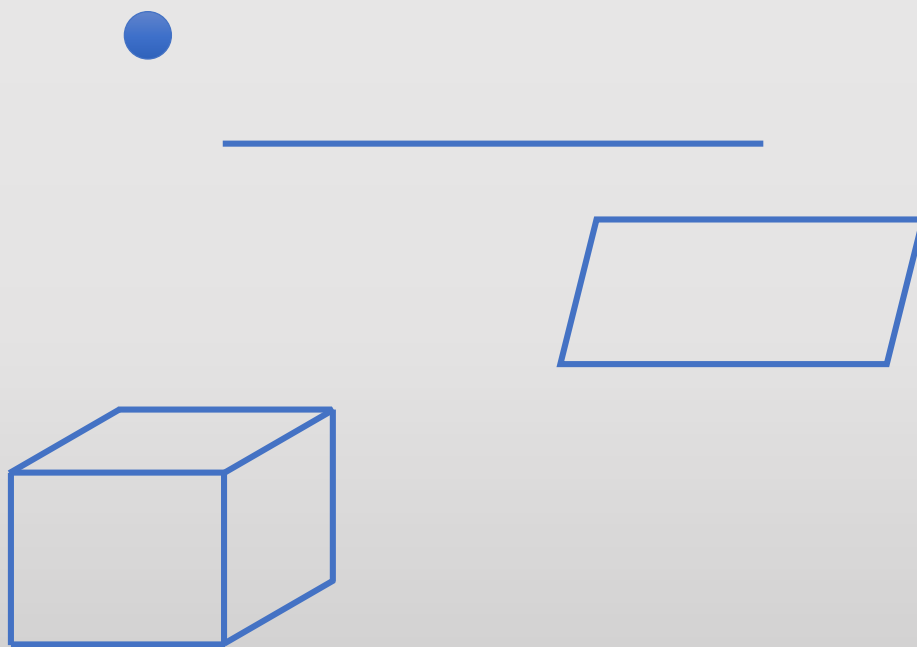


多維陣列

Sprout 2019

維度?

- 0維：點
- 1維：線
- 2維：面
- 3維：體積
-
- n維?
- 維度，是數學中獨立參數的數目 (by wiki)



對應到資料的儲存...

- 0維：單變數 m, n
- 1維：一維陣列 (上禮拜教的東西)

a

1	4	9	16	25
a[0]	a[1]	a[2]	a[3]	a[4]

- 2維?
- 3維?
-
- n維?

a

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15

複習:一維陣列

陣列宣告

- 不建議用變數指定大小

```
// 型態 陣列名[陣列大小(可用變數個數)];  
int a[100];  
int a[100] = {1, 2, 3}; // 沒指定滿大小，後面會自動補零  
bool b[] = {1, 0, 1}; // 沒指定大小，會自動計算大小
```

二維陣列宣告

- 型態 名稱 [大小m] [大小n]

```
int a[3][5];  
int b[3][5]={ {1,2,3,4,5}, {2,4,6,8,10}, {3,6,9,12,15} };  
int c[3][5]={ {1,2,3}, {2} };  
int d[3][5]={0};
```

- 有m個東西，每個有個n元素
- 什麼都不指定→垃圾值
- 會自動補0
- 也可以自動幫你算，最後一個一定要給明確大小

- 會自動補0

```
int c[3][5]={1,2,3},{2}};
```

```
int c[3][5]={1,2,3,0,0},{2,0,0,0,0},{0,0,0,0,0}};
```

- 也可以自動幫你算，最後一個一定要給明確大小

```
int c[][5]={1,2,3},{2}};
```

```
int d[][5]={1,2,3,0,0},{2,0,0,0,0},{0,0,0,0,0}};
```

↑相當於 c[2][5] 和 d[3][5]

複習:一維陣列

陣列存取

- 索引值從0開始算，只能放整數
- 索引值可以是變數

```
// 陣列名[索引值];  
int a[100]; //可以用 a[0] ~ a[99]  
std::cout << a[0];  
std::cin >> a[1];  
a[2] = a[2] * 2;  
  
int i = 3;  
a[i] = i;  
a[a[i]] = a[i];
```

二維陣列存取

```
int b[3][5]={ {1,2,3,4,5},  
              {2,4,6,8,10},{3,6,9,12,15}};
```

b[0][0]	b[0][1]	b[0][2]	b[0][3]	b[0][4]
b[1][0]	b[1][1]	b[1][2]	b[1][3]	b[1][4]
b[2][0]	b[2][1]	b[2][2]	b[2][3]	b[2][4]

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15

二維陣列存取

```
int b[3][5]={ {1,2,3,4,5},  
              {2,4,6,8,10},{3,6,9,12,15}};
```

- 一樣從0開始算，
索引值只能是整數

```
b[0][1]=50;  
b[2][4]=100;
```

b[0][0]	b[0][1]	b[0][2]	b[0][3]	b[0][4]
b[1][0]	b[1][1]	b[1][2]	b[1][3]	b[1][4]
b[2][0]	b[2][1]	b[2][2]	b[2][3]	b[2][4]

1	50	3	4	5
2	4	6	8	10
3	6	9	12	100

二維陣列存取

```
int b[3][5]={ {1,2,3,4,5},  
              {2,4,6,8,10},{3,6,9,12,15}};
```

- 也可以用變數當索引

```
int i=1,j=0;  
b[i][j]=0;  
b[ b[i][j] ][ b[i][j] ]=87;
```

- 小心存取的範圍
(不然會有吃不完的RE)

b[0][0]	b[0][1]	b[0][2]	b[0][3]	b[0][4]
b[1][0]	b[1][1]	b[1][2]	b[1][3]	b[1][4]
b[2][0]	b[2][1]	b[2][2]	b[2][3]	b[2][4]

87	50	3	4	5
0	4	6	8	10
3	6	9	12	100

多維陣列

- 型態 名稱[大小A][大小B][大小C][大小D].....
- N維就N個 []

```
int array[5][5][5][5][5]={0};
```

- 一樣用索引值指定存取位置

```
array[0][1][2][3][4]=2019;  
std::cout<<array[1][1][1][1][1];
```

練習-1419 矩陣翻轉

題目備份

- 給你一個 $N * M$ 的矩陣，輸出它的轉置
- 輸入：
兩個整數 $N, M (1 \leq N, M \leq 1000)$
接下來 N 行，每行有 M 個數字 (int)，中間以空白隔開
- 輸出：
轉置
 N 行，每行有 M 個數字 (int)，中間以空白隔開

圖示

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



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