



Yuan Ze University

$$WS = E_{\xi} \left[\min_{\tau \geq t} \xi \right] E_{\xi}(\tau | \xi, S)$$
$$A_t = \sum_{s=0}^{\infty} \beta^s \left(\frac{1}{\tau} \sum_{i=0}^{\tau-1} \xi_i \right) \sum_{j=0}^{\infty} \beta^j \left(\frac{1}{\tau} \sum_{i=0}^{\tau-1} \xi_i \right)$$

C++ 程式初探 III

2015暑期





C++ 程式 III – 大綱

1. 陣列
2. 副函式
3. 補充:變數範圍(Scope)與生命週期
4. 補充:條件運算子
5. 補充:中斷迴圈
6. 補充:數學函式庫
7. 檔案輸入/輸出



1 維陣列

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

int main(void)
{
    int a[10] = { 0 }; //設定初始值為0
    for (int i = 0; i<10; i++) {
        a[i] = i + 1;
    }
    system("PAUSE");
    return 0;
}
```

陣列型別	陣列名稱	陣列大小	;
int	array	[10]	;

0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---

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

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

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



2維陣列 – 宣告範例

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

```
int main(void)
{
```

```
    int array_a[2][3];    //設定大小，未初始化
```

```
    int array_b[2][3] = { //設定大小，並初始化
        { 1, 2, 3 },
        { 4, 5, 6 }
    };
```

```
    int array_c[][3] = { //設定大小(第1個維度留空)，並初始化
        { 1, 2, 3 },
        { 4, 5, 6 }
    };
```

```
    system("PAUSE");
    return 0;
```

```
}
```

陣列型別	陣列名稱	第一維大小	第二維大小;
int	array	[10]	[11] ;



2維陣列 – 宣告範例

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

```
int main(void)
{
```

```
    int array_d[2][3] = { 0 }; //初始化初值為0
```

```
    const int row = 2;
```

```
    const int col = 3;
```

```
    int array_e[row][col]; //陣列大小由常數值設定
```

```
    system( "PAUSE" );
```

```
    return 0;
```

```
}
```

陣列型別	陣列名稱	第一維大小	第二維大小;
int	array	[10]	[11] ;

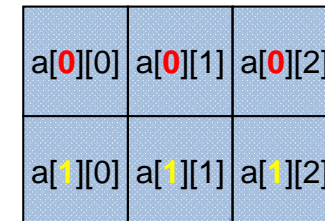


2維陣列 – 宣告範例

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

//記憶體配置

```
int main(void)
{
    // 一般初始化方法，採用兩層大括號
    int array_b[2][3] = {
        { 1, 2, 3 },
        { 4, 5, 6 }
    };
```

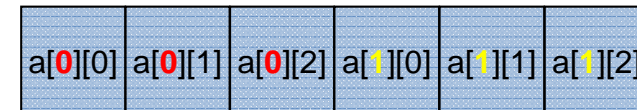


// 以1維方式初始化，僅用一層大括號

```
int array_b[2][3] = { 1, 2, 3, 4, 5, 6 };
```



```
system("PAUSE");
return 0;
```



```
}
```



2維陣列 – 操作範例

```
#include <iostream >
using namespace std;
int main(void)
{
    const int row = 5;
    const int col = 10;
    int array[row][col];
    for (int i = 0; i < row; i++)
        for (int j = 0; j < col; j++)
            array[i][j] = (i + 1) * (j + 1);
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++)
            cout << array[i][j] << "\t";
        cout << endl;
    }
    system("PAUSE");
    return 0;
}
```

