

1

Lecture 08

多維陣列
函式

2

Outline

- 二維與多維陣列
- 標準函式庫
- 函式簡介

3

二維陣列與多維陣列

4

一維陣列

- 陣列為多個**同一型態**變數之組合
- `int a[10];`
 - 可存放 10 個整數資料的陣列，可視為 10 個變數排成一個隊伍，並給每一個變數一個編號，自 0 號開始、至 9 號為止。
 - `a[0], a[1], a[2], a[3], ..., a[9]`

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

5

二維陣列的宣告

- 陣列為多個**同一型態**變數之組合
- `int A[3][10];`
 - 可存放 30 個整數資料的二維陣列，可視為 30 個變數排成 3 個隊伍 (3列)，並給每一個變數兩個編號，一個編號指定在第幾個隊伍、另一個編號指定在隊伍內的順序。
 - `A[0][0], A[0][1], A[0][2], ..., A[0][9], A[1][0], A[1][1], ...`

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

6

二維陣列 (Array) 的使用

- `int cars[3][6];`
 - 宣告一整數陣列，可存放3列6行共 18 個整數型別的資料
- 使用
 - `cars[0][1] = 3;` → 將 3 存入 cars 陣列中的第 0 列第1行的元素
 - `cars[2][0]++;` → 將 cars 陣列中的第2列第0行的元素增加 1
 - `cars[1][0]=cars[1][1]+a;` → 將cars陣列中的第1列第1行的值取出，與變數 a 作加法運算後，將結果存入第1列第0行的元素。
- 為什麼有二維陣列或多維陣列？
 - 對某些運算來說很自然 (e.g. 行列式、矩陣) ...

7

8-1.cpp

```
int a[3][5];

for(int i=0;i<3;i++) {
    for(int j=0;j<5;j++) {
        a[i][j] = i*10 + j;
    }
}

for(int i=0;i<3;i++) {
    for(int j=0;j<5;j++) {
        cout << a[i][j] << " ";
    }
    cout << "\n";
}
```

0	1	2	3	4
10	11	12	13	14
20	21	22	23	24

8

二維陣列的初始化

- int a[3][4] = {{1,2,3,4}, {5,6,7,8}, {9,10,11,12}};

A[0][0]	A[0][1]	A[0][2]	A[0][3]
A[1][0]	A[1][1]	A[1][2]	A[1][3]
A[2][0]	A[2][1]	A[2][2]	A[2][3]

1	2	3	4
5	6	7	8
9	10	11	12

9

請想想以下陣列初始化後之內容為?

- int a[3][4] = {1,2,3,4};
- int a[3][4] = {{1,2,3,4}, {8, 1}, {4, 2}};
- int a[3][4] = {{0}, {1,2,3,4}, {5,1,2,3}};
- int a[3][4] = {1,2,3,4,5,6,7,8};
- int a[][4] = {{1, 2}, {3, 4, 5}};

1 2 3 4	1 2 3 4	0 0 0 0	1 2 3 4	1 2 0 0
0 0 0 0	8 1 0 0	1 2 3 4	5 6 7 8	3 4 5 0
0 0 0 0	4 2 0 0	5 1 2 3	0 0 0 0	

在陣列有給初始值時，且陣列大小被省略時，編譯器會根據初始值的個數來決定陣列大小。但是只有第一個維度可以省略! double A[][4], b[], c[][10][40];

10

矩陣向量相乘

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 3 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \times 4 + 2 \times 3 + 3 \times 2 + 4 \times 1 \\ 1 \times 4 + 0 \times 3 + 1 \times 2 + 0 \times 1 \\ 2 \times 4 + 1 \times 3 + 0 \times 2 + 1 \times 1 \\ 0 \times 4 + 1 \times 3 + 0 \times 2 + 1 \times 1 \end{bmatrix}$$

$$c_i = \sum_{j=1}^4 a_{ij} b_j \quad c_1 = a_{11}b_1 + a_{12}b_2 + a_{13}b_3 + a_{14}b_4$$

11

8-2.cpp

```
int a[4][4] = {
    {1,2,3,4},
    {1,0,1,0},
    {2,1,0,1},
    {0,1,0,1}
};
int b[] = {4,3,2,1}, c[4]={0};
```

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

12

8-2.cpp

```
for(int i=0;i<4;i++) {
    for(int j=0;j<4;j++) {
        c[i] += a[i][j] * b[j];
    }
}

for(int i=0;i<4;i++) {
    cout << c[i] << " ";
}
```

$$c_i = \sum_{j=1}^4 a_{ij} b_j \quad c_1 = a_{11}b_1 + a_{12}b_2 + a_{13}b_3 + a_{14}b_4$$

