

```

int m=0;
int c[10000][2];
for (int i=0;i<=n1-1;i++){
    found=false;
    for (int j=0;j<=n2-1;j++){
        if (a[i][0]== b[j][0] && a[i][1]==b[j][1]){
            c[m][0]=b[j][0];
            c[m][1]=b[j][1];
            m++;
            break;
        }
    }
}

cout<<m*0.01<<endl;//86.44 交集面積
cout<<r1*r1*3.14+r2*r2*3.14-m*0.01;//268.38，聯集面積
return 0;
}

```

Python 有兩個集合的交集、聯集運算子，所以以上程式的 Python 程式如下：

```

def aa(x,y,r):
    a=[]
    dx=0.1
    s=10
    for i in range((x-r)*s,(x+r)*s):
        for j in range ((y-r)*s,(y+r)*s):
            if ((i-x*s)*(i-x*s)+(j-y*s)*(j-y*s))/(s*s) <=r*r:
                a.append(999*i+j*1.2)#蒐集起來
    return (a)
x1=18;y1=22;r1=7
x2=17;y2=28;r2=8
a1=aa(x1,y1,r1)
#a1=aa(1,3,2)
print(a1)
print(len(a1))
a2=aa(x2,y2,r2)
b1=set(a1)#將陣列轉為集合
print(len(b1))

```

```
b2=set(a2) #將陣列轉為集合
print(len(b2))
b=(b1 | b2)#求出兩個集合的聯集
print(len(b))
```

程式說明

```
append(999*i+j*1.2)
```

因為 Python 有集合的聯集與交集運算子，所以本題用 Python 較快，其次原來小方塊的座標是(x,y)，但是 Python 的集合僅能單一元素，所以我就將(x,y)轉為單一元素，為了避免座標的對稱問題轉為相同的元素,例如（18,19）與（19,18）很容易轉為相同的元素，所以多次試驗後將轉換公式調整為 $999*i+j*1.2$ 。

延伸實作

- 1、 以上僅為兩個圓的交集與聯集，請調整為圓的個數可為 2 到 5 個。例如，測試資料如下：

```
n=3//代表 3 筆測資
```

```
2,3,2 //代表(x,y),r
```

```
0,1,4
```

```
-2,4,3
```

```
n=4//代表 4 筆測資
```

```
2,3,2 //代表(x,y),r
```

```
0,1,4
```

```
-2,4,3
```

```
3,4,5
```

- 2、 請寫一程式，可以求得任意兩個矩形的交集面積。測試資料如下：

```
0,0,2,2 //左下角座標，右上角座標
```

```
-1,-1,1,1
```

- 3、 請寫一程式，可以求得任意矩形與圓的交集面積。測試資料如下：

```
0,0,2,2 //代表 3 筆測資
```

```
0,0,1 //代表 x,y,r)
```

三角形面積

以上僅為圓的交集與聯集，題目若改為幾何圖形還有三角形、或任意凸四邊形、或任意矩形呢？前面是使用(x,y)座標小於等於 $r*r$ ，判斷小正方形是否在圓內，那如何判斷任意座標是否在三角形內呢？我的方法是 S 點若在三角形 ABC 內，則

（SAB 面積+SBA 面積+SCA 面積）一定等於 ABC 的面積。

所以程式如下：

```
def aa(x1,y1,x2,y2,x3,y3):
    return abs(x1*y2+x2*y3+x3*y1-x2*y1-x3*y2-x1*y3)
dx=0.01
s=100
sum=0
x1=0;y1=0
x2=2;y2=0
x3=3;y3=3
a=aa(x1*s,y1*s,x2*s,y2*s,x3*s,y3*s)
print (a)
for i in range(0*s,3*s):
    for j in range (0*s,3*s):
        a1=aa(i,j,x1*s,y1*s,x2*s,y2*s)
        a2=aa(i,j,x1*s,y1*s,x3*s,y3*s)
        a3=aa(i,j,x2*s,y2*s,x3*s,y3*s)
        print(a1+a2+a3)
        if a==(a1+a2+a3) :
            sum=sum+1
    print()
print(sum*dx*dx)
```

以上是理想狀況，若是任意三角形，則 x 與 y 的要從三角形 3 點裡面取最小的與最大的當作檢驗範圍，所以程式如下：

```
def aa(x1,y1,x2,y2,x3,y3):
    return abs(x1*y2+x2*y3+x3*y1-x2*y1-x3*y2-x1*y3)
dx=0.01
s=100
sum=0
x1=0;y1=0
x2=2;y2=0
x3=3;y3=3
a=aa(x1*s,y1*s,x2*s,y2*s,x3*s,y3*s)
print (a)
for i in range(min(x1,x2,x3)*s,max(x1,x2,x3)*s):
    for j in range (min(y1,y2,y3)*s,max(y1,y2,y3)*s):
        a1=aa(i,j,x1*s,y1*s,x2*s,y2*s)
        a2=aa(i,j,x1*s,y1*s,x3*s,y3*s)
        a3=aa(i,j,x2*s,y2*s,x3*s,y3*s)
```

```

        print(a1+a2+a3)
    if a==(a1+a2+a3) :
        sum=sum+1
print()
print(sum*dx*dx)

```

以上公式可推廣到任意凸多邊形。以上是 Python 程式，C++程式自己做，才有樂趣。

三角形面積修正

因為三角形和圓形稍微不同，此時可提高精密度，也就是 dx 取 0.01 或 0.001，但這樣有點費時，經過觀察，三角形的小正方形(x,y)很容易落在線上，所以面積誤差較大，經多次試驗，其機率大約 1/3，所以面積修正如下：

```

def aa(x1,y1,x2,y2,x3,y3):
    return abs(x1*y2+x2*y3+x3*y1-x2*y1-x3*y2-x1*y3)
dx=0.1
s=10
sum=0
x1=0;y1=0
x2=3;y2=0
x3=3;y3=3
a=aa(x1*s,y1*s,x2*s,y2*s,x3*s,y3*s)
print (a/(s*s*2))
for i in range(0*s,3*s):
    for j in range (0*s,3*s):
        a1=aa(i,j,x1*s,y1*s,x2*s,y2*s)
        a2=aa(i,j,x1*s,y1*s,x3*s,y3*s)
        a3=aa(i,j,x2*s,y2*s,x3*s,y3*s)
        #print(a1+a2+a3,end=',')
        if a==(a1+a2+a3) :
            sum=sum+1
        if a1==0 or a2==0 or a3==0:
            sum=sum-0.33
    #print()
print(sum/(s*s))

```

以上是 Python 程式，C++程式自己做，才有樂趣。

延伸實作

1、請寫一個程式，可以求得任意三角形與任意圓形的交集面積。測試資料如

下：

0,0,3,0,3,3//代表三角形的 3 個點

0,0,2//代表 (x,y) ,r

2、請寫一個程式，可以求的任意矩形與圓的交集面積。測試資料如下：

0,0,3,0,3,3,0,3//代表矩形的四個點

0,0,2//代表 (x,y) ,r

3、請寫一個程式，可以求的任意四邊形與圓的交集面積。測試資料如下：

0,0,3,3,3,0,0,-3 //代表 4 邊形的四個點

0,0,2 //代表 (x,y) ,r

(本專題靈感來自高雄市飆程式網 <http://khcode.m-school.tw/> 『不定數量圓面積的交集與聯集』)

6_5 APCS 試題

一、完全奇數 (107/10試題)

13311,13199 稱為完全奇數，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13256，則M=13311；P=13199，而

$13311 - 13256 = 55$, $13256 - 13199 = 57$ ，所以輸出55。（此與106年第1次試題類似，

以下的遞增數也類似，所以一定要做考古題）

```
#include <iostream>
```

```
#include <cmath>
```

```
using namespace std;
```

```
int judge(int n){
```

```
    int t;
```

```
    //將數字分解成個位數，十位數...
```

```
do{

    t=n %10;

    if (t%2==0)

        return(0);

    //找到偶數就離開

    n=n/10;

}while (n>0);

//通過層層考驗，就是完全奇數

return(1);

}

int main() {

    int n=13256;

    int m=n;

    int p=n;

    int odd;//bool

    //found big

    odd=judge(m);

    while(odd==0) {//要先判斷，避免給的數字是完全奇數

        m=m+1;
```

```

        odd=judge(m);

    }

    //cout<<"m="<<m<<endl;

    //found small

    odd=judge(p);

    while(odd==0) {

        p=p-1;

        odd=judge(p);

    }

    //cout<<"p="<<p<<endl;

    cout<<min(abs(n-m),abs(n-p));

    return 0;

}

```

實作

1. 請寫一程式，可以判斷所輸入數字（可任意長度）是否完全奇數。
2. 請寫一程式，可以判斷所輸入數字（可任意長度）是否完全遞增。例如，
12345 稱為完全遞增，12344,5433 等，就不是。
3. 請寫一程式，可以計算所輸入數字（可任意長度）的奇位數和。例如，
12345 的奇位數和是9。

4. 請寫一程式，可以輸入一個任意長度的長整數，並求其左邊 n 個數字與右邊 n 個數字的和。例如，輸入1234567,3，答案是123+567=690。

5. 請問1 到10000 的整數共有幾個2。例如，122 有兩個2。

6. 請問1 到10000 的整數共有幾個38。例如，13838 有兩個38。

7. 請問1 到10000 的整數共有幾個168。例如，91681 有1 個168。

8. 請問1 到10000 的整數共有幾個完全奇數。例如375 是完全奇數，376 就不是。

9. **遞增數 (APCS107/03試題)**。12 稱為遞增數，11 和21 就不是。請輸入一個數 n ，並求1.. n 共有幾個遞增數。例如，輸入25，求1 到25 的遞增數，答案是20。

二、城市移動 (APCS 10803試題)

假設有一個遊戲，有3 個城市，玩家依亂數 L 在3 個城市移動，移到城市1 得 L 分，如果 L 是偶數則下一回合移到城市2，否則留在城市1；移到城市2 得到 $2L$ 分，如果 L 是3 的倍數，下一回合移到城市3，否則前往城市1；移到城市3 得分是 L 除以10 的商，如果 L 是5 的倍數，下一回合移到城市3，否則移到城市1。測試資料如下：

4 1 //玩4回合，起始城市是1

1 2 3 12 //四個亂數

備註：此題目為憑個人記憶，題目很好，還是呼籲主辦單位公佈試題，讓學生

練習。

演算法：因為每一城市有其不同移動規則與得分方式，所以可用（ switch ）分成3 部分，且因每一城市都有其移動規則，所以每一城市都要自己用1 個if else，這樣就可以計算每一次轉移到哪一城市與每一次的得分。所以這題是屬於雙層決策問題，拿到分數沒有問題。

執行結果

G502

程式列印

```
#include <iostream>

using namespace std;

int main()

{

    int a,n,i;

    cin>> n>>a;

    //n=4;a=1;

    int i;

    int b;//得分

    int bt=0;//總得分

    for (i=1;i<=n;i++){//迴圈，重複的次數
```

```
cin>>l;
```

```
switch(a){
```

```
    case 1:
```

```
        b=l;
```

```
        if (l %2==0)
```

```
            a=2;
```

```
        else
```

```
            a=1;
```

```
        break;
```

```
    case 2:
```

```
        b=2*l;
```

```
        if(l%3==0)
```

```
            a=3;
```

```
        else
```

```
            a=1;
```

```
        break;
```

```
    case 3:
```

```
        b=l/10;
```

```
        if (l%5==0)
```

```

        a=3;

    else

        a=1;

    break;

}

bt=bt+b;

cout<<"a="<<a<<" b="<<b<<" bt="<<bt<<endl;

}

cout<<"bt="<<bt;

return 0;

}

```

三、數位編碼（APCS107試題）

1、輸入。

第一列為取碼長度 a。例如，

1

第二列輸入若干數字，以 0 結束。例如，

252 373 28 0

2、處理

將第二列數字從左邊取長度 a 的數字，取極大值輸出。

將第二列數字從右邊取長度 a 的數字，取極小值輸出。

3、範例

資料編號	輸入	輸出
1	1 (取碼長度) 252 373 28 0	2 3 2 max=3 2 3 8 min=2 3 2
2	2 (取碼長度) 258 0	25 max=25 58 min=58 25 58
3	2 (取碼長度) 123 58 612 0	12 58 61 max=61 23 58 12 min=12 61 12

程式列印

```
#include <iostream>
#include <cmath>
using namespace std;
int main() {
    int a;//取碼長度
    int b;
    int c;
    int t;
    cin>>a;
    int min=99999999;
    int max=0;
    while (cin>>b && b!=0){
        c=b;
        //取左邊
        while (b>=pow(10,a)) {
            b=b /10;
        }
        if (b>max)
            max=b;
        //取右邊
        t=pow(10,a);
        c=c % t;
        if (c<min)
            min=c;
    }
}
```

```

    cout<<max<<" "<<min<<endl;
    return 0;
}