```
C:\Users\WUKC\documents\visual studio 2013\Projects\c++_tutorial\Debug\c++_tutori...
1      2      3      4      5      6      7      8      9      10
2      4      6      8      10     12     14     16     18     20
3      6      9      12     15     18     21     24     27     30
4      8      12     16     20     24     28     32     36     40
5      10     15     20     25     30     35     40     45     50
請按任意鍵繼續 . . .
```

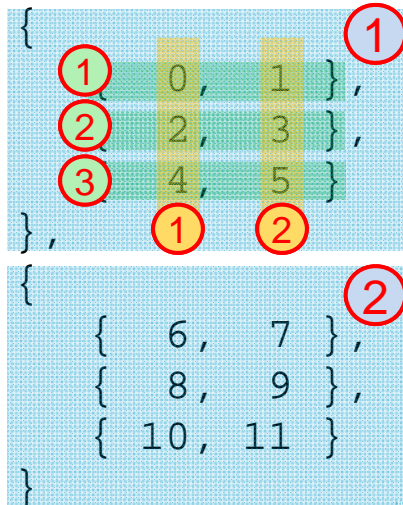


多維陣列

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

```
int main(void)
{
```

```
int ary3d1 [2][3][4]; //設定大小，未初始化
int ary3d2 [2][3][2] = { //初始化
```



```
};
system("PAUSE");
return 0;
```

}



多維陣列

```
int main(void)
{
    int ary3d2 [2][3][2] = {
        {
            { 0, 1 },
            { 2, 3 },
            { 4, 5 }
        },
        {
            { 6, 7 },
            { 8, 9 },
            { 10, 11 }
        }
    };
    int ary3d3 [2][3][2] = {
        { 0, 1, 2, 3, 4, 5,
          6, 7, 8, 9, 10, 11 };

    system("PAUSE");
    return 0;
}
```



練習 1 - 找最大值

```

#include <iostream>
using namespace std;

int main(void)
{
    const int n = 7;
    int a[n] = { 8, 4, 6, 9, 2, 3, 1 };
    int max = a[0]; //或-1e9 = 10-9
    for (int i = 1; i < n; i++) {
        if (a[i] > max) {
            max = a[i];
        }
    }
    cout << "max = " << max << endl;
    system("PAUSE");
    return 0;
}

