集训课-10：二维数组

课程目标

01 二维数组定义

02 二维数组初始化

03 二维数组元素获取

04 二维数组输入输出

05 总结回顾

Part 1

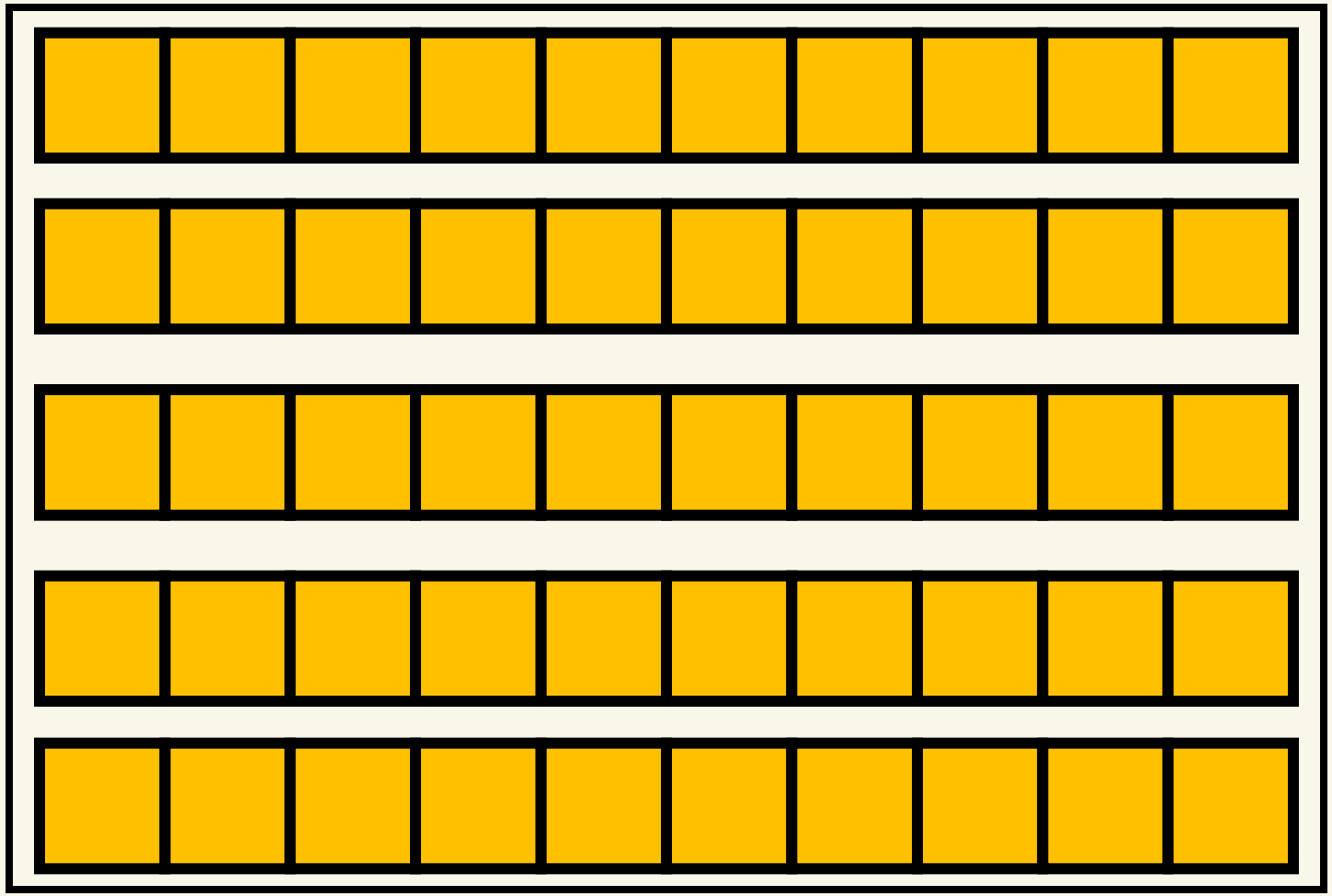
二维数组定义

二维数组定义

二维数组：相同数据类型的一维数组的集合

二维数组定义

多个一维数组整体保存



二维数组定义

二维数组定义格式：

数据类型 数组名[行数][列数];