```

p150 新增

23	<pre> int main() { int s=0; for (int i=1 ;i<=10;i++){ s=s+i; i++; } cout <<s; } </pre>	
24	請問以下程式有解否？ <pre> int main() { int s=0; for (int i=1; i!=10;i=i+2){ s=s+i; } cout <<s; return 0; } </pre>	
25	<pre> #include <iostream> using namespace std; int main() { int s=0,i; for (i=1;s<=20;i++){ s=s+i; } cout<<i<<endl; cout<<s<<endl; return 0; } </pre>	
26	<pre> #include <iostream> </pre>	

	<pre> using namespace std; int main() { int s=0,i; for (i=1;s<=60 && i<10;i++){ s=s+i; } cout<<i<<endl; cout<<s<<endl; return 0; } </pre>	
27	<pre> #include <iostream> using namespace std; int main() { for (int i=0 ;i<=5;i++){ for (int j=0;j<=i;j++){ printf("%-6d",j);//left } printf("\n"); } printf("\n"); for (int i=0 ;i<=5;i++){ for (int j=0;j<=i;j++){ printf("%6d",j);//right } printf("\n"); } return 0; } </pre>	

P154

10、請寫一個程式，可以計算所指派的整數，共有多少組連續的 1。例如，1011001111 共有 4 組連續的 1。

P156

前面處理少量的資料，可以個別設定一個變數，但是如果資料龐大，例如，若

有資料如下：

12,88,34,45,18,99,22,11,48,77,66

要求極大、極小、平均等運算，若還是使用 11 個變數儲存，那程式將會很冗長，本章將會介紹另一種資料結構『陣列』，然後配合上一章的 for 或 while 迴圈，就可以以非常精簡的程式，處理以上大批資料。其次，陣列還有維度之分，分別是一維、二維等等，那就要看您資料運算的方式，若只有同批資料的關係，例如，上例就是一維資料陣列就可以。但是，若一個班級所有學生的國、英、數、自然、社會等 5 科成績，我們不僅要求縱向的各科平均，還要求橫向的每人平均，那就需要使用二維陣列。請看本章以下說明。

p158

還有，C/C++陣列的宣告只是參考，它不會幫您檢查邊界，索引超界也可使用。例如，

```
int a[]={0,1,2,3};
cout <<a[4]<<endl;//無法預期
a[4]=4;
cout <<a[4]<<endl;//4，竟然可用
a[5]=5;
cout <<a[5]<<endl;//5
```

P158_1

本例資料個數多，應該以一維陣列儲存，再用迴圈輸出、與統計其和，這樣程式寫來才會短。

P161

2、請寫程式完成以下條件。

(1) 產生20 個1 到6 的亂數。(2) 以陣列儲存以上資料。(3) 輸出以上資料。(4) 計算並輸出共有多少個3。(5)輸出1到6各幾個。

3. 請寫程式完成以下條件。(1) 產生50 個1 到100 的亂數。(2) 以陣列儲存以上資料。(3) 輸出以上資料。(4) 統計與輸出0~9,10~19,20~29,30~39,40~49,50~59,60~69,70~79,80~89,90~100 等

區間的人數。

P163

- 1、 假設有一個大富翁遊戲，共有 10 個位置，每個位置的獎罰如下表，遊戲一開始假設有 10000 元，請寫程式讓使用者玩，本例假設使用者每按一下任意鍵，電腦就丟一顆骰子，決定移動的步數。

0	1	2	3	4	5	6	7	8	9
中發 票得 200 元	闖紅 燈罰 200 元	中樂 透得 10000 元	任意 停 車， 罰 1000 元	繳汽 車稅 2000 元	繳房 屋稅 3000 元	停 玩 一 次	得到 股息 1000 元	抽獎 得到 3000 元	繳 停 車 費 30 元

- 2、 假設 2 月永遠是 28 天，請寫一程式，可以輸入一個日期，且轉為今年第幾天。例如，輸入 2,1 代表 2 月 1 日，那就是第 32 天，輸出 32。

- 3、 同上題，但考慮閏年，例如，輸入 2000,3,1 則輸出 61。

- 4、 同上題，可以輸入兩個日期，計算其相差的天數。例如，輸入

2018 9 30

2018 10 1

則輸出 1。(此為國內高中生程式解題系統(<https://zerojudge.tw/>)基礎題。

- 5、 同上題，但可比較兩個日期的先後。例如，應該還書日期是 2018 9 30，但是 2018 10 1 來還，表示逾期。

- 6、 寫一個程式，可以判斷所輸入日期，是否在指定日期的中間，例如，指定日期是 2022 3 15 到 2022 6 3，其次輸入日期是 2022 3 18 或 2022 8 12 等，判斷是否在指定日期內。

範例 7_1e 可以輸入天數，但轉為日期輸出，例如輸入 32，能得到 2 月 1 日。

(本例先簡化題目，假設 2 月通通 28 天)

執行結果

G706

演算法

- 1、 將每個月的天數以陣列儲存。

- 2、 從 1 月開始，只要天數大於每月天數，就扣掉，所以程式如下：

程式列印

```
#include <iostream>
```

```
using namespace std;
int main() {
    int a[]={0,31,28,31,30,31,30,31,31,30,31,30,31};
    int d=365;
    int m=1;
    int d1=d;
    while (d>a[m]){
        d=d-a[m];
        m++;
    }
    cout<<d1<<" is "<<m<<"/"<<d;
    return 0;
}
```

自我練習

- 1、同上範例，但考慮閏年。
- 2、假設 2020/1/1 是星期三，請寫一程式，可以輸入 2020 年任一日期，而得到該日星期幾。

範例 7_1f KTV 音樂伴奏與字幕。我們已經於範例 2_3b 作出音樂伴奏，但卻是一個音一個指令，程式非常冗長，本範例資料結構改用陣列，那就可以分別儲存音階、音長與對應歌詞。（本例同樣以小毛驢為例）

資料結構

本例一分鐘80 拍，所以1 拍的時間是60000ms/80，以一個四分之一音符為1

拍，但實際上每小節是2 拍，所以一個四分之一音符所佔的時間是

60000ms/160。我就以陣列兩兩一組儲存每一音符的音高與拍數如下（以下僅作

4 小節，『我有一隻小毛驢我從來也不騎』）：

```
const int a[]={1,1,1,1,1,1,3,1,5,1,5,1,5,1,5,1,6,1,6,1,6,1,8,1,5,4};
```

然後每一個音高就剩查表了，如以下陣列。

```
const int b[]={0,523,587,659,698,784,880,988,1046};
```

程式列印

```
#include <iostream>
using namespace std;
```

```

int main() {
    const int b[]={0,523,587,659,698,784,880,988,1046};
    //小毛驢
    const int s=80;
    const int a[]={1,1,1,1,1,3,1,5,1,5,1,5,1,5,1,6,1,6,1,6,1,8,1,5,4};
    const int l=26;
    //蝴蝶
    //const int
a[]={1,2,1,2,2,1,3,2,3,2,2,1,1,1,2,1,3,1,1,2,3,2,3,1,4,1,5,2,5,2,4,1,3,1,4,1,5,1,3,2};
    //const int l=40;
    while(1){
        for (int i=0;i<l/2;i=i+1){
            _beep(b[a[2*i]],a[2*i+1]*60000/(s*2));
        }
    }
    return 0;
}

```

實作

- 1、 同上範例，那如何顯示歌詞呢？
- 2、 請自己選一首歌，完成以上工作。
- 3、 請將全班的歌蒐集起來，合併在同一程式，那如何點歌呢？

P164

請鍵入以下程式，觀察執行結果：

```

#include <iostream>
using namespace std;
int main() {
    int a[3][5]={ {10, 11, 12,13, 14},
                  {20, 21, 22, 23, 24},
                  {30, 31, 32, 33, 34} };
    cout<<a[0][0]<<endl;
    cout<<a[2][4]<<endl;
    for (int i=0;i<3;i++){
        for (int j=0;j<4;j++){
            cout<<a[i][j]<<"  ";
        }
        cout<<endl;
    }
}

```

```

    }
    return 0;
}

```

P165

本例不但要計算橫向的個人成績，也要計算縱向的各科成績，所以應該以二維陣列儲存，程式才能精簡，本例程式如下：

P168

本例要同時處理縱向與橫向資料，所以應該以二維陣列儲存，程式才能精簡，本例程式如下：

P170

以下字串是使用C語言表示：

```

#include <iostream>
#include <stdlib.h> //使用rand()
#include <time.h> //使用 time()
using namespace std;
int main(int argc, char** argv) {
    int p,c;
    //字串陣列
    char a[3][3][10]= {" 平手", " 人贏", " 電腦贏"},
                                {" 電腦贏", " 平手", " 人贏"},
                                {"人贏", " 電腦贏", " 平手"};
    char b[3][10]={"剪刀","石頭","布"}    ;
    //需要宣告#include <time.h>
    srand(time(NULL)); //隨機選取亂數起點， 若未使用此指令，每次亂數起點
    都相同
    cout<< "input 0(剪刀) 1(石頭) 2(布) :";
    cin >>p;
    c=rand()%3; //產生0，1，2的亂數，需要#include <stdlib.h>
    cout <<"you          take:"<<b[p]<<endl;
    cout <<"computer take:"<<b[c]<<endl;
    cout <<"Result          : "<<a[c][p]<<endl;
    return 0;
}

```

以下字串使用C++語言表示：

```
#include <iostream>
#include <stdlib.h> //使用rand()
#include <time.h> //使用 time()
using namespace std;
int main() {
    int p,c;
    //字串陣列
    string a[3][3]= {" 平手", " 人贏", " 電腦贏"},
                  {" 電腦贏", " 平手", " 人贏"},
                  {"人贏", " 電腦贏", " 平手"};
    string b[3]={"剪刀","石頭","布"} ;
    //需要宣告#include <time.h>
    srand(time(NULL)); //隨機選取亂數起點， 若未使用此指令，每次亂數起點
    都相同
    cout<< "input 0(剪刀) 1(石頭) 2(布) :";
    cin >>p;
    c=rand()%3; //產生0，1，2的亂數，需要#include <stdlib.h>
    cout <<"you      take:"<<b[p]<<endl;
    cout <<"computer take:"<<b[c]<<endl;
    cout <<"Result      : "<<a[c][p]<<endl;
    return 0;
}
```

P172

```
struct stuGrade
{
    string name; //C 請用 char *name;
    int chi;
    int math;
    int eng;
};
```

P174

```
struct stuGrade
{
    string name ;// C 請改為 char *name;
```

```

    int score[3];
};

P175
#include <iostream>
using namespace std;
//struct stdGrade//C的字串用法
//{
//    //int SID;
//    //char *name;//C的字串用法
//    //int grade[4];
//    //char *memo;
//};
struct stdGrade
{
    int SID;
    string name;//C++字串用法
    int grade[4];
    string memo;
};
int main(int argc, char** argv) {
    struct stdGrade students[4];
    struct stdGrade temp;
    int i,j, sum;

    students[0].SID=1;
    students[0].name="aa";
    students[0].grade[0]=80;
    students[0].grade[1]=90;
    students[0].grade[2]=70;

    students[1].SID=2;
    students[1].name="bb";
    students[1].grade[0]=20;
    students[1].grade[1]=10;
    students[1].grade[2]=80;

    students[2].SID=3;

```

```

students[2].name="cc";
students[2].grade[0]=90;
students[2].grade[1]=80;
students[2].grade[2]=70;
for(i=0;i<3; i++)
{
    sum=0;
    for(j=0; j<3; j++)
        sum+=students[i].grade[j];
    students[i].grade[3]=(int)sum/3;
    if(students[i].grade[3]<60)
        students[i].memo="Fail";
    else
        students[i].memo="";
}
for(i=0; i<=3; i++)
{
    sum=0;
    for(j=0; j<3; j++)
        sum+=students[j].grade[i];
    students[3].grade[i]=(int)sum/3;
    students[3].SID=0;
    students[3].name="";
    students[3].memo="";
}
cout<<"\n";
cout<<" No    name    chi  eng    math    avg    memo\n";
for(i=0; i<4; i++)
{
    cout.width(6);cout<< students[i].SID;
    cout.width(6);cout<< students[i].name;
    cout.width(6);cout<< students[i].grade[0];
    cout.width(6);cout<< students[i].grade[1];
    cout.width(6);cout<< students[i].grade[2];
    cout.width(6);cout<< students[i].grade[3];
    cout.width(6);cout<< students[i].memo;
    cout<<endl;
}