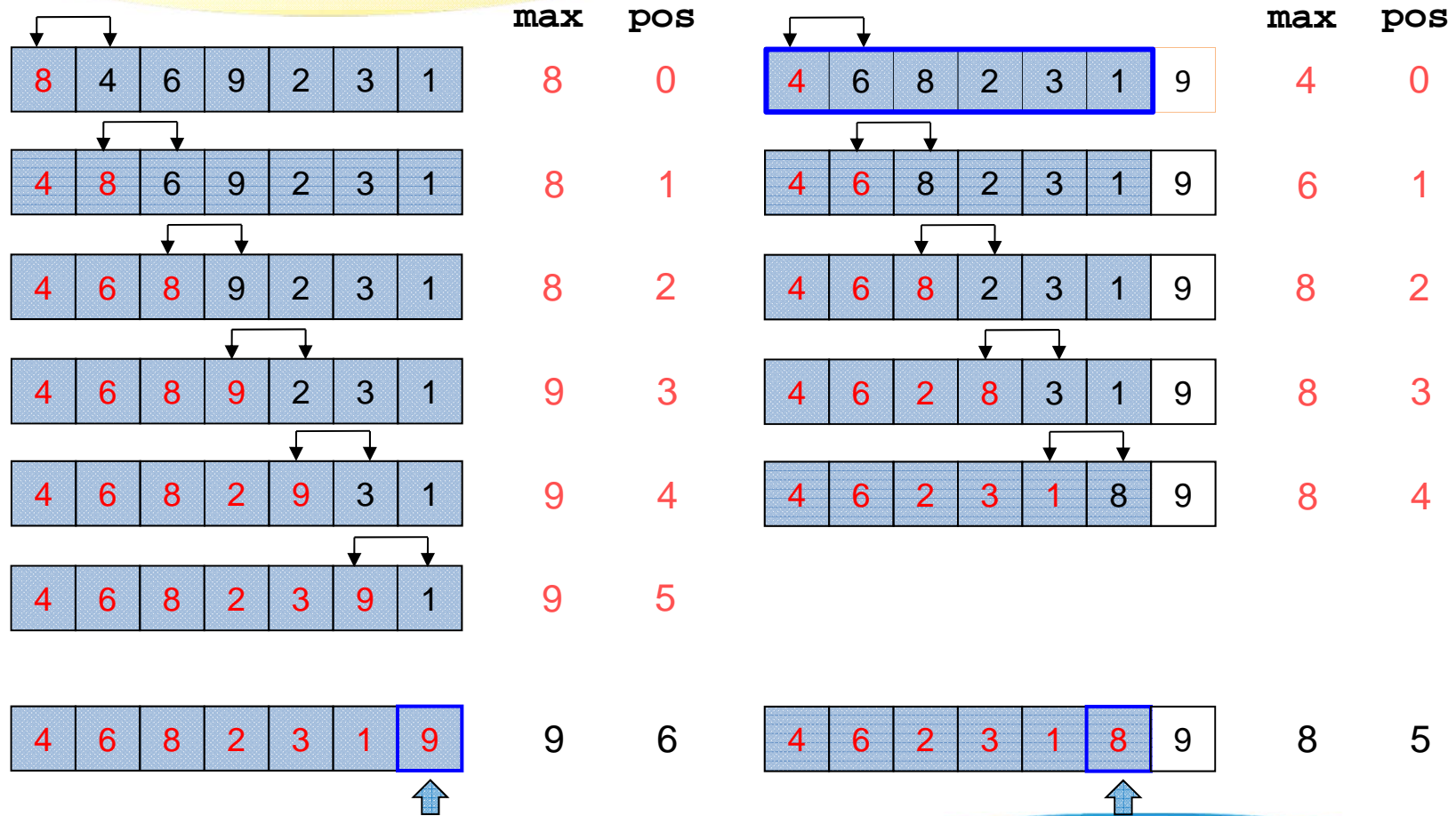
```

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

							max
8	4	6	9	2	3	1	8
8	4	6	9	2	3	1	8
8	4	6	9	2	3	1	8
8	4	6	9	2	3	1	9
8	4	6	9	2	3	1	9
8	4	6	9	2	3	1	9
8	4	6	9	2	3	1	9

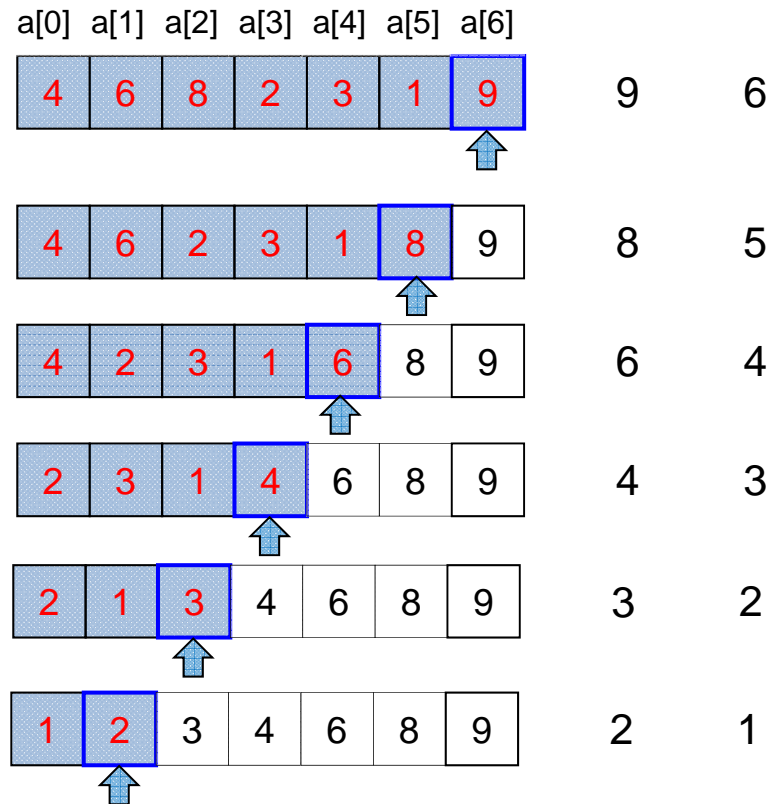


練習 2 - 氣泡排序法(1/2)





練習 2 - 氣泡排序法(2/2)





練習 3 – 移除重複數字