二维数组定义

数据类型 数组名[行数][列数];

行\列	0	1	2	3	4
0					
1					
2					
3					

例如：int a[4][5];

二维数组定义

小明班级的同学排队去体检身高(身高都是小数类型)，排成了5行7列，现在想要使用数组存储每个人的身高情况，那么数组应该如何定义呢？

A、double a[5][7];

B、int a[5][7];

C、double a[4][6];

Part 2

二维数组初始化

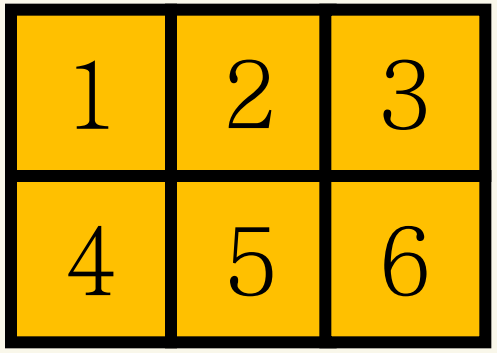
二维数组初始化

依次初始化每个元素

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

或

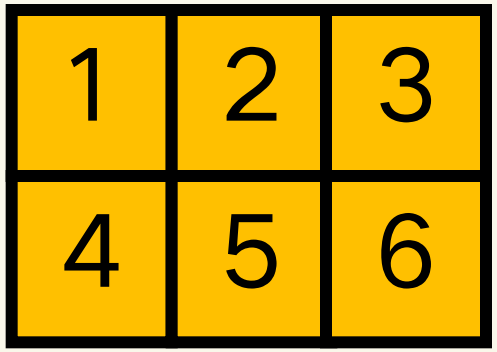
```
int a[2][3] = {{1,2,3},
               {4,5,6}};
```



二维数组初始化

依次初始化每个元素

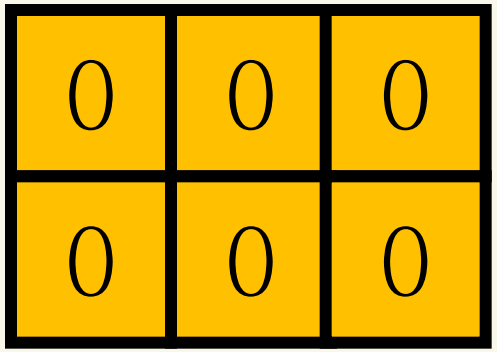
```
int a[2][3] = {1,2,3,4,5,6};
```



二维数组初始化

二维数组整体初始化为0

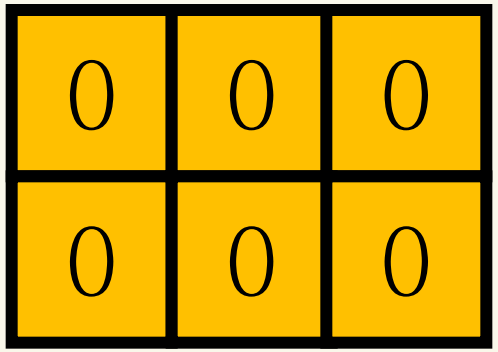
```
int a[2][3] = {};
```



二维数组初始化

二维数组整体初始化为0

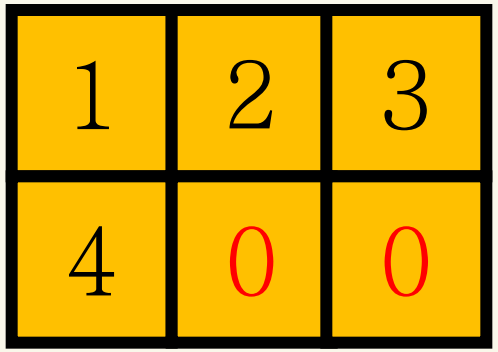
int a[2][3]; //设置为全局数组(主函数之外)