```

```
    return 0;
}
```

P179

以上是索引從 1 開始，這樣比較好理解。但電腦的運算思維，索引經常是從 0 開始，以下則是索引從 0 開始的程式，請要熟記，這樣考試時，解題速度才快。

```
#include <iostream>
using namespace std;
int main() {
    int a[]={ 99, 88,77, 66, 55};
    int i, j, t;
    const int N=5;
    //排序
    for(i=0; i<N-1; i++)
        for(j=0; j<N-i-1; j++)
            if(a[j]>a[j+1])
            {
                t=a[j];
                a[j]=a[j+1];
                a[j+1]=t;
            }
    //輸出
    cout<<"\nAfter sort:  \n";
    for(i=0; i<N; i++)
        cout<< a[i]<<" ";
    return 0;
}
```

排序在很多場合會用到，以下是直接套用 sort()函式，sort()函式語法如下：

sort(a+排序開始索引, a+排序個數)

但使用前應匯入#include <algorithm>，所以本例程式如下：

```
#include <iostream>
#include <algorithm> //使用sort
using namespace std;
int main() {
    int a[]={ 99, 88,77, 66, 55};
    const int N=5;
```



```

//排序
sort(a,a+N);//include <algorithm>
//輸出
for(int i=0; i<N; i++)
    cout<< a[i]<<" ";
return 0;
}

```

P182

```

#include <iostream>
using namespace std;
int main() {
    int a[]={77, 66, 99, 44, 55};
    const int N=5;
    int b[N];
    int i, j, t;
    cout<<"\ngrade: \n";
    for(i=0; i<N; i++) {
        cout.width(5);
        cout<< a[i]<<" ";
        b[i]=1;
    }
    /* sort */
    for(i=0; i<N-1; i++){
        for(j=i+1; j<N; j++){
            if(a[i]>a[j])
                b[j]++;
            else
                b[i]++;
        }
    }
    cout<<"\nrank: \n";
    for(i=0; i<N; i++){
        cout.width(5);
        cout<< b[i]<<" ";
    }
    return 0;
}

```

P184

實作

1、密碼解碼（APCS107 試題）

（1）、定義密碼如下：

0101	0111	0010	1101	1000	1100
A	B	C	D	E	F

（2）、輸入、輸出如下：

輸入	1 0101	2 0010 1101	4 1000 1100 0101 0111	備註 輸入的值只有以上 6 種
輸出	A	CD	EFAB	

程式列印

這是很典型的線性搜尋題目，找到以後，將其結果累加起來，所以程式如下：

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    string a[6][2]={ {"0101","A"},  
                      {"0111","B"},  
                      {"0010","C"},  
                      {"1101","D"},  
                      {"1000","E"},  
                      {"1100","F"}};
```

```
    int n;
```

```
    int i,j;
```

```
    string b[100];
```

```
    string c="";//記得要初始化
```

```
    cin>>n;//數量
```

```
    for (i=0 ;i<=n-1;i++){
```

```
        cin>>b[i];
```

```
    }
```

```
    //輸出看看,交卷前去掉
```

```
    for (i=0 ;i<=n-1;i++){
```

```

        cout<<b[i]<<endl;
    }
    //逐一搜尋
    for (i=0 ;i<=n-1;i++){
        for (j=0 ;j<=5;j++){
            if ((b[i]==a[j][0])){
                c=c+a[j][1];
            }
        }
    }
    cout<<c<<endl;
    return 0;
}

```

2、同上題，但輸入『CD』得到『00101101』。

3、同上範例，但輸入『2C11D』得到『001000101101 共 11 個 1101』。

P187

2、以下是字串用C++風格的程式列印：

```

#include <iostream>
using namespace std;
int main() {
    string a[13][2]={{"0", "0"},
                     {"aa", "1111168"},
                     {"hh", "2222168"},
                     {"cc", "3333168"},
                     {"gg", "4444168"},
                     {"ff", "5555168"},
                     {"ii", "6666168"},
                     {"ee", "7777168"},
                     {"bb", "8888168"},
                     {"jj", "9999168"},
                     {"dd", "1688168"},
                     {"kk", "2688168"},
                     {"ll", "3688168"}};

    int i, j, l, u, m;
    string name, t;
}

```

```

bool found=false;
for(i=1; i<=11; i++){
    for(j=1; j<=11; j++)    {
        if(a[j][0]> a[j+1][0]){//有比較運算子>
            t=a[j][0];a[j][0]=a[j+1][0];a[j+1][0]=t;
            t=a[j][1];a[j][1]=a[j+1][1];a[j+1][1]=t;
        }
    }
}
for(i=1; i<=12; i++)
    cout <<a[i][0]<<" "<<a[i][1]<<endl;
cout<<"Input a name to search:";
cin>> name;
l=1;//下界
u=13;//上界
while((l<=u) & (found==0))    {
    m=(int)((l+u)/2);
    if(name== a[m][0]) {
        cout<<"the phone number is : "<< a[m][1];
        found=true;
        break;
    }
    else {
        if(name> a[m][0])
            l=m+1;//調整下界
        else
            u=m-1;//調整上界
    }
}
if(!found)
    cout<<"no data found";
return 0;
}

```

P193

以下程式索引從0開始，所以在寫範圍時，就用『for (i=0;i<n;i++)』，這樣就不用『for (i=0;i<=n-1;i++)』。

```

#include <iostream>
using namespace std;
int main() {
    int m,n,p,i,j,k;
    n = 2;    m = 3;    p = 2;
    int a[2][3]={ {-1, 0, 3}, {2, 1, -2}};
    int b[3][2]= {{1,0}, {2,-3}, {-1,4}};
    int c[2][2];
    for (i=0;i<n;i++){
        for (j=0;j<m;j++){
            cout.width(3);
            cout<<a[i][j];
        }
        cout<<endl;
    }
    cout<<endl;
    for (i=0;i<m;i++){
        for (j=0;j<p;j++){
            cout.width(3);
            cout<<b[i][j];
        }
        cout<<endl;
    }
    cout<<endl;
    //矩陣相乘
    for (i=0;i<n;i++){
        for (j=0;j<p;j++){
            c[i][j]=0;
            for (k=0;k<m;k++){
                c[i][j]=c[i][j]+a[i][k]*b[k][j];
            }
        }
    }
    for (i=0;i<n;i++){
        for (j=0;j<p;j++){
            cout.width(3);
            cout<<c[i][j];
        }
    }
}

```

```

        cout<<endl;
    }
    cout<<endl;
    return 0;
}

```

P196

實作

1、同上範例，但是題目以亂數出現。

P198

實作

1、同上範例，但 4 個選項順序，可隨機亂數排列。但請留意要把答案記起來。

P199

2. 由左到右，每兩個數字分為一組。

3. 找出 a 。從9,8,7 的平方，依序找出小於等於最左邊那一組的數字，本例是

13，9,8,7,6,5,4 的平方都太大，所以找到3，並扣掉3 的平方9，剩下4。

4. 使用迴圈，從第二組數字開始逐一找 $b_1b_2b_3\cdots$ 。

(1) 計算每一次的餘數。餘數 $d = \text{前面餘數} \times 100 + \text{這一組數字}$ 。

(2) 用迴圈找 b 。 b 從9,8,7 找 $b(20a + b)$ 小於等於餘數 d 。本例9,8 都太大，找到

7。

(3) 餘數 d 扣掉 $b(20a + b)$ 。 $d = d - b(20a + b)$ 。

(4) 把得到的37 看成 a ，繼續用 $b(20a+b)$ 找下一位數的 b 。

5. 輸出 $ab_1b_2b_3\cdots$ ，即為所求。

7_5 字元陣列與字串

C 語言沒有字串型態，僅能使用字元陣列或指標，因為 APCS 的學科採用 C 語言，所以本書還是介紹 C 語言如何處理字串，這樣考試的時候才能看得懂題目。

C 字串宣告與設定初值

在C 語言中，基本資料型態並無字串資料型態（C++ 才有字串資料型態）。例如：無法使用

```
string a;
```

若要處理字串，則要使用字元陣列或指標，指標請參看8-3 節。在C 語言中，字串是利用簡單的一維字元陣列加上一個字串結束字元'\0'（NULL）來表示。例如：以下列敘述是宣告並初始化變數a 的初值為字串"abc"。其次，請注意字串使用雙引號（" "），字元使用單引號（' '），如果誤用則會造成錯誤。

```
char a[ ]="abc";
```

上式的效果等於

```
char b[ ]={'a','b','c','\0'}; /* 將字元單獨放入字元陣列中 */
```

雖然第一種字串宣告方式中的字串"abc" 的長度為3，但是C/C++ 語言編譯時會自動在字串末端加上一個結束字元（'\0'）表示字串的結束，因此，所宣告的a 字元陣列在記憶體中實際上是占了4 個字元組。請鍵入以下程式，並觀察執行結果。

```
#include <iostream>
```

```
#include <string.h> //使用了strcpy函式，請看9_3字串函式
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    char a[ ]="abc"; //會自動放\0
```

```
    char b[ ]={'a','b','c','\0'}; //不會自動放'\0'
```

```
    char c[50];
```

```
    cout <<a[0]<<endl; //a
```

```
    cout<<a[3]<<endl;
```

```
    cout<<a<<endl; //abc
```

```
    cout<<b<<endl; //abc
```

```
    //字串的複製
```

```

//c=a;//不行，要用strcpy函式
strcpy(c,a);// #include <string.h>
cout<<c<<endl;//abc
char d;
//計算長度，人工複製，本例使用字串結束字元\0
int n=0;int i=0;
do {
    d=a[i];
    c[i]=d;
    i++;
}
while (d!='\0');
cout<<c;//abc
n=i-1;
cout<<"the length of a is: "<<n;//3
return 0;
}

```

字元陣列（字串）的複製

C 字元陣列的複製，並不能使用

```
c=a;
```

一定要使用字元陣列複製函式strcpy，請參考9-3 節。

```
strcpy(c,a);
```

字串輸入與輸出

字串的輸出和其他資料型態一樣都可以使用printf()、scanf() 與cin、cout物件，唯須注意下列事項：

1. 列印格式需要置換為%s。
2. 使用scanf()時必須使用取址符號(&)，但輸出時就不用再取址了，程式如下：

```

char a[4];
scanf("%s", &a);
printf("%s", a);

```

3. 使用cin 物件輸入、輸出如下：（C++ 才可以）

```

char a[10];
cin>>a;
cout<<a<<endl;

```

4. 若所輸入的字串中包含空白字元，則必須以讀入字元（getchar()）的方式將字串輸入到陣列中，並在陣列末端加入一結束字元'\0'；若仍以scanf()或cin方式

輸入，則資料項的輸入會在遇到第一個空白字元時結束。請比較下列二程式片段：

```
char b[10];
scanf("%s", &b); // 本例假設輸入b="abc xyz"
printf("%s", b);
cout<<endl;cout
cout<<"Please input a string:";
cin >>a; // 本例假設輸入b="abc xyz"
cout<<a;
```

兩者均輸出結果為"abc"，顯然有誤。以下程式使用getchar() 函式實現。

```
#include <iostream>
#include <stdio.h> //使用了getchar函式
using namespace std;
int main()
{
    char c, d[10]; int i=0;
    while ((c=getchar( )) != EOF && c!='\n'){
        /* 本例假設輸入d="abc xyz" */
        d[i]=c;
        i++;
    }
    d[i]='\0';
    printf("%s\n", d);
    cout<<d<<endl;
    return 0;
}
```

則輸出結果即為輸入的字串"abc xyz"。若是C++，還可使用cin.getline函式，其語法如下：

```
cin.getline(字串名稱,最大字串長度[,結束字元]);
結束字元編譯器會預設"\0"，所以以下程式，功能同上。
#include <iostream>
using namespace std;
int main(){
    char d[10]; //C style
    cout<<"請輸入字串（可含空白）：";
    cin.getline(d,10);
    cout<<d<<endl;
    return 0;
```

```
}
```

以上是C style字串，若是C++ style，則程式如下：

```
#include <iostream>
using namespace std;
int main(){
    string d;
    cout<<"請輸入字串（可含空白）："；
    getline(cin,d);
    cout<<d<<endl;
    return 0;
}
```

字串陣列

若有連續字串如下：

Hand、Mary、洪國勝、陳惠敏、王一平

那該如何儲存呢？答案是使用字元二維陣列，請鍵入以下程式，並觀察執行結果。

```
#include <iostream>
#include <string.h> //使用strcmp, strcpy
using namespace std;
int main()
{
    char a[][10]={"Hand","Mary","洪國勝","陳惠敏","王一平"};
    cout<<a[0][0]<<endl; //H
    cout<<a[0]<<endl; //Hank
    cout<<a[2][0]<<a[2][1]<<endl; //洪, 中文佔兩個字元
    cout<<a[2]<<endl; //洪國勝
    cout<<endl;
    int i,j;
    int n=5;
    // 輸出全部
    for (i=0;i<=n-1;i++){
        cout<<a[i]<<endl;
    }
    cout<<endl;
    //排序, 由小而大，英文按字母，中文按筆畫的多寡
    char t[10];
    for(i=0;i<=n-1-1;i++){
```