多維陣列

- 二維以上陣列 (稱為多維陣列) 在實務上應用比較少。
- 宣告
 - `int a[10][5][3];`
 - 宣告一三維陣列，第一維度有10個、第二維度5個、第三維度3個。→ 共可儲存150個不同的整數值
 - 可想成我們宣告了10個不同的5列3行之二維陣列
 - `float q[100][200][10][5];`
 - 宣告一個四維陣列，共可儲存100 x 200 x 10 x 5個不同的浮點數。
 - 可想成我們宣告了一個100列200行之二維陣列，而此二維陣列中每個元素又是一個10列5行之二維陣列。

C-字串陣列 / 字元的二維陣列

回顧

- 字串 = 字元的一維陣列
 - `char a[10] = "ABCDEFGHI";`
- `cin` 可以從鍵盤讀取字串
 - `char a[10];`
 - `cin >> a;`
- `cout` 可以將 C字串輸出
 - `char a[]="Hello";`
 - `cout << a;`

8-3.cpp

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

```
int main() {
    char msg[200] = "This is a test.";
    cout << msg;

    cout << "\n請輸入一串字: ";
    cin >> msg;
    cout << msg;

    return 0;
}
```

```
This is a test.
請輸入一串字: Hello Kitty
Hello
```

`cin` 利用空白或是換行符號來分隔不同資料，所以只能藉此得到不含空白的字串。

二維字元陣列 / 字串陣列

- `char a[][6] = {"Pig"}, {"Tiger"}, {"Dog"};`
- `char a[3][6] = {"Pig"}, {"Tiger"}, {"Dog"};`
- `char [3][6] =`
`{{'P', 'i', 'g', 0}, {'T', 'i', 'g', 'e', 'r', 0},`
`{'D', 'o', 'g', 0}};`

'P'	'i'	'g'	0		
'T'	'i'	'g'	'e'	'r'	0
'D'	'o'	'g'	0		

8-4.cpp

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

```
int main() {
    char a[][6] = {"Pig"}, {"Tiger"}, {"Dog"};
    int i;

    for(i=0;i<3;i++) {
        cout << a[i] << endl;
    }

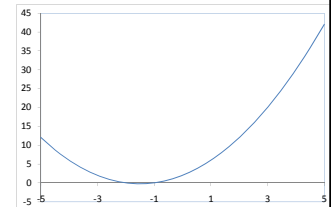
    return 0;
}
```

```
Pig
Tiger
Dog
```

函式簡介

數學函數 mathematical function

- 數學上有所謂的函數 (function)
 - $y = f(x) \rightarrow x$ 為自變數, y 為應變數
- 舉例
 - $y = f(x) = x^2 + 3x + 2$
 - $y = \sin(\theta)$



- 數學函式指定一些運算使用的參數，並計算出一些結果。

函式簡介

- 在電腦程式上，函式 (function) 為獨立的程式運作單元 (unit)，或稱為模組 (module)、副程式 (subroutine)、程序 (procedure)、...
- 函式常用來包裝複雜的程式碼，使主程式看起來較為簡單易懂
- 函式是用來創造「黑盒子」(black box) – 只看得見進去盒子的東西，和出來盒子的東西，盒子裡怎麼運作是看不到、也不想看到的 ...
 $y = \sin(\theta)$
- C/C++ 所提供的指令很少
 - 基本算術運算(+,-,*,/,%)、選擇性述敘、迴圈述敘, ...
 - 許多功能藉由函式所提供與擴充 (如三角函數, 字串處理, ...)
- 當需要使用函式時，會需要兩個要件：
 - 函式的原型宣告 (prototype) \rightarrow 由標頭檔 (header file) 取得
 - 函式的定義 (definition) \rightarrow 由函式庫 (library) 取得

1-1.cpp

```
#include <iostream>
using namespace std;
int main() {
    // 輸出 "Hello World" 至螢幕上
    cout << "Hello World";
    /* 函式執行完畢，回傳0 */
    return 0;
}
```

將 iostream 這個標頭檔內含有以定義標準輸入輸出之類別、物件、與函式等等的宣告。

標準函式庫裡的東西都定義在 std 這個命名空間裡，用來避免名稱的衝突

使用標準程式庫的 cout 物件輸出

函式的來源

- C/C++ 標準函式庫 (standard library)
 - 為程式語言認為必要提供的基本功能
- 自訂函式
- 第三方 (third party) 免費的或商業的程式庫

C/C++ 標準函式庫

C/C++ Standard Library

數學函式庫

- **#include <cmath>**
- 注意，在電腦裡的三角函數以弧度為單位，而非我們一般使用的角度。
 - 角度：以 360 度為一個圓
 - 弧度：以 2π 為一個圓
 - 角度轉弧度
 - 角度 * $\pi / 180$
 - 弧度換角度
 - 弧度 * $180 / \pi$

類別		
三角函數	sin	asin
	cos	acos
	tan	atan
	sinh, cosh, tanh	
指數與對數	exp	log
		log10
次方與根號	pow	sqrt
進位與捨位	ceil	floor
取絕對值	fabs, ...	