```
int a[10] = {1,1,1,3,3,4,5,5,5,6};
```

將上述已排序陣列中重複的元素移除，使得陣列Array中新的陣列元素為 {1,3,4,5,6,0,0,0,0,0};

1. 允許使用額外的陣列宣告
 - 宣告一個新的陣列b[10]，將array中不重複的元素儲存至b中。
2. 僅可使用同一陣列的空間
 - 不額外宣告b[10]，直接對array做判斷與運算。



練習 4 – 重新洗牌

```
int a[10] = {1,2,3,4,5,6,7,8,9,10};
```

如何將上述陣列隨機重新排序？



函式

回傳型別 函式名稱 (變數型態 變數名稱)
int fun (int input)

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

```
int pow2(int num) {  
    return num*num;  
}  
int pow(int n, int p) {  
    int r = 1;  
    for (int i = 0; i < p; i++)  
        r *= n;  
    return r;  
}
```

```
int main(void)  
{  
    // 接下一頁...
```



函式

// 續上一頁...

```
int main(void)
{
    int num = 0;
    int power = 0;
    cout << "輸入數值：";
    cin >> num;
    cout << "輸入次方項：";
    cin >> power;
    cout << num << " 平方：" << pow2(num) << endl;
    cout << num << " 的" << power << "次方：";
    cout << pow(num, power) << endl;

    system("PAUSE");
    return 0;
}
```




函式 – 陣列引數

```
int max(int n, int ary[]) {  
    int max = ary[0];  
    for (int i = 1; i < n; i++)  
        if (ary[i] > max)  
            max = ary[i];  
    return max;  
}
```

```
int main(){  
    int a[5] = { 5, 9, 8, 1, 6 };  
    int maxa = max(5, a);  
    cout << maxa << endl;  
    system("PAUSE");  
    return 0;  
}
```

回傳型別 函式名稱 (變數型態 陣列名稱 [[大小])

```
int fun (int ary[]) {  
    ...  
}  
  
int fun (int ary[][3]) {  
    ...  
}
```



函式 – 陣列引數

```
void max2last(int n, int ary[]) {
    int temp;
    for (int i = 0; i < n-1; i++)
        if (ary[i] > ary[i+1]) {
            temp = ary[i];
            ary[i] = ary[i+1];
            ary[i + 1] = temp;
        }
}

int main(){
    int a[5] = { 5, 9, 8, 1, 6 };
    max2last(5, a);
    for (int i = 0; i < 5; i++)
        cout << a[i] << " ";
    cout << endl;
    system("PAUSE");
    return 0;
}
```



函式 – 傳址 vs 傳值

```
void Swap(int &a, int &b)
{
    int temp = a;
    a = b;
    b = temp;
}
void max2last(int n, int ary[]) {
    int temp;
    for (int i = 0; i < n-1; i++)
        if (ary[i]>ary[i+1]) {
            Swap(ary[i], ary[i+1]);
        }
}
```



函式 – 傳址 vs 傳值

// 傳址呼叫的優點

```
void Swap(BigData &a, BigData &b)
{
    BigData temp = a;
    a = b;
    b = temp;
}
```

當函式的引數是一個很大的物件，傳址比傳值快。

// 指標也是異曲同工之效

```
void Swap(BigData *a, BigData *b)
{
    BigData temp = *a;
    *a = *b;
    *b = temp;
}
```



變數範圍(Scope)與生命週期

```
int n = 8;
int max(int n, int ary[]) {
    int max = ary[0];
    for (int i = 1; i < n; i++)
        if (ary[i] > max)
            max = ary[i];
    return max;
}

int main(){
    int a[5] = { 5, 9, 8, 1, 6 };
    int maxa = max(5, a);
    cout << maxa << endl;
    cout << n ;
    system("PAUSE");
    return 0;
}
```



變數範圍(Scope)與生命週期

```
int main(){
    for (int i = 0; i < 2; i++) {
        for (int i = 0; i < 2; i++) {
            cout << i << " ";
        }
        cout << endl;
    }
    cout << endl;
    system("PAUSE");
    return 0;
}
```



變數範圍(Scope)與生命週期