```

        for (j=0;j<=n-i-1-1;j++){
            if (strcmp(a[j],a[j+1])>0){//字串比較，請看9_3
                strcpy(t,a[j]);//字串複製，請看9_3
                strcpy(a[j],a[j+1]);
                strcpy(a[j+1],t);
            }
        }
    }
    for (i=0;i<=n-1;i++){
        cout<<a[i]<<endl;
    }
    return 0;
}

```

程式說明

1、字串宣告時，僅可省略最高維度的大小。本例預設字串長度最多為10，程式如下：

```
char a[][10]={"Hand","Mary","洪國勝","陳惠敏","王一平"};
```

4、以上程式，若使用C++的string型態，則程式如下：

```

#include <iostream>
#include <cstring>
using namespace std;
int main() {
    string a[]={"Hand","Mary","洪國勝","陳惠敏","王一平"};;
    cout<<a[0][0]<<endl;//H
    cout<<a[0][1]<<endl;//a
    cout<<a[1]<<endl;//Mary
    int n=5;
    string t;
    int i,j;
    for(i=0;i<=n-1-1;i++){
        for (j=0;j<=n-2-i;j++){
            if (a[j]>a[j+1]){//字串比較
                t=a[j];//字串複製
                a[j]=a[j+1];
                a[j+1]=t;
            }
        }
    }
}

```

```

    for (i=0;i<=n-1;i++)
        cout<<a[i]<<endl;
    return 0;
}

```

實作

1、請寫一個程式，可以輸入若干個人名，輸入0時，結束輸入，輸出這些人名，且由大而小輸出這些人名。

二維字串陣列

若有二維資料如下：

姓名	數學	APCS	平均
Tom	90	80	
John	60	90	
Mary	80	60	

請問如何儲存，如何計算平均。請鍵入以下程式，並觀察執行結果。

```

#include <iostream>
#include <cstdlib>//使用atoi函式，請看9_3
using namespace std;
int main()
{
    char a[][4][10]={ {"Tom","90","80",""},
                      {"John","60","90",""},
                      {"Mary","80","60",""} };

    cout<<a[0][0][0]<<endl;//T
    cout<<a[0][0][1]<<endl;//o
    cout<<a[0][0]<<endl;//Tom
    cout<<a[0][1]<<endl;//90
    cout<<a[1][0]<<endl;//John
    int i,j;
    //輸出
    for (i=0;i<=2;i++){
        for (j=0;j<=3;j++){
            cout<<a[i][j]<<" ";
        }
        cout<<endl;
    }
    //計算總分、平均
    int sum=0,avg;

```

```

for (i=0;i<=2;i++){
    sum=0;
    for (j=1;j<=2;j++){
        sum=sum+atoi(a[i][j]);//轉數值，請看9_3
    }
    avg=sum/2;
    itoa(avg,a[i][3],10);//轉字串，請看9_3
}
cout<<endl;
//輸出
for (i=0;i<=2;i++){
    for (j=0;j<=3;j++){
        cout<<a[i][j]<<" ";
    }
    cout<<endl;
}
return 0;
}

```

以下則是C++程式列印：

```

#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    string a[][4]={ {"Tom","90","80",""},
                    {"John","60","90",""},
                    {"Mary","80","60",""} };

    cout<<a[0][0][0]<<endl;//T
    cout<<a[0][0][1]<<endl;//o
    cout<<a[1][1][0]<<endl;//6
    cout<<a[0][0]<<endl;//Tom
    cout<<a[0][1]<<endl;//90
    cout<<a[1][0]<<endl;//John
    int i,j;
    //輸出
    for (i=0;i<=2;i++){
        for (j=0;j<=3;j++){
            cout<<a[i][j]<<" ";

```

```

    }
    cout<<endl;
}
//計算總分、平均
int sum=0,avg;
for (i=0;i<=2;i++){
    sum=0;
    for (j=1;j<=2;j++){
        sum=sum+stoi(a[i][j]);//轉數值，請看9_3
        //sum=sum+atoi(a[i][j]);//C style轉數值，請看9_3
    }
    avg=sum/2;
    a[i][3]=to_string(avg);//轉字串，請看9_3
    //itoa(avg,a[i][3],10);//C style轉字串，請看9_3
}
cout<<endl;
//輸出
for (i=0;i<=2;i++){
    for (j=0;j<=3;j++){
        cout<<a[i][j]<<" ";
    }
    cout<<endl;
}
return 0;
}

```

實作

- 1、請寫一程式，可以輸入以上資料。
- 2、同上題，可計算每人不及格科數、每科平均、每科不及格人數、依照平均排序。
- 3、同上題，可計算每個人的名次，輸出按照原來的順序排列。
- 4、同上題，可輸入姓名，而查得其平均。
- 5、鍵入以下程式，並觀察、寫出執行結果。

```

#include <iostream>
using namespace std;
int main()
{
    char a[]="ABCD";
    char b[][10]={"AA1","BB2","CC3","DD4"};

```

```

char c[][2][10]={{"AA","90"},{"BB","80"},{"CC","70"}};
cout <<a[0]<<endl;//A
cout <<a<<endl;//ABCD
cout <<b[0][0]<<endl;//A
cout <<b[0]<<endl;//AA1
cout <<c[0][0][0]<<endl;//A
cout<<c[0][0]<<endl;//AA
cout<<c[0][1]<<endl;//90
return 0;
}

```

p204

一、填充題

題號	題目	結果
1	<pre> int main() { int a[7][7]; for (int i=1;i<=6;i++) for(int j=1;j<=6;j++) if (i==j j<i) a[i][j]=1; else a[i][j]=i+j; cout<<a[3][4]<<endl; int b[7][7]; for (int i=1;i<=6;i++) for(int j=1;j<=6;j++) if (i==j j<i) b[i][j]=1; else b[i][j]=b[i+1][j-1]; cout<<b[6][1]<<endl; return 0; } </pre>	
2	int main() {	

	<pre> int a[]={7,2,5,4,3,2,1}; int b[]={3,2,7,3,2,5,8}; int c[10]; int i=0; for (;i<=6;i++) c[i]=a[i]+3*b[i]; cout<<c[5]; return 0; } </pre>	
3	<pre> int main() { int a[]={9,4,3,5,3,2}; int b[]={0,1,1,2,4,2,1,0}; int s=0; for (int i=0;i<=4;i++) s=s+a[b[i]]; cout <<s; return 0; } </pre>	
4	<pre> int main() { int a[]={3,2,1,0,2,3,1,1,2}; int b[10]; for (int i=0;i<=5;i++) b[a[i]-1]=b[a[i-1]]+1; cout<<b[4]; return 0; } </pre>	
5	<pre> #include <iostream> using namespace std; int main() { char a[7][8]={"There","are","four","people","in","my","family."}; cout<<a[0][0]<<endl; cout<<a[0][4]<<endl; cout<<a[0][5]<<endl; cout<<a[0]<<endl; cout<<a[1][0]<<endl; cout<<a[1][3]<<endl; cout<<a[1][5]<<endl; } </pre>	

	<pre> cout<<a[1]<<endl; return 0; } </pre>	
--	--	--

P207

特獎是6球都相同，頭獎是中1個特別號，與任5球，二獎是中了任5球，三獎是中了任4球，四獎是中了任3球。)

P209

16、請觀察捷運站都有顯示現在時間與還有幾分鐘班車進站，撰寫此程式，請問您的班車時刻表如何儲存與顯示。

17、請觀察高鐵火車站都有顯示現在時間與下一班車出發時間與停靠站，撰寫此程式，請問您的班車時刻表如何儲存與顯示。

P210

16、二維座標平面上，若有任意6個點，請寫一個程式，可以判定哪兩點的距離最近。

17、請模擬產生0到99的數字，請問第幾個數字時，剛好0到99可以通通填滿至少1個。

18、請模擬產生100個0到99的數字，請列出哪幾個數字沒有產生。

19、主客場籃球賽制 (APCS10806 試題)

(1)連打兩場，主場球隊兩場皆贏，輸出 Win。

(2)主場球隊兩場皆輸，輸出 Loss。

(3)其餘輸出 Tie。

(4)輸入格式：每場 4 節得分，主場先輸出。以下分別是主隊、客隊兩場比賽的四小節得分。

15 20 25 30 #第一天主場球隊四小節得分

15 16 17 18 #第一天客場球隊四小節得分

5 10 15 20 #第二天主場球隊四小節得分

5 6 7 8 #第二天客場球隊四小節得分

20、音階校正 (APCS108 年第三梯次試題)

給定若干音階，例如：

8 (8 個數字)

1 7 8 8 9 16 17 26

從第 4 個起，若

$|y_i - y_{i-1}| > 5$

則將此音修正為前 3 音的中位數。例如，以上音符修正如後如下：

1 7 8 8 9 8 8 8

以後不管

140-2

循序猜值法

所謂循序猜值法，就是將所有可能的解一一代入，又稱為暴力猜值法。例如，

您要求任一數的開根號，因為開根號的結果一定在 0 與此數之間，那我就將從 0

開始，每次遞增 1、0.1、0.01 或 0.001...，至於是 1、0.1、0.01 或 0.001...，那就

依您要的精密度了。以上整數解操作如下：

```
#include <iostream>
using namespace std;
int main() {
    int a=9,i;
    for (i=0;i<=a;i++)//0,1,2,3,一個一個猜
        if (i*i>=a)
            break;//找到就提早離開
    cout <<i;
    return 0;
}
```

實數解操作如下：

```
#include <iostream>
using namespace std;
int main() {
```

```

    int a=10;
    double i;
    for (i=0;i<=a;i=i+0.01)
        if (i*i>=a)
            break;//找到就提早離開
    cout <<i;
    return 0;
}
實作

```

P213

```

#include <iostream>
using namespace std;
int main() {
    int a=5,b=3,c;
    int *p,*q;
    p=&a;q=&b;
    cout<<*p<<endl;
    cout<<*q<<endl;
    c=*p+*q;
    cout<<c<<endl;
    return 0;
}

```

P232-1

3.傳回的角度是 $-\pi/2 \sim \pi/2$ 。

P232_2

傳回的角度是 $-\pi \sim \pi$ 。

p232-3

11.trunc

語法：`double trunc ( double arg );`

說明：去掉小數部分。

例如：`trunc(0.5)=0;trunc(-0.5)=0;`

12.rint

語法：`double rint( double arg );`

說明：取最近的整數。

例如：`rint(0.51)=1;rint(0.5)=0;rint(-0.51)=-1;rint(-0.5)=0;rint(-0.49)=0;`

實作

1、請執行以上程式片段。

2、請線上查詢以上數學函式。(<https://en.cppreference.com/w/> 點選『[Common math functions](#)』

P234

```
#include <iostream>
//#include<math.h>//載入 C 函式
#include <cmath>
using namespace std;
int main() {
    cout<<M_PI<<endl;
    int x=4,y=3;
    double r,s;
    r=sqrt(x*x+y*y);
    s=atan2(y, x)*180/M_PI;
    cout<<r<<" ° "<<s<<endl;
    return 0;
}
```

P239

3、若要以圖形界面顯示牌面，那就要先掃描這些牌面，分別以g0到g51儲存，請看 www.goodbooks.com.tw的JavaScript撲客牌遊戲。

P247

2、請寫出以下程式執行結果。

```
#include <iostream>
using namespace std;
int main(int argc, char** argv) {
    char a[]={'a','b','c','d','e','f','g','h','\0'};
    printf("%d",a[0]);
    printf("%c\n",a[0]);
    cout<<endl;
```

```

    for (int i=0;i<=7;i++){
        a[i]=a[i]+i;
    }
    for (int i=0;i<=7;i++){
        printf("%d",a[i]);
        printf(",");
        printf("%c\n",a[i]);
    }
    return 0;
}

```

P248_1

C語言沒有字串比較運算子(>,<,>=,==,<=)，所以『if(s1>s2)』是錯的，C++才有。

P248_2

C語言沒有字串指派運算子(=)，所以『s1=s2』是錯的，C++才有。

P249_1

C語言沒有字串連結運算子(+)，所以『s1=s1+s2』是錯的，C++才有。

P253-1

有了string型態就精彩了，這就和所有高階語言相同了，也就是關於字串的處理也要盡量和數值處理的習慣相同。請鍵入以下程式，並觀察執行結果。

```

#include <iostream>
using namespace std;
int main() {
    string a="12345";
    string b;
    b=a;//字串指派
    cout<<b<<endl;
    a=a+"abcd";//字串累加
    cout<<a<<endl;
    if (a==b)//字串比較
        cout<<"equal"<<endl;
    cout<<a[0]<<endl;//取子字串
}

```