<http://www.cplusplus.com/reference/cmath/>

8-5.cpp

```
#include <iostream>
#include <cmath>
using namespace std;
int main() {
    double theta;
    double rad;
    double coeff = 3.14159265358979323846 / 180.0;
    for(theta=0; theta<=90; theta=theta+10) {
        rad = theta * coeff;
        cout << theta << " : " << sin(rad);
        cout << ", " << cos(rad) << endl;
    }
    return 0;
}
```

程式執行結果

```
0: 0, 1
10: 0.173648, 0.984808
20: 0.34202, 0.939693
30: 0.5, 0.866025
40: 0.642788, 0.766044
50: 0.766044, 0.642788
60: 0.866025, 0.5
70: 0.939693, 0.34202
80: 0.984808, 0.173648
90: 1, 6.12323e-017
```

8-6.cpp

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

int main() {
    double a;
    a = pow(2,3); // 計算 2 的 3 次方
    cout << "2^3 = " << a;
    cout << "\nlog10(2) = " << log10(2);
    cout << "\nlog10(2^3) = " << log10(a);
    cout << "\nsqrt(3.0) = " << sqrt(3.0);
    cout << "\npow(3.0, 0.5) = " << pow(3.0, 0.5);
    return 0;
}
```

程式執行結果

```
2^3 = 8
log10(2) = 0.30103
log10(2^3) = 0.90309
sqrt(3.0) = 1.73205
pow(3.0, 0.5) = 1.73205
```

8-7.cpp

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

int main() {
    double a;
    a = log10(3.0);
    cout << "a = log10(3.0) = " << a << endl;
    cout << "pow(10.0, a) = " << pow(10.0, a) << endl;
    cout << "exp(1) = " << exp(1.0) << endl;
    cout << "log(exp(1)) = " << log(exp(1.0)) << endl;

    return 0;
}
```

程式執行結果

```
a = log10(3.0) = 0.477121
pow(10.0, a) = 3
exp(1)=2.71828
log(exp(1)) = 1
```

8-8.cpp

```
#include <iostream>
#include <cmath>
using namespace std;
int main() {
    double a = sqrt(2.0);
    cout << "a=sqrt(2) = " << a << endl;
    cout << "floor(a) = " << floor(a) << endl;
    cout << "ceil(a) = " << ceil(a) << endl;
    a=-a;
    cout << "a=-sqrt(2) = " << a << endl;
    cout << "floor(a) = " << floor(a) << endl;
    cout << "ceil(a) = " << ceil(a) << endl;
    cout << "fabs(a) = " << fabs(a) << endl;
    return 0;
}
```

```
a=sqrt(2) = 1.41421
floor(a) = 1
ceil(a) = 2
a=-sqrt(2) = -1.41421
floor(a) = -2
ceil(a) = -1
fabs(a) = 1.41421
```

floor: 傳回最接近且小於傳入參數的整數
 ceil: 傳回最接近且大於傳入參數的整數
 fabs: 取絕對值

C 標準函式庫

- **#include <cstdlib>**

分類	函式名稱	說明
C 字串轉換	atoi, strtod	字串轉浮點數
	atol	字串轉整數
	atol, strtol	字串轉長整數
亂數	srand	設定亂數產生的種子
	rand	產生一亂數
整數相關	abs, labs	取整數的絕對值
	bsearch	在陣列中找尋資料
進階函式	qsort	對陣列中的資料排序
重要常數	RAND_MAX	亂數的最大值

<http://www.cplusplus.com/reference/cstdlib/>

31

8-9.cpp

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main() {
    int i, seed;
    cout << "請輸入亂數種子: ";
    cin >> seed;
    srand(seed);
    for(i=0;i<10;i++) {
        cout << rand() << ", " << rand()%6 << endl;
    }
    return 0;
}
```

請輸入亂數種子: 1	請輸入亂數種子: 2
18467.5	29216.3
26500.4	17795.0
15724.5	19650.0
29358.0	26431.4
24464.4	18316.1
28145.5	28189.1
16827.1	606.4
491.1	17829.5
11942.1	30367.5
5436.3	28961.4

32

8-10.cpp

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;
int main() {
    int i;
    srand(time(0));
    cout << endl;
    for(i=0;i<10;i++) {
        cout << ((double) rand())/RAND_MAX*6-2 << endl;
    }
    return 0;
}
```

2.71017	2.72756
0.730186	3.80499
-1.84802	3.01193
-1.45836	0.319285
1.0997	1.87078
-1.83172	1.0195
1.96655	0.493789
-1.13425	1.32182
0.368908	-1.95367
1.00778	-1.6389

33

C 字串函式庫

- #include <cstring>

函式名稱	說明
strncpy	字串拷貝
strncat	字串串接
stricmp	傳回字串比較結果
strlen	傳回字串長度
strchr	在字串中找尋一字元,並傳回其位置
strrchr	從字串尾端開始往回找尋一字元,並傳回其位置
strstr	在字串中找尋一子串,並傳回其起始位置
strtok	將字串依指定字元符號作切割

<http://www.cplusplus.com/reference/cstring/>

34

作業六