二维数组初始化

初始化部分元素

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



二维数组初始化

初始化部分元素

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

1	2	3
4	0	0

二维数组初始化

下面哪种形式不能使二维数组中的元素全部初始化为0呢？

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

int a[3][4];
int main()
{
    return 0;
}
```

A

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

int main()
{
    int a[3][4];
    return 0;
}
```

B

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

int main()
{
    int a[3][4] = {};
    return 0;
}
```

C

Part 3

二维数组元素获取

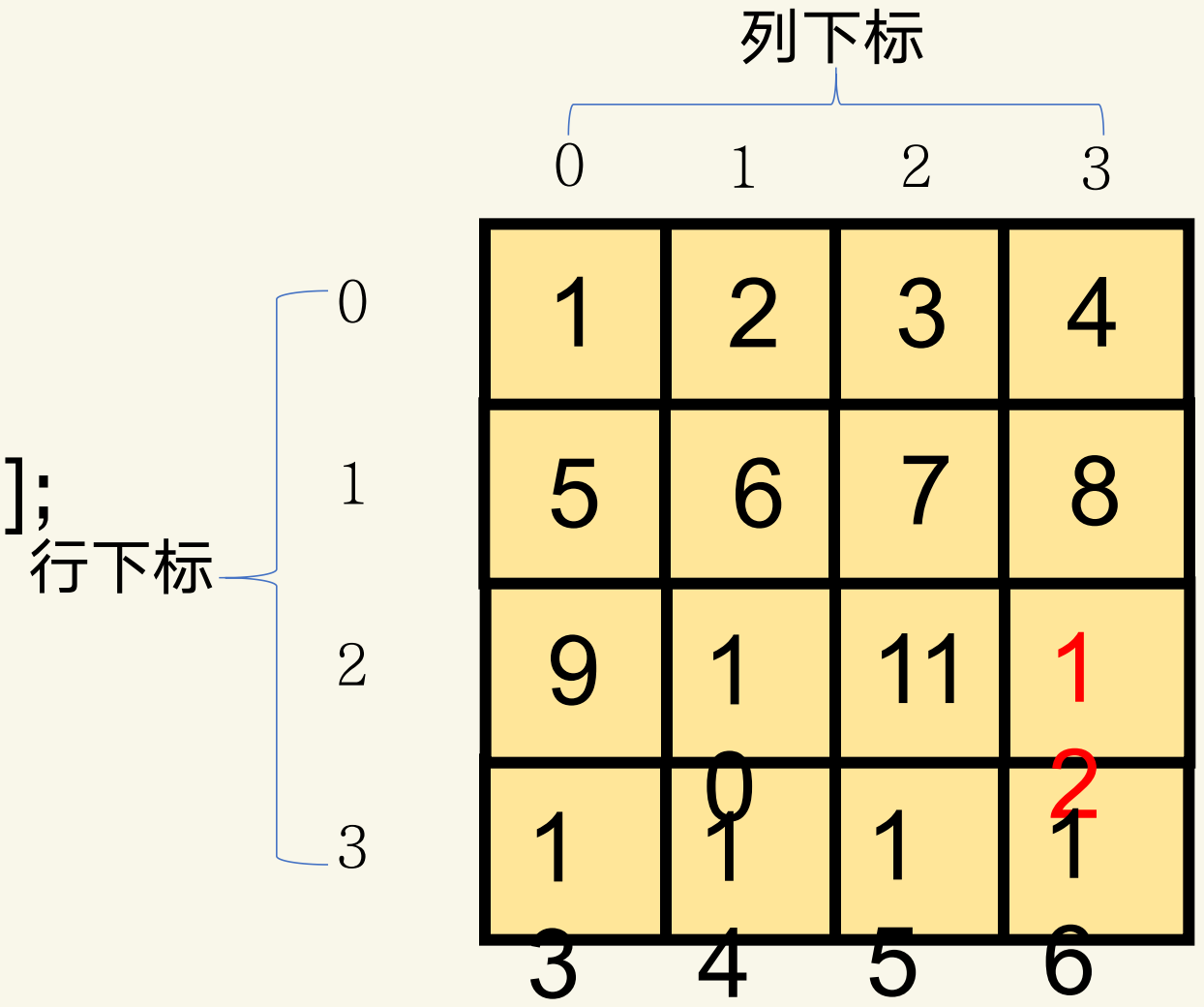
二维数组元素获取

二维数组元素获取