```

        if (isdigit(a[0]))
            cout<<"isdigit"<<endl;
    }

```

關於C++的字串處理，線上查詢如下圖：

(https://en.cppreference.com/w/cpp/string/basic_string)

G901

253

，使用string物件的方法，請先載入

```
#include<string> or #include<cstring>
```

但因 `cstring` 包含 `string`，所以本書直接通通載入

```
#include<cstring>
```

P254

7.stoi:將字串轉數值。請鍵入以下程式，並觀察執行結果。

```

string a="123";
string b="123";
string c=a+b;//123123
cout<<c<<endl;
int d=stoi(a)+stoi(b);
cout<<d<<endl;//246

```

8.to_string:將數值轉字串。請鍵入以下程式，並觀察執行結果。

```

int a=123;
int b=123;
int c=a+b;
cout<<c<<endl;//246
string d=to_string(a)+to_string(b);
cout<<d<<endl;//123123

```

P256

實作

1、請寫一程式，可以檢驗某一字串出現的次數。例如，『There are four people in my family. My family is happy.』輸入『family』可得到2。

P257

6.以下是Python程式，請自己於Python執行，感受電腦運算思維的精準（電腦大約5次就可找出答案）。然後C++程式自己寫，把快樂留一些讓您享受。

```

a=[]
b=[]
for i in range(10):
    for j in range(10):
        for k in range(10):
            for l in range(10):
                a.append(str(i)+str(j)+str(k)+str(l))
print(len(a))
for i in range(len(a)):
    print(a[i])
#逐一蒐集數字不重複者
for i in range(len(a)):
    flag=True
    c=a[i]
    for j in range(3):
        for k in range(j+1,4):
            if c[j]==c[k]:
                flag=False
                break
    if flag:#沒有重複才蒐集
        b.append(c)
for i in range(len(b)):
    print(b[i])
a=b[:]#串列的複製
b=[]#重新清為空串列
print(len(b))
right=False
while (not right):
    d=a[0]
    print ('Computer guess:%s' %d)
    a1,b1=eval(input('input na , nb:'))
    if a1==4 :
        right=True
        print("Bingo")
        break
#將相同 a 相同 b 的蒐集起來
for i in range(1,len(a)):
    a2=0

```

```

b2=0
f=a[i]
for j in range(4):
    if f[j]==d[j]:
        a2=a2+1
    for k in range(4):
        if f[j]==d[k]:
            b2=b2+1
b2=b2-a2
if a1==a2 and b1==b2 :
    b.append(f)
a=b[:]
b=[]

```

範例 9_4b 影像解壓縮（107/03/02 試題）

同一張照片中，解析度取的越精密，相鄰的像素的值相同的比例就會很高，所以就有很多壓縮技術，以下是最簡單通用的影像壓縮技術。若連續出現 6 個 A，5 個 B，那就以 6A5B 表示。請寫一程式，將一個壓縮的檔案，還原為原始檔案。例如，

輸入：1A1P1C1S3AC14KDA（1 個 A，1 個 P，1 個 C，1 個 S，3 個連續 AC...

輸出：APCSACACACKDAKDA(KDA 共 14 次，如下圖)

程式列印

這種題目對高中生很棒很合理，可以精進迴圈的使用。演算法如下：

1、處理每個子集合

1_1、抓數字，但還要繼續抓連續數字，例如1或14（此一while迴圈技術，請熟練，因為考題一直出現）。

1_2、抓字元，但還要抓連續字元，例如『A』或『KAD』。

1_3、輸出一個單元。所以程式如下：

```

#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    string a="1A1P1C1S3AC14KAD";
    string b;
    string c;
    int i=0;
    int j,k,p;

```



```

//計算字串長度
int n=a.length();
do {
    //抓每一子集合第1個數字
    b=a[i];
    i++;
    int d=stoi(b);//轉為數字
    //繼續抓連續數字
    while (isdigit(a[i])){
        b=a[i];
        d=10*d+ stoi(b);//總數
        i++;
    }
    //d 是每個子集合個數
    //抓每一子集合第1個字元
    c=a[i];
    //繼續抓連續字元
    i++;
    while (isalpha(a[i])){
        c=c+a[i];
        i++;
    }
    //輸出一個子集合
    for (int k=1 ;k<=d;k++){
        cout<<c;
    }
    //一個子集合完畢
}while (i!=n-1);
return 0;
}

```

實作

- 1、請寫一個程式，可以計算一個字串有多少個最長連續大寫。例如，AABaabABCDFa為5個。

範例9_4c 孤獨字串。

當一個字串，使用字母(A ~ Z)的個數最少,則他就是一個孤獨字串。例如，

“HELLO”

字串長度5，但是使用四個字母組成，但是

“ABBBAA”

字串長度 6，僅使用 2 個字母組成，那就比上一個字串『HELLO』孤獨，本例請於所輸入字串中，找出最孤獨字串，也就是含字母種類最少者。其次，若孤獨字串有兩個以上，請繼續將這些孤獨字串按照字母順序排列，找出排在最前面的字串。例如，『ABBBAA』與『AAAABB』均僅含兩個字母，那就還要繼續比較，找出字母順序最前面的字串，所以是『AAAABB』，以下是參考的測資：

5

HELLO

ABBBAA

AAAABB

GOODBYE

ABCDEFGHIJKL

表示 5 個測資，則輸出

AAAABB

輸出結果

考試的時候，不可能一次就對，那就一面寫，一面輸出看看，結果對了，再繼續做下去，本例輸出如下，等到都對了，再將這些輸出通通刪除。

G902

程式列印

- 1、計算每一個字母出現的個數。我是用陣列挖26個洞，然後將每一個字母丟進每一個字母的洞，例如，A就進第1個洞，B就近第2個洞。
- 2、計算每一個字串含有幾個字母。上面26個洞，統計不是0的個數，那就擁有幾個字母。
- 3、找出字母個數的極小值。
- 4、統計擁有極小值的個數。
- 5、若是個數只有1個，那輸出此字串，結束了。
- 6、若最小值不只一個，那就繼續找這些字串的最小值。

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int N=5;//測資個數
```

```
    string a[]={"HELLO","ABBBAA","AAAABB","GOODBYE","ABCDEFGHIJKL"};
```

```
    //string a[]={"CC","ABBBAA","AAAABB","GOODBYE","ABCDEFGHIJKL"};
```

```
    int b[100][28];
```

```
    //b[][0] 不用
```

```
    //b[][1~26] is the frequency of the char
```

```
    //b[][27] is the number of char（使用字母的個數）
```

```

string e[100]; //minstring
int i,j;
char c;
int d;
int n=0;
c=a[0][0];
cout<<c<<endl; //H
d=(int)c-64;
cout<<d<<endl; //8
for (i=0;i<=N-1;i++){
    //吃龜苓膏
    for (j=1;j<=26;j++){
        b[i][j]=0;
    }
    for (j=0;j<a[i].length();j++){
        c=a[i][j];
        d=(int)c-64;
        b[i][d]++; //按照字母位置，丟進去
    }
    //計算使用字母的個數
    n=0;
    for (j=1;j<=26;j++){
        if (b[i][j]!=0)
            n++;
    }
    b[i][27]=n;
}
//輸出使用字母的個數
for(i=0;i<=N-1;i++){
    cout<<b[i][27]<<" ";
} //4 2 2 6 12
cout<<endl;
int min=100;
int minindex=0;
for(i=0;i<=N-1;i++){
    if (b[i][27]<=min){
        min=b[i][27];
        minindex=i;
    }
}

```


演算法則

這題有點難度了，看著提示二，經過不斷思考，我的演算法如下，

(1) 先將所有大小寫轉為1 與-1 的陣列。以

```
k=3;  
a="DDaassAAAbbCC";//長度6
```

為例，其結果如下：

```
1,1,-1,-1-1,-1,1,1,1,-1,-1,1,1
```

(2) 求出每群大寫或小寫的長度。上例結果如下：

```
2,4,3,2,2 //以上兩個階次還可簡化成一階次，請自我練習
```

(3) 依據題意，找出指定k 值的長度。本例k=3，所以找到索引2 為3。

(4) 其次，觀察以上陣列，符合k 值子字串的前後，若長度大於k 也要計算在

內。例如，本例索引2 的3 的前面有1 個4，那這也是交錯字串，所以交錯字串

長度是6。又例如，

```
2,4,3,5,2
```

那3 的前後面都是，所以交錯字串長度將會是9。(但要留意，陣列第1個不用再

檢查前面，最後1個，不用再檢查後面)

(5) 有時候交錯字串還會有許多群，所以還要找出最長的群。例如，

```
k=3  
a="DDaassAAAbbCCasAssssssAAAbbbAAAbbbAAAAA";//長度18
```

第1 群是6，但是第2 群是18，所以答案是18。

(6) 以下是

```
k=3;  
a="DDaassAAAbbCC";
```

的輸出結果。考試的時候，當然不可能一次答對，所以就這樣，一面做題目，

一面輸出結果，這個步驟對了，再繼續往下作。

G903

程式列印

```
#include <iostream>
using namespace std;
int main()
{
    int k;
    string a;
    int c[100000];
    k=1;
    a="aBBdaaa";//2
    //a="aBBdaaaBaBa";//5
    //a="aBBdaAaBaBa";//7
    //k=2;
    //a[]="aafAXbbCDCCC";//8
    //a[]="aAAaafAXbbCDCCC";//8
    //a[]="AAaafAXbbCDCCC";//8,
    k=3;
    a="DDaassAAAbbCC";//3
    //a="DDaaAAbbCC";//no
    a="DDaassAAAbbCC";//6
    //a="DDaassAAAbbCCasAssssssAAAbbbAAAbbbAAAAA";//18
    char b;
    int d[100000];
    int num=a.size();
    int i=0;
    int j;
    //轉為小寫是-1，大寫是1的陣列
    for (i=0;i<num;i++) {
        b=a[i];
        if (islower(b))
            c[i]=-1;
        else
            c[i]=1;
    }
    for (i=0;i<num-1;i++){
```

```

        cout<<c[i]<<" ";
    }
    cout<<endl;
    //算出每一個連續字串的長度
    int start;
    int sum;
    start=c[0];//抓一個開頭
    i=1;sum=1;
    j=1;
    while (i<=num){
        while (c[i]==start ) {
            start=c[i];//踩進來，前進
            sum++;
            i++;
        }
        d[j]=sum;
        sum=1;
        i++;
        j++;
        start=-start;
    }
    int num2=j-1;
    for (j=1;j<=num2;j++){
        cout<<d[j]<<" ";
    }
    cout<<endl;
    //計算交錯字串長度
    i=1;sum=0;
    j=1;
    int flag=0;
    int max=0;
    while ( i<=num2){
        while (d[i]==k ) {
            sum=sum+k;//踩進來
            flag=1;
            if (i>=2 && d[i-1]>k)//陣列第1個，不能再檢查前面
                sum=sum+k;//檢查前面>k
            i++;

```

```

    }
    cout<<i<<" "<<d[i]<<endl;
    if (flag && i<=num& d[i]>k ) //陣列最後1個，不能再檢查後面
        sum=sum+k;//檢查後面>k
        i++;
    //完成一群
    flag=0;
    if (sum>max){//又找到另一群
        max=sum;
        cout<<"max="<<max<<endl;
        sum=0;
    }
}
cout<<max<<endl;
return 0;
}

```

實作

- 1、請寫一程式，可以檢驗所輸入的字串是否大小寫交錯。例如，ACbdRT 有交錯，aaaBGHdfg 有交錯，但是 aFHKnnGJ 沒交錯。
- 2、檢查所給資料的週期。例如，1,2,3,4,5,6,1,2,3,4,5,6,1,2,3,4 是 6。又例如，3,4,2,1,3,4,2,1,3,4,2,1 是 4。5,8,2,7,1,5,8,2,7,1,5,8,2 是 5。

P259

還是不瞭解玩法的，請到 www.goodbooks.com.tw JavaScript 資料夾試玩，然後將此程式以 C++改寫。

P265

也可以將函數放在主程式前面，這樣不用函數原型宣告，程式如下：

```

#include <iostream>
using namespace std;
//1.實作函式實體
int add (int a, int b){ //a,b 不可省略
    int c;
    c = a + b;
    return (c);
}
int main() {
    int c;

```



```

//2.函式呼叫
c = add (6, 2);
cout<<c;
return 0;
}

```

P267

2、前面已經完成加法函數，請繼續完成減法、乘法、除法與取餘數函數。

P270

請依照範例名稱置換

10_2a

```

#include <iostream>
using namespace std;
void swap(int, int);
int main() {
    int x1=3, x2=4;
    swap(x1, x2);
    cout<<"3: "<< x1<<" , "<< x2<<endl;
    return 0;
}
void swap(int p, int q){
    int t;
    cout<<"1: "<< p<<" , "<< q<<endl;
    t=p; p=q; q=t;
    cout<<"2: "<< p<<" , "<< q<<endl;
}