```
int main(){  
    for (int i = 0; i < 2; i++) {  
        cout << i << " ";  
    }  
    cout << endl;  
    for (int i = 0; i < 2; i++) {  
        cout << i << " ";  
    }  
    cout << endl;  
    cout << i << endl;  
  
    system("PAUSE");  
    return 0;  
}
```

error C2065: 'i' : 未宣告的識別項



練習 6

- ✓ 練習將氣泡排序法轉為函式。
 - ▶ 將氣泡排序程式放置在命名為BubbleSort的函式中。
 - ▶ 在main()函式中利用rand()隨機產生大小為10的陣列。
 - ▶ 在main()函式中呼叫BubbleSort函式進行排序。
 - ▶ 將排序結果輸出。



補充語法 條件運算子?:

```
int max(int a, int b) {  
    if (a > b)  
        return a;  
    else  
        return b;  
}
```

```
int max(int a, int b) {  
    return a > b ? a : b;  
}
```

```
void max(int a, int b) {  
    cout << (a > b) ? "a" : "b";  
}
```

函數

判斷式 ? 成立時為 : 不成立時為
(a>b) ? a : b

變數型態需相等 (或可隱式轉型)

```
int max = a>b ? a : "b" ;
```



補充語法 break, continue

```
//99乘法表
for(int i=2; i<=9; i++) {
    for(int j=2; j<=9; j++) {
        cout<<i<<"*"<<j<<"="<<setw(3)<<left<<i*j;
        if(j==4)
            break;
    }
    cout<<endl;
}
```

```
for(int i=2; i<=9; i++) {
    for(int j=2; j<=9; j++) {
        cout<<i<<"*"<<j<<"="<<setw(3)<<left<<i*j;
        if(i==4)
            break;
    }
    cout<<endl;
}
```

break

跳出一個迴圈

```
C:\Program Files\Borland\CBuilde
2*2=4 2*3=6 2*4=8
3*2=6 3*3=9 3*4=12
4*2=8 4*3=12 4*4=16
5*2=10 5*3=15 5*4=20
6*2=12 6*3=18 6*4=24
7*2=14 7*3=21 7*4=28
8*2=16 8*3=24 8*4=32
9*2=18 9*3=27 9*4=36
```

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



補充語法 break, continue

//99乘法表

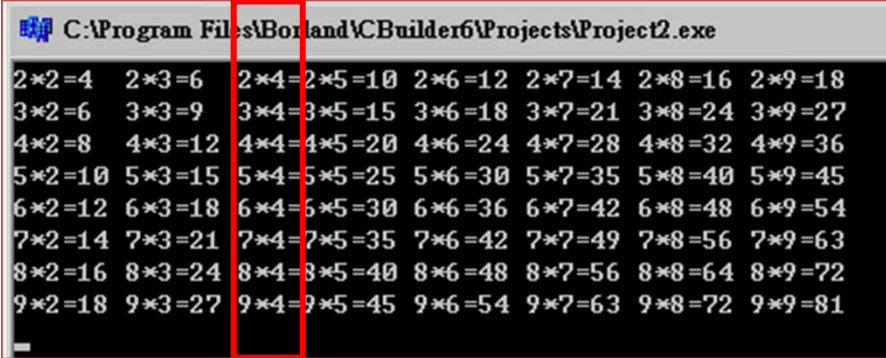
```

for(int i=2; i<=9; i++) {
    for(int j=2; j<=9; j++) {
        cout << i << "*" << j << "=" ;
        if(j==4)
            continue;
        cout << setw(3) << left << i*j;
    }
    cout<<endl;
}

```

continue

略過此次迴圈內容





補充語法 數學函式庫

`#include <cmath>`

方法	說明	方法	說明
<code>ceil(x)</code>	最小大於等於x的整數	<code>pow(x, y)</code>	x的y次方
<code>floor(x)</code>	最大小於等於x的整數	<code>sqrt(x)</code>	x開根號
<code>fabs(x)</code>	絕對值	<code>tan(x)</code>	tangent of x
<code>exp(x)</code>	指數e的x次方	<code>sin(x)</code>	sine of x
<code>log(x)</code>	x的自然對數(e為底)	<code>cos(x)</code>	cosine of x
<code>log10(x)</code>	x的對數(10為底)	<code>fmod(x, y)</code>	浮點數的x/y餘數



練習 7-多維陣列之排序

假設有某班共有8位學生參加國文與英文考試，考試成績如下述之2維矩陣，第一列為國文成績，第二列為英文成績。