数组名[行下标][列下标];

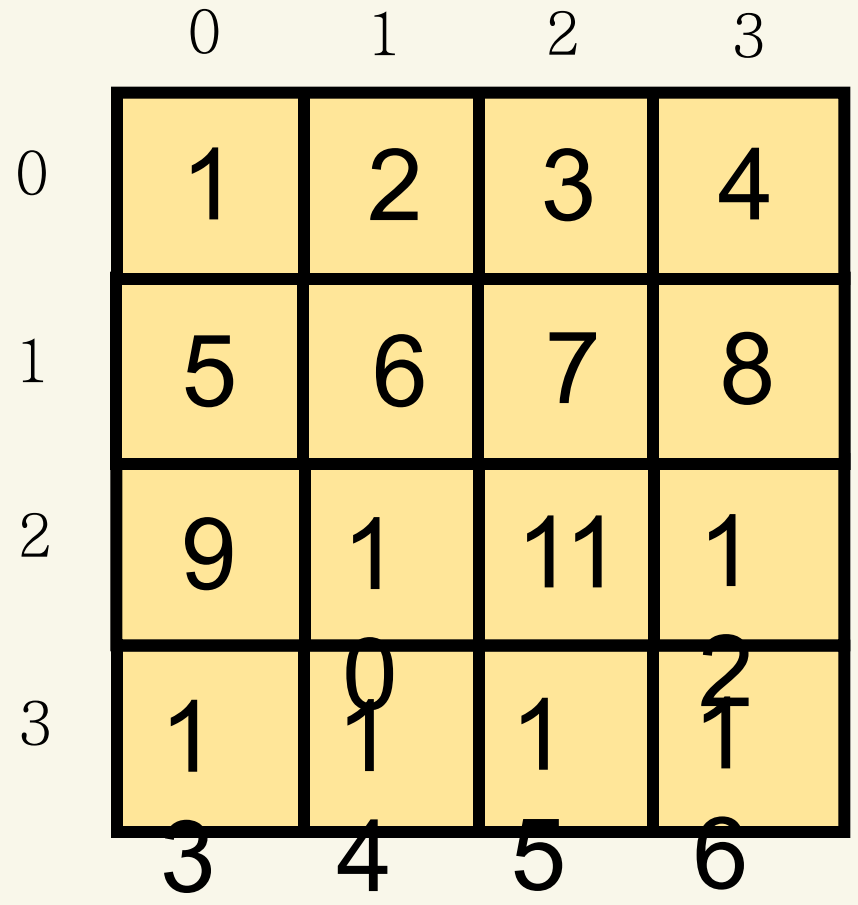
例如:

cout << a[2][3];



二维数组元素获取

a数组的元素初始化如下图所示，如何获取14这个元素呢？



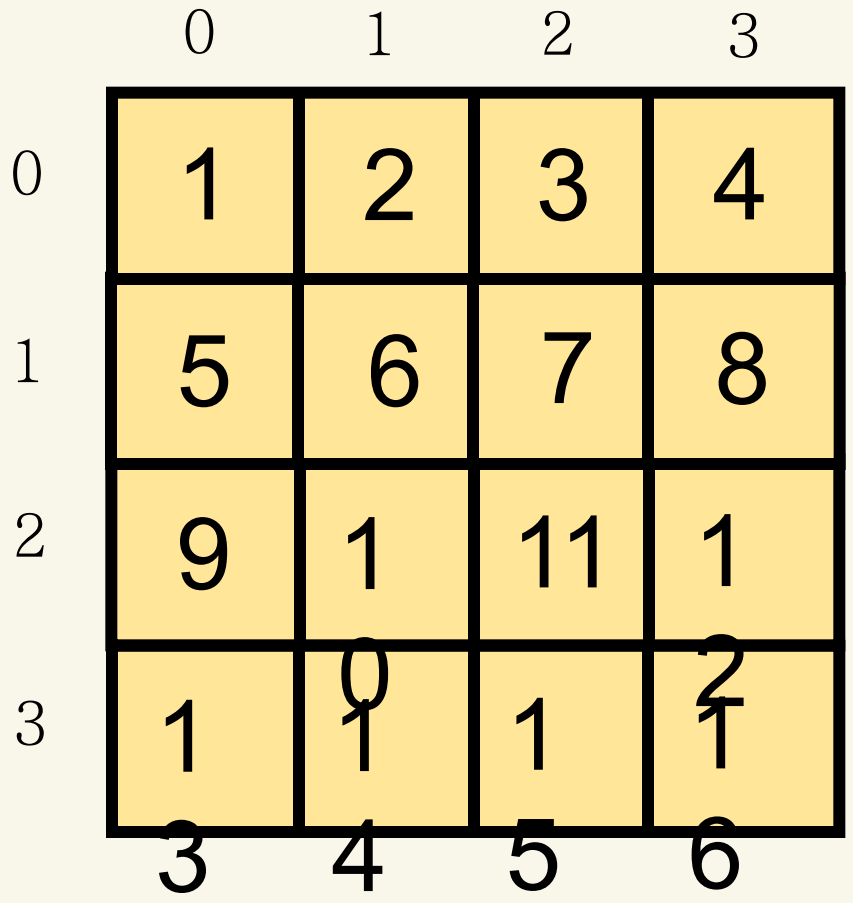
A、a[1][3];

B、a[3][1];

C、a[4][2];

二维数组元素获取

a数组的元素初始化如下图所示,通过a[1][2]获取到的是哪个元素呢



A、2

B、7

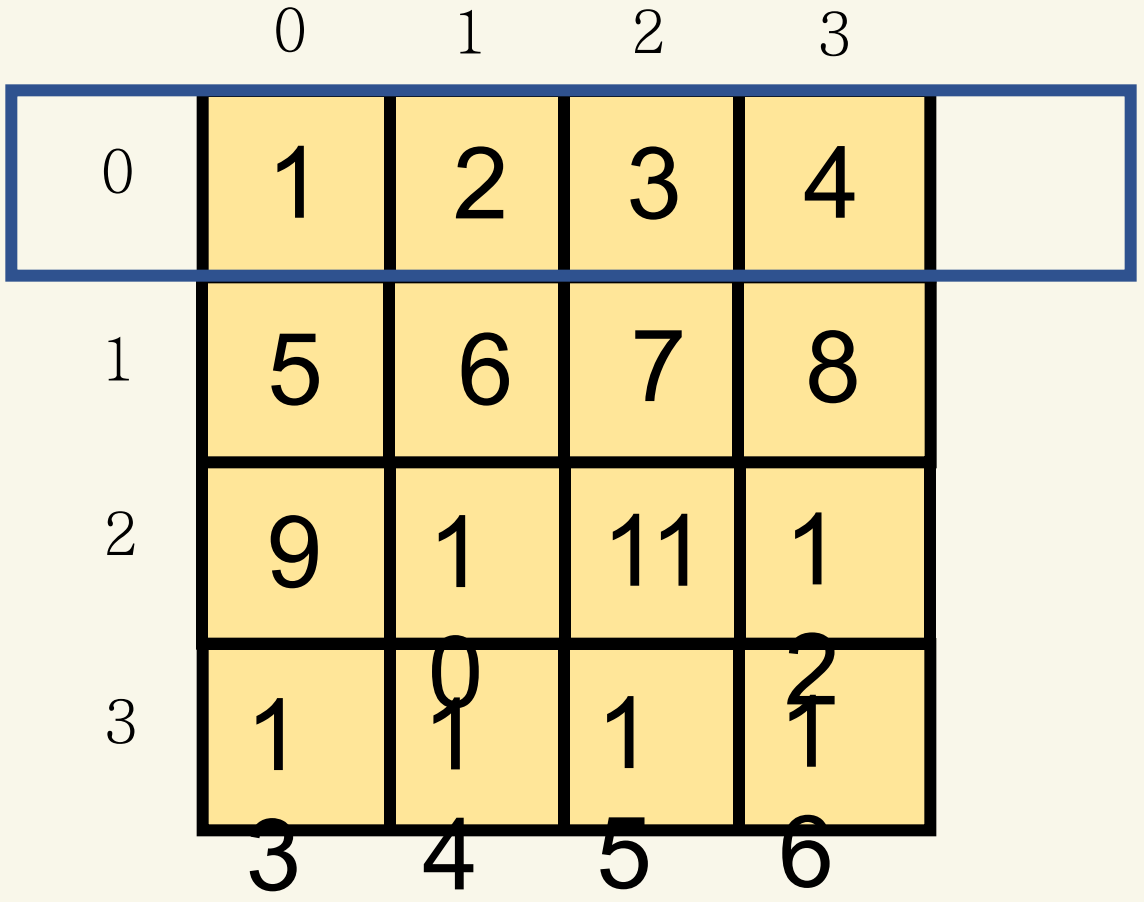
C、10