```

10_2b

```

#include <iostream>
using namespace std;
void swap(int *, int *);
int main() {

```

```

    int x1=3, x2=4;
    swap(&x1, &x2);
    cout<<"3: "<< x1<<" , "<< x2<<endl;
    return 0;
}
void swap(int *p, int *q){
    int t;
    cout<<"1: "<< *p<<" , "<< *q<<endl;
    t=*p; *p=*q; *q=t;
    cout<<"2: "<< *p<<" , "<< *q<<endl;
}

```

10_2c

```

#include <iostream>
using namespace std;
void swap(int &, int &);
int main() {
    int x1=3, x2=4;
    swap(x1, x2);
    cout<<"3: "<< x1<<" , "<< x2<<endl;
    return 0;
}
void swap(int &p, int &q){
    int t;
    cout<<"1: "<< p<<" , "<< q<<endl;
    t=p; p=q; q=t;
    cout<<"2: "<< p<<" , "<< q<<endl;
}

```

10_2d

```

#include <iostream>
using namespace std;
void swap(int*, int*);
int main() {
    int a[]={0, 77, 66, 99, 44, 55};
    int i, j;
    const int N=5;
    cout<<"Before sort:"<<endl;

```

```

    for(i=1; i<=N; i++){
        cout.width(5);
        cout<<a[i];
    }
    cout<<endl;
    /* sort */
    for(i=1; i<=N-1; i++)
        for(j=0; j<=N-i; j++)
            if(a[j]>a[j+1])
                swap(&a[j], &a[j+1]);
    /* output */
    cout<<"After sort:"<<endl;
    for(i=1; i<=N; i++){
        cout.width(5);
        cout<<a[i];
    }
    return 0;
}
void swap(int *p, int *q){
    int t;
    t=*p; *p=*q; *q=t;
}

```

10_2e

```

#include <iostream>
using namespace std;
void printarray (const int a[ ], int n);
void bubblearray(int a[], int n);
int main() {
    int a[]={0, 77, 66, 99, 44, 55};
    int i, j;
    const int N=5;
    cout<<"Before sort:"<<endl;
    printarray(a, N);
    bubblearray(a, N);
    // output
    cout<<"After sort:"<<endl;
    printarray(a, N);
}

```

```

        return 0;
    }
    void bubblearray(int a[], int n){
        int i, j, t;
        void swap(int *, int *);
        for(i=1; i<=n-1; i++)
            for(j=0; j<=n-i; j++)
                if(a[j]>a[j+1])
                    swap(&a[j], &a[j+1]);
    }
    void swap(int *p, int *q){
        int t;
        t=*p; *p=*q; *q=t;
    }
    void printarray (const int a[ ], int n){
        for(int i=1; i<=n; i++){
            cout.width(5);
            cout<<a[i];
        }
        cout<<endl;
    }
}

```

10_2f

```

#include <iostream>
#include <stdlib.h>
#include <cstring>
#define s 5 //stdnum
#define c 4 //column
void getarray(int[][c+1], int);
void printarray(int[][c+1], int);
using namespace std;
int main(int argc, char** argv) {
    //char *title[]={" ", "    NO", "    CHI", "    ENG", "    MATH"};
    //for(int i=1; i<=c; i++){
        //cout<< title[i];
    //}
    string title="    NO    CHI    ENG    MATH";
    int a[s+1][c+1];
}

```

```

        cout<<title<<endl;
        getarray(a, s);
        printarray(a, s);
        return(0);
    }
    void getarray(int a[][c+1], int n){
        for(int i=1; i<=n; i++) {
            a[i][1]=i;
            for(int j=2; j<=c; j++)
                a[i][j]=rand()%101;
        }
    }
    void printarray(int a[][c+1],int n){
        for(int i=1;i<=n;i++){
            for (int j=1;j<=c;j++){
                cout.width(6);
                cout<<a[i][j];
            }
            cout<<endl;
        }
    }
}

```

10_3a

```

#include <iostream>
using namespace std;
int sum(int n);
int main() {
    int n=6, d;
    d=sum(n);
    cout <<d;
    return 0;
}
int sum(int n){
    int p;
    if(n>=1)
        p=n*sum(n-1);
    else
        p=1;
}

```

```
    return (p);  
}
```

p278

遞迴(Recursive) 是一種分而治之的經典方法，因為在分治法裡，常常遇到先處理一個或一部份，剩下部分的方法竟然和原方法相同，既然和原方法相同，那可否呼叫或使用原函式？答案是可以的，這樣重複呼叫原函式，就稱為遞迴（此一演算法在高中數學有介紹，所以當然是APCS 考試重點）。

P279

前面的迭代法，是從前面一個一個作，遞迴則是從外面剝皮，先剝一層，剩下的也和原方法相同，那就繼續呼叫原函數，有點您傻瓜，他聰明，您只要會

『畫老虎、畫蘭花』就好，反正電腦會幫您完成所有的工作。所以，看者演算法，程式撰寫如下：

```
#include <iostream>  
using namespace std;  
int sum(int n);  
int main() {  
    int n=6, d;  
    d=sum(n);  
    cout <<d;  
    return 0;  
}  
int sum(int n){  
    int p;  
    if(n>1)  
        p=n*sum(n-1);//先處理自己，剩下 n-1 的方法也是相同  
    else  
        p=1;  
    return (p);  
}
```

```
}
```

遞迴函數也有人這樣寫：

```
int sum(int n){  
    if(n>=1)  
        return(n*sum(n-1));  
    else  
        return(1);  
}
```

P281

```
#include <iostream>  
using namespace std;  
int f(int n);  
int main() {  
    int n=6, d;  
    d=f(n);  
    cout <<d;  
    return 0;  
}  
int f(int n){  
    int p;  
    if(n>2)  
        p=f(n-1)+f(n-2); //n>2時，每一個都直接分解  
    else  
        p=1;  
    return (p);  
}
```

函數也可以這樣寫：

```
int f(int n){  
    if(n>2)  
        return(f(n-1)+f(n-2));  
    else  
        return(1);  
}
```

範例10_3c 請寫一個程式，使用遞迴法將字串反向輸出。

程式列印

每次都是先求此字串長度，只要字串長度不為0，那就是輸出就後一個字元，且

將剩下的前面字串繼續呼叫此方法，所以程式如下：

```
include <iostream>
#include <cstring>
using namespace std;
void aa(string a){
    int b=a.length();
    if (b!=0){
        cout<<a[b-1];//先印出最後一個字元
        aa(a.substr(0,b-1));//剩下的前面n-1，方法也是相同
    }
    else
        return ;
}
int main() {
    string a="12345";
    aa(a);
}
```

範例10_3d 請寫一個程式，使用遞迴法將整數反向輸出。

程式列印

每次都是求餘數與商，商大於0，就輸出餘數，且繼續呼叫自己。商等於0時，最後一個餘數也要輸出，所以程式如下：

```
#include <iostream>
using namespace std;
void aa(int n){
    int p,q;
    p=n%10;//求餘數
    q=n/10;
    if (q>0){
        cout<<p;//先輸出餘數
        aa(q);//剩下的也相同
    }
    else
        cout<<p<<endl;
}
int main() {
    int n=1234567;
    aa(n);
    return 0;
}
```



```
}
```

範例10_3e 請寫一個程式，使用遞迴法求出兩數的最大公因數。

每次都是求『餘數』，只要『餘數』大於0，就繼續將『除數』當『被除數』，『餘數』當『除數』，繼續呼叫自己，當『餘數』為『零』時，當時的『除數』就是『最大公因數』，所以程式如下：

```
#include <iostream>
using namespace std;
int gcd(int a,int b){
    int r;
    r=a %b;
    if (r>0){
        return(gcd(b,r));
    }
    else
        return(b);
}
int main() {
    int a=9,b=21,t;
    if (b> a){
        t=a;a=b;b=t;
    }
    int n=gcd(a,b);
    cout<<n<<endl;
    return 0;
}
```

範例10_3f 函數運算。108 年2 月APCS 試題，題目大致如下：定義

$f(x)=x+1, g(x,y)=x+y, h(x,y,z)=x+y-z$ ，輸入是簡化的，請看以下測試資料。其

次，評分依照3 個等級給分。等級1 例如， $f\ 1$ ，代表 $f(1)$ ，並求其值，答案是2;

$f\ f\ 1$ ，代表 $f(f(1))$ ，並求其值，答案是3，這樣可得10 分；等級2 例如， $h\ f\ 1$

$g\ 2\ 3\ 4$ 代表 $h(f(1),g(2,3),4)$ ，並求其值，答案是3，這樣可得40 分；等級3例

如， $h\ f\ 10\ g\ 20\ 300\ 4000$ ，數字長度不限，這樣可得滿分。

程式列印

以下是第1小題的參考解答，剩下的自己做，把樂趣留給您。（關於連續數字，

例如『14』的抓法，請複習範例6_4j的APCS考題）

```
#include <iostream>
using namespace std;
int f(string a,int p){
    char b=a[p];
    if (isdigit(b)){
        int x=(int)b-48;
        return(x);//遇到數值，抓數值回去
    }
    else {
        p=p+2;
        int x=(f(a,p));//遇到f要繼續取一個，且將結果傳回去
        return(x+1);//並回結果
    }
}
int main() {
    //string a="f f l";
    string a="f l";
    int n=0;
    int m=f(a,n);
    cout<<m<<endl;
    return 0;
}
```

實作

- 1、請用遞迴法重做6_1a。
- 2、請用遞迴法重做，以暴力法解平方根。
- 3、請用遞迴法重做，以二分猜值法解平方根。
- 4、請用遞迴法重做範例6_3a。
- 5、請用遞迴法重做6_3b。
- 6、請寫出以下程式執行結果（APCS試題）。

```
#include <iostream>
using namespace std;
int n=6;
int main() {
    cout <<n<<endl;
```

```

        n=n-1;
        if (n>=3)
            main();
        return 0;
    }

```

7、請寫出以下程式執行結果（APCS試題）。

```

#include <iostream>
using namespace std;
int f(int x,int y){
    if (x==0)
        return y;
    else
        f(x-1,x+y);
}
int main(int argc, char** argv) {
    int a=f(10,10);
    cout<<a<<endl;
    return 0;
}

```

P283

2、請寫三個函式，函式名稱都是 **aa()**，可以解一元一次、二元一次與三元一次方程式。請用參數的數量決定執行哪一個 **aa()** 函數。

P290

1、請設計一個類別，內有 3 個資料成員 **a,b,c**，4 個方法成員，分別是 **add(a,b)**、**sub(a,b)**、**mul(a,b)**、**div(a,b)**，可以得到兩個數 **a,b** 的加、減、乘、除等結果，此結果請放在 **c**。例如，您用 **p** 樣例類別後，**cout<<p.add(2,3)**，可以得到 5。

P293

1、請將上一範例的 **div()** 函數，加上先判斷除數是否為零的機制，除數不為零，才能執行除法。

P301

1、請將範例 11_1a 的實作 1，加上繼承，新增 1 個取餘數的方法。

- 2、請將範例 11_1a 的實作 2，加上繼承，新增 1 個計算清潔費的方法，清潔費的計算是 $50 + \text{使用度數} * 2$ 。

P304

- 1、請將範例 11_1a 的實作 1，加上繼承，將加法改為『 $2 * (a + b)$ 』。也就是 $2 + 3$ 是等於 10。
- 2、請將範例 11_1a 的實作 2，加上繼承，將水費改為『還要增加基本費 100 元』。

P333

第十三章 跨領域自主學習歷程檔案（數學函數圖形的繪製）

P334

目前雖然有很多功能很強的繪圖軟體，但高中數學函數只是很簡單的圖形，只要定義『作圖範圍』與使用『繪點指令』就可繪出函數圖形，所以殺雞不用牛刀，本文選用 Visual Basic 6.0 就綽綽有餘，請看以下說明。

p354

將變數 i、空白數、星星個數整理如下：

i	空白個數	星星個數
1	4	1
2	3	3
3	2	5
4	1	7
5	0	9

由上表可知，是星星個數錯誤，所以將 $m = m + 1$ 改為 $2 * i + 1$ 即可。

p355

1、用人腦追蹤如下：

i	a[i]	index	a[index]	index
1	3	0	1	1
2	9	1	3	2
3	2	2	9	2
4	5	2	9	2
5	8	2	9	2
6	4	2	9	2
7	9	2	9	7

8	6	7	9	7
9	7	7	9	7

2、用電腦追蹤其值如下：(斜粗體字)

p369

此題有遞迴又有堆疊，當副程式程式繼續呼叫副程式時，若還有未執行的程式片段，則要先進堆疊，等到遇到return 時，就要將堆疊全部輸出。

$f(1)=1+f2(3)+$

1

堆疊

$=1+3+f1(2)+$

3
1

堆疊

$=1+3+2+f2(4)+$

2
3
1

堆疊

$=1+3+2+4+return$ (要清光堆疊)

