

# 二维数组

夏秦

# 定义方法

## ▶ 语句格式

<类型> <数组名>[<常量表达式1>][<常量表达式2>;

## ▶ 例如

- int score[60][3]
- const int N=10,M=3;  
double point[N][M];



行标列标不能是变量

# 初始化方法

## ▶ 语句格式

<类型> <数组名>[<行数>][<列数>]={<表达式1>,<表达式2>,...};

## ▶ 初始化方式

### ◦ 全部元素

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

### ◦ 部分元素

- `int B[3][3]={1,2,3};`
- `int B[3][3]={{1,2},{4,5},{7,8}};`

# 使用方法

## ▶ 语句格式

<数组名>[<行标>][<列标>]



下标不能越界

## ▶ 例如

- `int A[2][2];`
- `A[0][0]=2;`
- `A[1][1]=1;`

# 问题

---

如果使用一维数组 $a[M*N]$ 和二维数组 $b[M][N]$ 表示同一矩阵，则 $b[i][j]$ 与 $a$ 中哪个下标的元素对应？

# 解决方法

1	2	3
4	5	6

$b[0][0] == a[0]$

$b[0][1] == a[1]$

$b[0][2] == a[2]$

$b[1][0] == a[3]$

$b[1][1] == a[4]$

$b[1][2] == a[5]$

$b[i][j] = a[i*N+j]$

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

```
{    const int m=2,n=3;
    int a[m*n],b[m][n];
    int i,j,x=1;
    for(i=0;i<6;i++,x++)
        a[i]=x;
    x=1;
    for(i=0;i<2;i++)
    {    for(j=0;j<3;j++,x++)
        {b[i][j]=x;
        cout<<a[i*n+j]<<"\t"<<b[i][j]<<endl;}
    }
    return 0;}
```

# 例：月份转换

---

编写程序，将用户输入的阿拉伯数字转换成对应月份的英文单词。

# 月份转换的程序

```
#include <iostream>
using namespace std;
int main()
{ char month[12][10]={"January", "February", "March", "April", "May", "June",
"July", "August", "September", "October", "November", "December"};
  int m;
  cin>>m;
  if(m>0&&m<13)
    cout<<month[m-1];
  else
    cout<<"The month is wrong";
  return 0;
```



# 本例学到

---

- ▶ 1.二维数组定义和初始化

```
char month[13][12]={"", "January", "February", "March", "April",  
"May", "June", "July", "August", "September", "October",  
"November", "December"};
```

- ▶ 2.二维数组使用

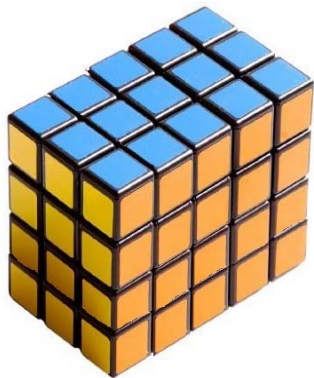
```
cout<<month[m]<<endl;
```

# 多维数组

---

## ► 例如

- `int A[5][4][3]`
- `double B[3][3][3]`



# 问题

---

- ▶ 还有什么算法可以实现月份转化程序？
- ▶ 如果要将数字转换为星期的英文单词，该如何编写程序。
- ▶ 如果用一维数组 $a[X*Y*Z]$ 和三维数组 $b[X][Y][Z]$  表示同一个空间结构，则 $b[i][j][k]$ 对应于数组 $a$ 哪个位置的元素？