Part 4

二维数组输入输出

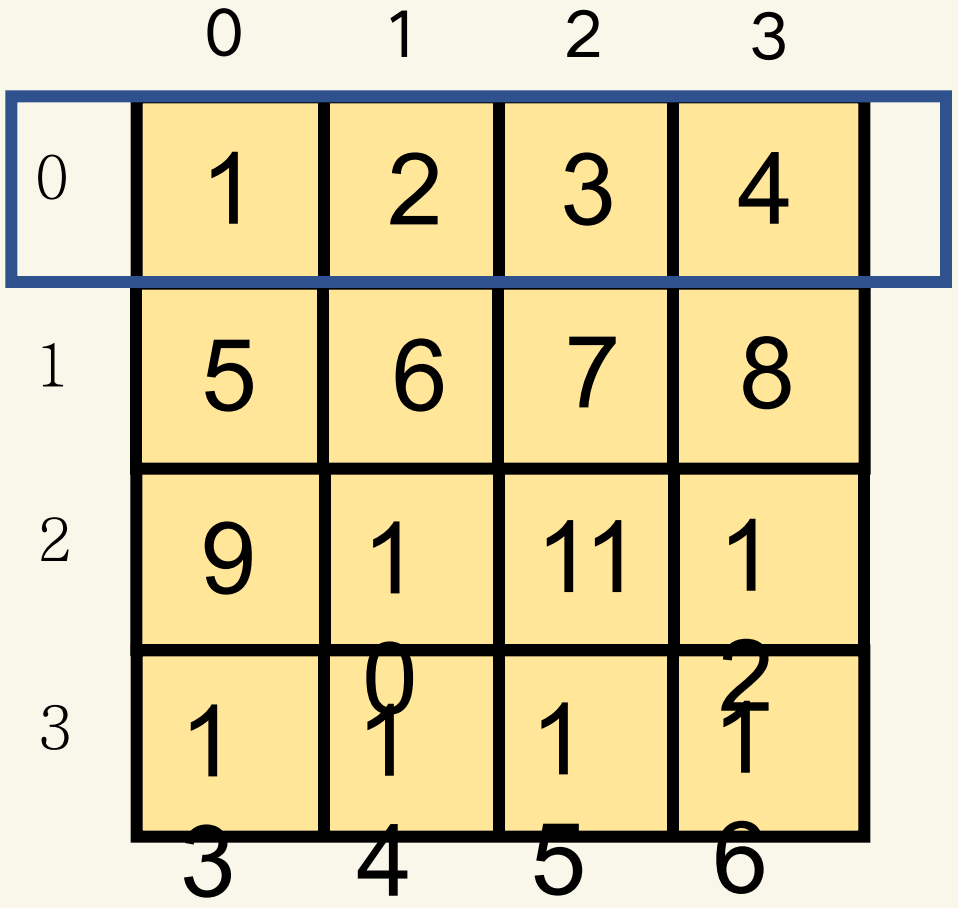
二维数组输入输出

如何获取第一行元素？



二维数组输入输出

在同一行的元素行下标相同



a[0][0]:

a[0][1]:

a[0][2]:

a[0][3]:

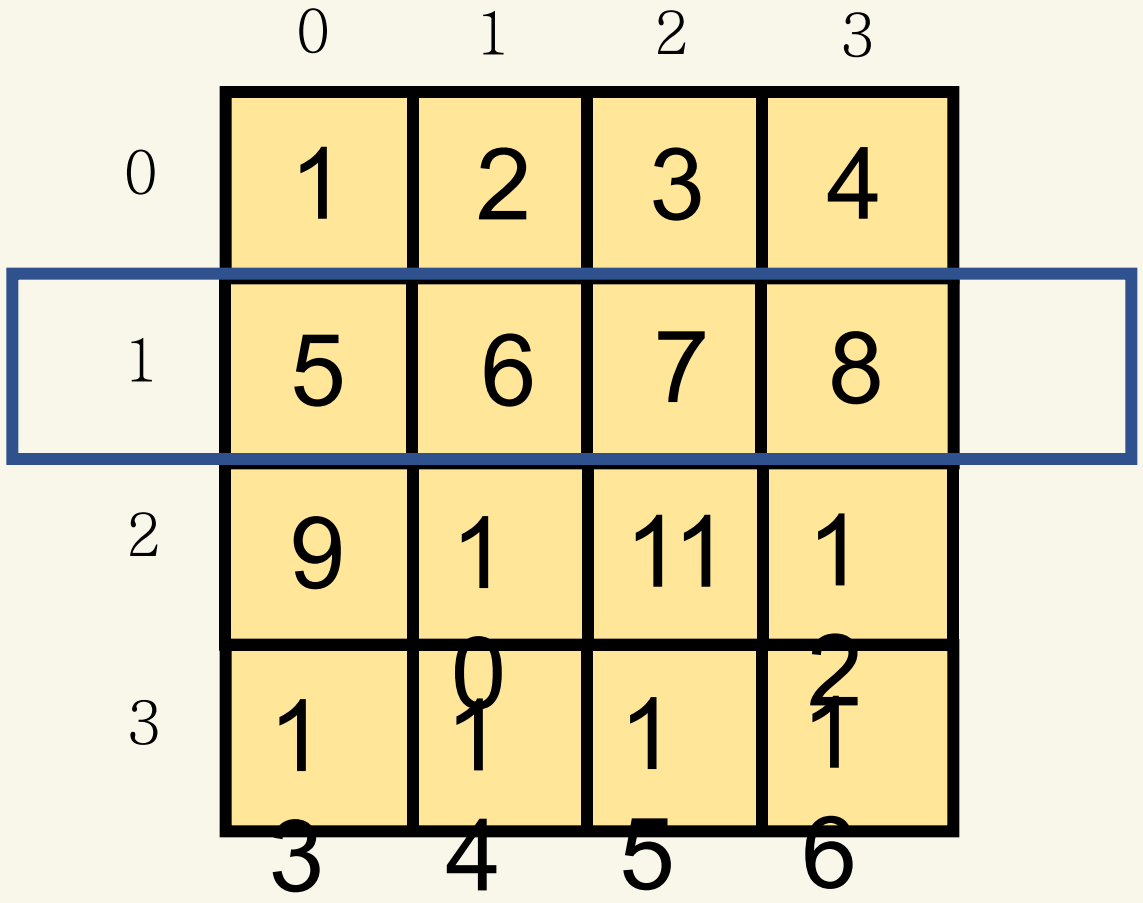
二维数组输入输出

二维数组遍历行

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

二维数组输入输出

如何获取第二行元素？



二维数组输入输出

二维数组遍历行