$=1+3+2+4+2+3+1$

p371

此題數字很大，應先找其規律性簡化問題。我們發現都是每15個就1個循環回到自己。所以先扣到大於15

$$15106/15=1007\cdots 1$$

$$15106-15*1006=16$$

```
foo(15106)=foo(16)=bar(6)=bar:6  
bar(3091)=bar(16)=foo(11)=bar(1)=bar:1  
foo(6693)=foo(18)=bar(8)=bar:8
```

p377

f(2)=5 (看上面，已經重複)

f(3)=2 (看上面，已經重複)

P380

參考程式如下：

```
#include <iostream>  
#include <algorithm> //使用 sort方法  
using namespace std;  
int main() {  
    int n;  
    int a[21];  
    int i;  
    cin >> n;  
    for (i=0; i<=n-1; i++){  
        cin >> a[i];  
    }  
    cout << endl << endl;  
    sort(a, a+n); //使用 <algorithm>  
    for (i=0; i<=n-1; i++){  
        cout << a[i] << " ";  
    }  
    cout << endl;  
    if (a[0] >= 60) {  
        cout << "best case" << endl;  
        for (i=0; i<=n-1; i++){  
            if (a[i] >= 60){  
                cout << a[i] << endl;  
                break;  
            }  
        }  
    }  
}
```

```

else
    if (a[n-1]<=60){
        for (i=n-1;i>=0;i--){
            if (a[i] <60){
                cout<<a[i]<<endl;
                break;
            }
            cout<<"worst case"<<endl;
        }
    }
    else {
        for (i=n-1;i>=0;i--){
            if (a[i] <60){
                cout<<a[i]<<endl;
                break;
            }
        }
        for (i=0;i<=n-1;i++){
            if (a[i] >=60){
                cout<<a[i]<<endl;
                break;
            }
        }
    }
    return 0;
}

```

測試資料很多時，每次執行程式都打一遍，這樣很浪費時間，此時可以找一個類似記事本的程式、或於CodeBlocks新增一個空白的檔案(點選『File/New Empty File』)，將測試資料打在此，如下圖，每次就複製資料來貼就好，這樣比較省時間。

E01

p385

右邊是已知，且r=4,c=3，右邊旋轉為左邊，旋轉後

(1,3)->(1,1) (2,3)->(1,2) (3,3)->(1,3) (4,3)->(1,4)
 (1,2)->(2,1) (2,2)->(2,2) (3,2)->(2,3) (4,2)->(2,4)
 (1,1)->(3,1) (2,1)->(3,2) (3,1)->(3,3) (4,1)->(3,4)

規則是

i->j (請看斜體字)

i+j=c+1 （請看底線字）

所以，

b[i][j]=a[j][c-i+1]

參考程式如下：

```
#include <iostream>
using namespace std;
int main() {
    int a[10][10];
    int b[10];
    void rotate(int a[10][10], int &r, int &c);
    void turn(int a[10][10], int r, int c);
    void print(int a[10][10], int r, int c);
    int r,c,m;
    cin >>r>>c>>m;
    int i,j;
    for (i=1;i<=r;i++){
        for (j=1;j<=c;j++){
            cin>>a[i][j];
        }
    }
    //操作方式
    for (i=1 ;i<=m;i++){
        cin>>b[i];
    }
    cout<<endl<<endl;
    //顛倒作回來
    for (i=m;i>=1;i--){
        switch(b[i])    {
            case 0:      //旋轉
                rotate(a,r,c);
                //print(a,r,c);//可先輸出看看
                break;
            case 1:      //翻轉
                turn(a,r,c);
                //print(a,r,c);
                break;
        }
    }
}
```



```

    print(a,r,c);//題目只要求最後結果，所以在此輸出
    return 0;
}

void rotate(int a[10][10], int &r, int &c){
    int b[10][10];
    int rr=c;
    int rc=r;
    int i,j;
    for (i=1;i<=rr;i++)
        for (j=1;j<=rc;j++)
            b[i][j]=a[j][c-i+1];

    for (i=1;i<=9;i++)
        for (j=1;j<=9;j++)
            a[i][j]=b[i][j];

    r=rr;
    c=rc;
}

void turn(int a[10][10], int r, int c){
    int b[10][10];
    int i,j;
    for (i=1;i<=r;i++)
        for (j=1;j<=c;j++)
            b[i][j]=a[r-i+1][j];

    for (i=1;i<=9;i++)
        for (j=1;j<=9;j++)
            a[i][j]=b[i][j];
}

void print(int a[10][10], int r, int c){
    int i,j;
    cout<<r<<" "<<c<<endl;
    for(i=1; i<=r; i++) {
        for(j=1; j<=c; j++){
            //cout.width(5);
            cout<< a[i][j]<<" ";
        }
        cout<<endl;
    }
}

```

```

    }
    cout<<endl;
}

```

P388

這一題坊間的書、網路或高手的解法可說各家增鳴大放異彩，但大都是用大學的線段長度演算法，唯有學過大學的演算法才能理解。我的想法就簡單了，我是將線段數位化，首先，宣告1個通通為0的大陣列，線段(4,8)的數位化，就將4,5,6,7,8索引的值通通指派為1。下表是將(5,6)、(1,2)、(4,8)、(7,9)數位化的結果：

	0	1	2	3	4	5	6	7	8	9	10
(5,6)	0	0	0	0	0	1	1	0	0	0	0
(1,2)	0	1	1	0	0	0	0	0	0	0	0
(4,8)	0	0	0	0	1	1	1	1	1	0	0
(7,9)	0	0	0	0	0	0	0	1	1	1	0
OR	0	1	1	0	1	1	1	1	1	1	0

然後將所有的陣列，逐一每一索引，通通取邏輯OR運算，得到

0110111110

最後就是植樹問題，兩個連續1，才得到樹蔭長度1，所以以上線段長度是『6』，參考程式如下：

```

#include <iostream>
using namespace std;
void fill(bool a[1000000],int b,int c){//函式寫在前面，可省略宣告
    unsigned long j;
    const unsigned long M=1000000;
    for ( j=0 ;j<=M;j++)
        a[j]=0;
    for (j=b ;j<=c;j++)
        a[j]=1;
}
int main() {
    int num;
    int b,c;
    bool a[1000000];
    bool d[1000000];
    unsigned long j,n;
    const unsigned long M=1000000;

```

```

cin >>num;
cin>>b>>c;
fill(a,b,c);
for (int i=1;i<=num-1;i++){
    cin>>b>>c;
    fill(d,b,c);
    for ( j=0;j<=M;j++)//新增的線段與原線段作OR運算
        a[j]=a[j] || d[j];
}
for (j=0;j<=300;j++)
    cout<<a[j];
cout<<endl;
j=0;
n=0;
do{
    while(a[j]==0 & j<M) { //略過0
        j++;
    }
    if (j==M)
        break;
    while(a[j]==1 & j<M) { //計算1的個數
        j++;
        n++;
    }
    n--; //植樹問題，間隔距離比樹木少一
}while (j<M);
cout <<n;
return 0;
}

```

以上a[]、d[]陣列最多僅能到1000000，但若改為全域變數，竟然就可以到10000000，滿足題目的要求，程式如下：

```

#include <iostream>
using namespace std;
bool a[10000000];
bool d[10000000];
void filla(int b,int c){ //函式寫在前面，可省略宣告
    unsigned long j;
    const unsigned long M=10000000;

```

```

        for ( j=0 ;j<=M;j++)
            a[j]=0;
        for (j=b ;j<=c;j++)
            a[j]=1;
    }
void filld(int b,int c){//函式寫在前面，可省略宣告
    unsigned long j;
    const unsigned long M=100000000;
    for ( j=0 ;j<=M;j++)
        d[j]=0;
    for (j=b ;j<=c;j++)
        d[j]=1;
}
int main() {
    int num;
    int b,c;
    unsigned long j,n;
    const unsigned long M=100000000;
    cin >>num;
    cin>>b>>c;
    filla(b,c);
    for (int i=1;i<=num-1;i++){
        cin>>b>>c;
        filld(b,c);
        for ( j=0;j<=M;j++)//新增的線段與原線段作OR運算
            a[j]=a[j] | d[j];
    }
    //for (j=0;j<=300;j++)
        //cout<<a[j];
    //cout<<endl;
    j=0;
    n=0;
    do{
        while(a[j]==0 & j<M) { //略過0
            j++;
        }
        if (j==M)
            break;
    }

```

```

        while(a[j]==1 & j<M) { //計算1的個數
            j++;
            n++;
        }
        n--; //植樹問題，間隔距離比樹木少一
    } while (j<M);
    cout <<n;
    return 0;
}

```

P397

被除數與除數都是整數，結果也是整數，3個都是1，所以答案是3。

P414

```

#include <iostream>
using namespace std;
int main() {
    int a[21][21];
    int b[21];
    int r,c;
    int i,j;
    int s=0;
    while(cin>>r>>c){
        for (i=1;i<=r;i++){
            for (j=1;j<=c;j++){
                cin>>a[i][j];
            }
        }
        for (i=1;i<=r;i++){
            b[i]=a[i][1];
            for (j=2;j<=c;j++){
                if (a[i][j]>b[i])
                    b[i]=a[i][j];
            }
        }
        //求和
        s=0;
    }
}

```

```

        for (i=1;i<=r;i++)
            s=s+b[i];
        cout<<s<<endl;
        //找整除者
        bool found=false;
        for (i=1;i<=r;i++){
            if(s % b[i]==0){
                found=true;
                cout<<b[i]<<" ";
            }
        }
        if (!found)
            cout<<"-1"<<endl;
    }
    return 0;
}

```

P416

```

#include <iostream>
using namespace std;
struct element {
    unsigned long data;//資料
    unsigned long next;//指標
};
void print(struct element a[],unsigned long current,unsigned long length ){
    unsigned long i;
    for (i=1 ;i<=length;i++){
        cout<<a[current].data<<" ";
        current=a[current].next;
    }
    cout<<endl<<endl;
}
int main() {
    struct element a[200000];
    unsigned long n,m,k,i;
    while (cin >>n>>m>>k){
        for (i=0 ;i<=n-2;i++){
            a[i].data=i+1;

```

```

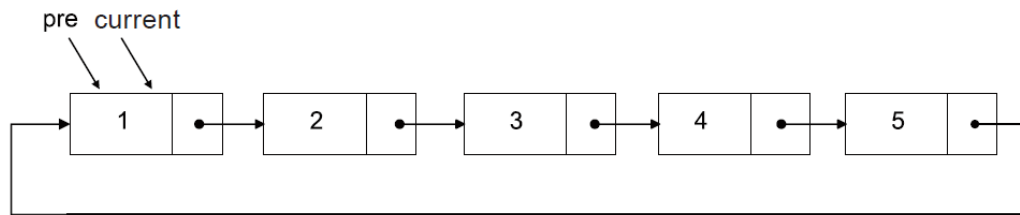
        a[i].next=i+1;
    }
    a[n-1].data=n;
    a[n-1].next=0;//環狀鏈串列
    print (a,0,n);
    unsigned long d=0;
    unsigned long current=0;
    unsigned long pre=0;
    i=0;
    do{
        d=d+1;    //計數器
        if (d==m){
            cout<<a[current].data<<" will be removed"<<endl;
            //remove the current
            //move the pre point to thr current point
            a[pre].next=a[current].next;
            n=n-1;    //the total number
            cout<<"The Remainder is:";
            print(a,pre,n);
            d=0;
            i++;//爆炸次數
        }
        pre=current;
        current=a[current].next;
    } while (i<k);
    cout<<endl;
    cout<<a[current].data;
}
return 0;
}

```

P418

程式說明

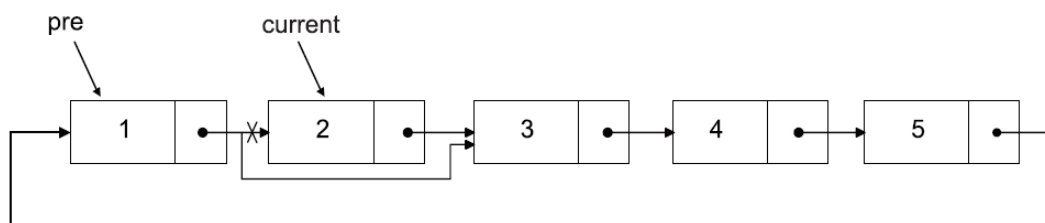
1. 本例初始狀態pre、current 指標如下：



2. 鏈串列的追蹤，pre 永遠在current 的前面，要刪除current 的資料，都是將

pre 的指標，指向current 的指標，那current 就孤立，宛如消失，如下圖：

```
//move the pre point to thr current point
a[pre].next=a[current].next;
```