```
int student[2][8] = {{68, 64, 91, 73, 55, 89, 23, 74},  
                    {67, 97, 54, 60, 34, 74, 77, 46}};
```

1. 請問如何將學生依國文成績排序?

```
Result: {{91, 89, 74, 73, 68, 64, 55, 23},  
        {54, 74, 46, 60, 67, 97, 34, 77}};
```

2. 若將學生依英文成績排序?

```
Result: {{64, 23, 89, 68, 73, 91, 74, 55},  
        {97, 77, 74, 67, 60, 54, 46, 34}};
```



檔案輸出

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

```
//主函式
int main(void)
{
    int num = 6;
    //設定物件名稱及路徑
    ofstream fout("result.txt");
    fout << "number = ";
    fout << num;
    fout << endl;

    system("PAUSE");
    return 0;
}
```

函數標頭檔
fstream

輸出類別 變數名稱 (檔名字串);
ofstream fout("fname.txt");

變數名稱 運算子 變數名稱 ;
fout << var ;



檔案輸入

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

//主函式
int main(void)
{
    int num;
    //設定讀檔物件名稱及路徑
    ifstream fin("result.txt");

    fin >> num;

    system("PAUSE");
    return 0;
}
```

函數標頭檔
fstream

輸入類別 變數名稱 (檔名字串);
ifstream fin("fname.txt");

變數名稱 運算子 變數名稱 ;
fin >> var ;

練習 8

- ✓ 若有一檔案 `position.txt`，其內容格式如右圖所示。
- ✓ 其中每一列有兩個數字，分別代表其位置 (x, y) ，請讀取 `position.txt` 檔案資料，並計算每個座標點的距離矩陣(即任兩點間的距離，如下圖)，並且輸出至螢幕上與 `DistMatrix.txt` 檔中。

```
Position.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
1.78    2.91
4.77    2.37
0.11    4.2
3.32    7.64
8.23    9.79
```

```
C:\Users\WUKC\documents\visual studio 2010\Proj...
x[] = 1.78  4.77  0.11  3.32  8.23
y[] = 2.91  2.37  4.2   7.46  9.79

DistMatrix =
0.0000  3.0384  2.1102  4.8036  9.4306
3.0384  0.0000  5.0064  5.2925  8.1871
2.1102  5.0064  0.0000  4.5751  9.8581
4.8036  5.2925  4.5751  0.0000  5.4348
9.4306  8.1871  9.8581  5.4348  0.0000

請按任意鍵繼續 . . .
```

```
DistMatrix.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
0.000  3.038  2.110  4.974  9.431
3.038  0.000  5.006  5.466  8.187
2.110  5.006  0.000  4.705  9.858
4.974  5.466  4.705  0.000  5.360
9.431  8.187  9.858  5.360  0.000
```




檔案輸出 - CSV format

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

//主函式
int main(void)
{
    ofstream fout("result.csv");

    fout << 1 << ", " << 2 << ", " << 3 << ", " << 4;
    system("PAUSE");
    return 0;
}
```



補充

讀取 CSV format 檔案

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

int main(void)
{
    ifstream fin("result.csv");
    string fline;
    while (getline(fin, fline)){
        stringstream fss(fline);
        string s;
        while (getline(fss, s, ',')) {
            int g = atoi(s.c_str());
            cout << g << endl;
        }
    }

    system("PAUSE");
    return 0;
}
```



作業 - 輪盤法

同上次射飛鏢問題，但此次標靶有20個大小不同面積。

1. 假設此20個大小不同的面積記錄於陣列內(例如int area[20])。
2. 設計一個函式，每次呼叫會回傳射到面積index。