```
for (int j = 0; j < 4; j++)
{
        cout << a[1][j] << " ";
}
```

二维数组输入输出

输出所有元素：

使用循环嵌套，外层循环遍历行，

内层循环遍历一行的所有元素

二维数组输入输出

输出a中的所有元素

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

二维数组输入输出

下面程序想要遍历输出a数组中的所有元素，则①和②处应该填写的内容为

```
int a[4][3] = {{1,2,3},{4,5,6},
{7,8,9},{10,11,12}};
for (int i = 0; ①; i++)
{
    for (int j = 0; ②; j++)
    {
        cout << a[i][j] << " ";
    }
    cout << endl;
}
```

A、
i < 4 j < 3

B、
i < 3 j < 2

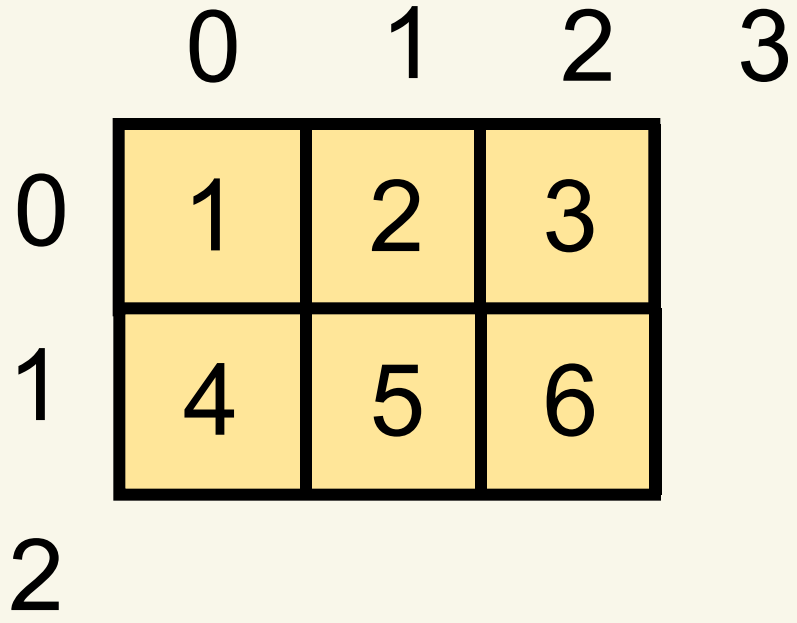
C、
i < 3 j < 4

二维数组输入输出

下标越界

a[2][3]

×



二维数组输入输出

二维数组下标范围：

行下标: 0~行长度-1

列下标: 0~列长度-1

二维数组输入输出

二维数组输入