但其實current 所指向資料還是存在，只是沒被連結，例如您刪除硬碟的資料，

其實此資料還是存在，只是沒被連結，透過其他技術還是可復原這些資料。

P421-1

2、本例測試資料很多，請先開新空白檔案『File/New/Empty file』，將測試資料先打好，執行時，再複製與貼上。

p421 -2

```
string s[10][6];
switch(s[i][j][0])    {}
```

p422

```
#include <iostream>
#include <queue> //使用佇列
using namespace std;
int main() {
```



```

string s[10][6];
int num;
int tnum=0;//題目給的總打擊人數
int tnum1=0;//累計出場人數
int a;
int out,outr;//出局數，總出局數
int c[10];//每局得分
queue <int> q;
q.push(0);//初始化3個壘包
q.push(0);
q.push(0);
int i,j;
for (i=1;i<=9;i++)
    c[i]=0;//每局得分歸零
for (i=1;i<=9;i++){//9人
    cin >>num; //每列人數
    tnum=tnum+num;//總打擊人數
    for ( j=1 ;j<=num;j++){
        cin>>s[i][j];
    }
}
cin >>outr;//題目要求的出局數
for(j=1 ;j<=5;j++){
    for (i=1 ;i<=9;i++){
        tnum1++;//逐一累計出場人數
        if (tnum1==tnum)//出場人數 題目給的總打擊人數
            break;
        switch(s[i][j][0])    {
            //switch運算式僅能整數或字元，所以取該字串第一個字元
            case '1'://1B
                q.push(1);
                a=q.front();
                c[outr/3+1]=c[outr/3+1]+a;//得分
                q.pop();
                break;
            case '2'://2B
                q.push(1);
                a=q.front();

```

```

        c[out/3+1]=c[out/3+1]+a;
        q.pop();

        q.push(0);
        a=q.front();
        c[out/3+1]=c[out/3+1]+a;
        q.pop();
        break;
case '3'://3B
        q.push(1);
        a=q.front();
        c[out/3+1]=c[out/3+1]+a;
        q.pop();

        q.push(0);
        a=q.front();
        c[out/3+1]=c[out/3+1]+a;
        q.pop();

        q.push(0);
        a=q.front();
        c[out/3+1]=c[out/3+1]+a;
        q.pop();
        break;
case 'H'://HR
        q.push(1);
        a=q.front();
        c[out/3+1]=c[out/3+1]+a;
        q.pop();

        q.push(0);
        a=q.front();
        c[out/3+1]=c[out/3+1]+a;
        q.pop();

        q.push(0);
        a=q.front();
        c[out/3+1]=c[out/3+1]+a;

```

```

        q.pop();

        q.push(0);
        a=q.front();
        c[out/3+1]=c[out/3+1]+a;
        q.pop();
        break;
    case 'F'://出局
    case 'G'://出局
    case 'S'://出局
        out++;//出局數
        if (out%3==0){
            q.push(0);
            q.push(0);
            q.push(0);
            q.pop();
            q.pop();
            q.pop();
        }
        if (out==outt)
            break;
        break;
    }//switch
    if (tnum1==tnum)//出場人數 題目給的總打擊人數
        break;
    if (out==outt)//出局數 題目給的出局人數
        break;
} //for i
if (tnum1==tnum) //出場人數 題目給的總打擊人數
    break;
if (out==outt) //出局數 題目給的出局人數
    break;
} //for j
int score=0 ; //總得分
//outt是題目要求的出局數
for (i=1;i<=outt/2;i++){//累加每局得分
    score=score+c[i];
}

```

```

    cout<<endl;
    cout<<score<<endl;
    return 0;
}

```

p425

x	0	1	2	3	4	5	6	7		
	0	2	4	6	8	10	12	14		
-1	l			m				h		
	l	m		h						
	l,m	h								
	l,h									
16	l			m				h		
					l	m		h		
							l,m	h		
								l,h		

輸入 16，傳回 14，但 14 沒有大於 16，所以選 D。

P427_1

F(2)印兩個星星，還要繼續 F(1)+F(1)，F(1)看上面，印一個星星，結束。所以是

最後一個 F(2)同上。所以是

也可以用以下方式解題。但只要看到到重複的數字，就直接抄，這樣才省時間。

P427_2

a1(5) =f(1)+f(4) =1+4+f(2)+f(2) =1+4+2+f(1)+f(1)+2+f(1)+f(1) =13	a2(5) =f(2)+f(3) //f(2)重複，抄上面 =4+3+f(1)+f(1) =9
--	--

$a1(13)$ $=f(2)+f(10)$ $=4+10+f(5)+f(5)$ $=14+2(5+f(2)+f(2))$ $=14+2(5+4+4)$ $=14+26$ $=40$	$a2(13)$ $=f(5)+f(7)$ $=13+7+f(3)+f(3)$ $=20+2*5$ $=30$
---	---

P432

1. 執行G(1)，因為 $K(0) = 1$ ，增加1。
2. 執行G(2)，因為 $K(0, 1) = 5$ ，增加4。
3. 執行G(3)，因為 $K(0, 1, 2) = 8$ ，增加3。
4. 執行G(4)，因為 $K(0, 1, 2, 3) = 10$ ，增加2。
5. 執行G(5)，因為 $K(0, 1, 2, 3, 4) = 11$ ，增加1，與上面1 重複，規律性已經出現，一輪增加10。

6. 所以 $G(100) = K(0, 1, 2, 3...99)$ ，因為

$(0+1+2+3) + (4+5+6+7) + \dots + (96+97+98+99) = 10*25$ （共 25 輪） $=250$ ，所以答案是

250.

P434

$$f(15)=15+f(76)$$

$$=15+76+f(38)$$

p435

17

此題與 105 第二次試題接近，前面兩個迴圈相同，同樣可推得

$$n(n+1)/2$$

其次，第 3 個迴圈

```
for (int k=1;k<=n;k=k*2)
```

```
    x=x+1
```

推導如下：

n	k	x
1	1	1
2	1,2	2
3	1,2	2
4	1,2,4	3
5	1,2,4	3
6	1,2,4	3
7	1,2,4	3
8	1,2,4,8	4
15	1,2,4,8	4
16	1,2,4,8,16	5
31	1,2,4,8,16	5
32	1,2,4,8,16,32	6

由上表可之 x 與 n 關係如下：

$$x=\log_2 n + 1$$

請鍵入以下程式，並觀察執行結果。(C++沒有 $\log_2 n$ 函數，所以要用換底公式)

P436

本例將 i,j 條列如下：(列是 i,行是 j)

右邊是 j 下面是 i	1	2	3	4	5
1	2	3	4		
2	3	4			
3	4				
4					

5					
---	--	--	--	--	--

累加(i+j)和是 2,3,4，所以 $2+3+4+3+4+4=20$

p441

程式列印

1、本題數字要能達到1000位，所以要以字串輸入。

```
#include <iostream>
```

```
#include <cstring> //使用stoi()
```

```
#include <cmath> //使用abs()
```

```
using namespace std;
```

```
int main() {
```

```
    string a,a1; // 本題數字要能達到1000位，所以要以字串輸入
```

```
    int c0=0,c1=0;
```

```
    int c;
```

```
    while (cin>>a){
```

```
        c0=0;c1=0; //每次要歸零
```

```
        for (int i=0;i<a.size();i++){
```

```
            a1=a[i];
```

```
            c=stoi(a1);
```

```
            if ((i % 2)==0)
```

```
                c0=c0+c;
```

```
            else
```

```
                c1=c1+c;
```

```
        }
```

```
        cout<<abs(c1-c0);
```

```
    }
```

```
    return 0;
```

```
}
```

實作

1、請將本題改為可輸入0到4294967295的整數。

2、這種數字分解，後續考很多次，例如，遞增數、完全奇數等。

P442

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int a[5000];
```

```

bool b[5000]; //記錄是否探訪
int i, j;
int n;
int num;
while (cin >> n) {
    for (i = 0; i <= n - 1; i++) {
        cin >> a[i];
        b[i] = 0; //未探訪
    }
    i = 0; num = 0;
    bool success = false;
    int ii;
    do {
        ii = i; //紀錄此小群體的頭
        num++; //小群體個數
        while (a[i] != ii & b[i] == 0) {
            cout << i << " " << a[i] << endl; //交卷前，記得刪除
            b[i] = 1; //探訪此群組
            i = a[i]; //繼續探訪此群組
        }
        //已經找到此群體的源頭
        cout << i << " " << a[i] << " " << num << endl; //交卷前，記得刪除
        b[i] = 1;
        success = true;
        //判斷是否全部探訪完畢
        //只要有一個未探訪，就未成功，且從其開始探訪
        for (i = 0; i <= n - 1; i++)
            if (b[i] == 0) {
                success = false;
                break;
            }
    } while (!success);
    cout << endl; //交卷前，記得刪除
    cout << num;
}
return 0;
}

```


p446_1

0, 1, 2, 2, 3, 3, 0, 0, 0, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 0, 0, 0, 0(請留意底線)

446-2

1, 2, 3, 3, 0, 0, 1, 1, 1, 2, 2, 2, 3, 3, 3, 3, 0, 0, 0, 0, 1, 1, 1, 1 , (請留意底線)

446-3

3、依照題目提示，

以 num=5 為例，a 陣列索引如下：

0, 0	0, 1	0, 2	0, 3	0, 4
1, 0	1, 1	1, 2	1, 3	1, 4
2, 0	2, 1	2, 2	2, 3	2, 4
3, 0	3, 1	3, 2	3, 3	3, 4
4, 0	4, 1	4, 2	4, 3	4, 4

題目說0是向左，向左向量是(0,-1)，1是向上，向上向量是(-1,0)，所以，定義dir方向向量如下：

```
int dir[4][2]={{0,-1},{-1,0},{0,1},{1,0}};
```

這樣方向d=0,1,2,3就會依續為向左、向上、向右、向下。本題與老鼠走迷宮有點類似，請先複習老鼠走迷宮程式。參考程式如下：

```
#include <iostream>
using namespace std;
int main() {
    int i,j;
    int num,n;
    int d;
    int a[51][51];
    while (cin >>num){ //陣列大小
        cin>> d;
        for (i=1 ;i<=num;i++){
            for ( j=1 ;j<=num;j++){
                cin>>a[i][j];
            }
        }
    }
```

```

int dir[4][2]={ {0,-1},{-1,0},{0,1},{1,0}};
//方向向量0,1,2,3，依序是向左、向上、向右、向下
n=1;
int x=num/2+1;
int y=num/2+1;
cout<<a[x][y];
for (i=1;i<=100;i++){
    for (j=1 ;j<=i;j++){
        x=x+dir[d][0];
        y=y+dir[d][1];
        cout<<a[x][y];
        n++;
        if (n>=num*num )//探訪完畢
            break;
    }
    if (n>=num*num )
        break;
    d=(d+1)%4;//方向
    for (j=1 ;j<=i;j++){
        x=x+dir[d][0];
        y=y+dir[d][1];
        cout<<a[x][y];
        n++;
        if (n>=num*num-1 )
            break;
    }
    d=(d+1)%4;//方向
    if (n>=num*num )
        break;
}
}
return 0;
}

```

P450

#include <iostream>

```

#include <algorithm>//使用 sort 方法
using namespace std;
int main() {
    int n;//服務點個數
    int k;//基地台個數
    int i,j;
    int a[50000];
    while(cin>>n>>k){
        for (i=0;i<=n-1;i++){
            cin>>a[i];//座標
        }
        sort(a,a+n);//使用 <algorithm>
        //for (i=0;i<=n-1;i++){
            //cout<<a[i]<<" ";
        //}
        int l;
        int kk;
        for (i=1;i<=(a[n-1]-a[0]);i++){ //直徑
            l=a[0];//每個基地台左界
            j=1;
            kk=1;//所需基地台個數
            do {
                while (a[j]-l<=i & j<=n-1) { //找出每個基地台可服務的點
                    j++;
                }
                if (j>n-1)//超出邊界
                    break;
                kk++;//所需基地台個數
                if (kk>k )//此一直徑所需基地台個數已經超過題目要求，提前
                    離開
                    break;
                l=a[j];//更新每個基地台左界
                j++; //尋求下一個
            } while (j<=n-1);
            //cout<<endl;
            cout<<i<<" "<<j<<" "<<kk<<",";//輸出看看
            if (kk==k)
                break;
        }
    }
}

```

```
    }  
    cout<<endl;  
    cout<<i<<endl;  
}  
return 0;  
}
```

P454

//a 陣列，b 每一次的結果，r 取 n,d 結果，e 個數

封面背面

特色增加

4、豐富的實作。程式設計並不是老師一直講，而是要能『依樣畫葫蘆』與『舉一反三』的作出題目，所以本書作者設計出很多與範例相近的『實作題目』，這樣學生才能享受解題的樂趣。

檔案結束，後面不管