```
for (int i = 0; i < 行数; i++)
{
        for (int j = 0; j < 列数; j++)
        {
                cin >> a[i][j];
        }
}
```

Part 5

总结回顾

总结回顾

二维数组的基本概念

二维数组：相同数据类型的一维数组的集合

行下标：决定了数组元素所在的行

列下标：决定了数组元素所在的列

越界：超出数组存储的范围

二维数组的定义方式：数据类型 数组名 [行长度][列长度];

二维数组遍历：通过使用二维数组下标，依次获取数组里面的所有元素

总结回顾

二维数组的初始化

1、全部初始化为0

```
int a[2][3] = {};
int a[2][3]; //定义成全局数组
```

2、依次初始化每个元素

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

3、初始化部分元素

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

部分初始化中未被初始化的部分会被直接初始化为0

总结回顾

二维数组的遍历输入

```
for (int i = 0; i < 行数; i++)
{
    for (int j = 0; j < 列数; j++)
    {
        cin >> a[i][j];
    }
}
```

二维数组的遍历输出

```
for (int i = 0; i < 行数; i++)
{
    for (int j = 0; j < 列数; j++)
    {
        cout << a[i][j] << " ";
    }
    cout << endl;
}
```

总结回顾

二维数组和一维数组对比

	二维数组	一维数组
全部初始化为0	int a[10][10] = {};	int a[10] = {};
遍历方式	for循环嵌套	for循环
数组容量	行长度*列长度	数组长度

Class is over