

C 语言二维数组

上节讲解的数组可以看作是一行连续的数据，只有一个下标，称为**一维数组**。在实际问题中有很多量是二维的或多维的，因此 C 语言允许构造多维数组。多维数组元素有多个下标，以确定它在数组中的位置。本节只介绍二维数组，多维数组可由二维数组类推而得到。

二维数组的定义

二维数组定义的一般形式是：

```
dataType arrayName[length1][length2];
```

其中 ,dataType 为数据类型 ,arrayName 为数组名 ,length1 为第一维下标的长度 ,length2 为第二维下标的长度。例如：

```
int a[3][4];
```

定义了一个 3 行 4 列的数组，共有 $3 \times 4 = 12$ 个元素，数组名为 a，即：

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]

在二维数组中，要定位一个元素，必须给出一维下标和二维下标，就像在一个平面中确定一个点，要知道 x 坐标和 y 坐标。例如，a[3][4] 表示 a 数组第 3 行第 4 列的元素。

二维数组在概念上是二维的，但在内存中地址是连续的，也就是说存储器单元是按一维线性排列的。那么，如何在一维存储器中存放二维数组呢？有两种方式：一种是按行排列，即放完一行之后顺次放入第二行。另一种是按列排列，即放完一列之后再顺次放入第二列。

在 C 语言中，二维数组是按行排列的。也就是先存放 a[0] 行，再存放 a[1] 行，最后存放 a[2] 行；每行中的四个元素也是依次存放。数组 a 为 int 类型，每个元素占用 4 个字节，整个数组共占用 $4 \times (3 \times 4) = 48$ 个字节。

【示例】一个学习小组有 5 个人，每个人有三门课的考试成绩。求全组分科的平均成绩和各科总平均成绩。

--	张	王	李	赵	周
Math	80	61	59	85	76
C	75	65	63	87	77
English	92	71	70	90	85

可设一个二维数组 `a[5][3]` 存放五个人三门课的成绩。再设一个一维数组 `v[3]` 存放所求得各分科平均成绩，设变量 `average` 为全组各科总平均成绩。编程如下：

```
1. #include <stdio.h>
2. int main() {
3.     int i, j; //二维数组下标
4.     int sum=0; //当前科目的总成绩
5.     int average; //总平均分
6.     int v[3]; //各科平均分
7.     int a[5][3]; //用来保存每个同学各科成绩的二维数组
8.     printf("Input score:\n");
9.     for(i=0; i<3; i++){
10.        for(j=0; j<5; j++){
11.            scanf("%d", &a[j][i]); //输入每个同学的各科成绩
12.            sum+=a[j][i]; //计算当前科目的总成绩
13.        }
14.        v[i]=sum/5; // 当前科目的平均分
15.        sum=0;
16.    }
17.    average =(v[0]+v[1]+v[2])/3;
18.    printf("Math: %d\nC Language: %d\nEnglish: %d\n", v[0], v[1], v[2]);
19.    printf("Total:%d\n", average);
20.    return 0;
21. }
```

运行结果：

Input score:

80 61 59 85 76 75 65 63 87 77 92 71 70 90 85 ✓

Math: 72

C Language: 73

English: 81

Total:75

程序中首先用了一个双重循环。在内循环中依次读入某一门课程各个学生的成绩，并把这些成绩累加起来，退出内循环后再把该累加成绩除以 5 送入 `v[i]` 之中，这就是该门课程的平均成绩。外循环共循环三次，分别求出三门课各自的平均成绩并存放在 `v` 数组之中。退出外循环之后，把 `v[0]`、`v[1]`、`v[2]` 相加除以 3 即得到各科总平均成绩。最后按题意输出各个成绩。

二维数组的初始化

二维数组的初始化可以按行分段赋值，也可按行连续赋值。

例如对数组 `a[5][3]`，按行分段赋值可写为：

```
int a[5][3]={ {80, 75, 92}, {61, 65, 71}, {59, 63, 70}, {85, 87, 90}, {76, 77, 85} };
```

按行连续赋值可写为：

```
int a[5][3]={80, 75, 92, 61, 65, 71, 59, 63, 70, 85, 87, 90, 76, 77, 85};
```

这两种赋初值的结果是完全相同的。

【示例】求各科平均分和总成绩平均分。

```
1. #include <stdio.h>
2. int main() {
3.     int i, j;    //二维数组下标
```

```

4.      int sum=0;    //当前科目的总成绩
5.      int average;  //总平均分
6.      int v[3];     //各科平均分
7.      int a[5][3]={ {80,75,92}, {61,65,71}, {59,63,70}, {85,87,90},
    {76,77,85} };
8.
9.      for(i=0; i<3; i++){
10.         for(j=0; j<5; j++){
11.             sum+=a[j][i];    //计算当前科目的总成绩
12.         }
13.         v[i]=sum/5;    // 当前科目的平均分
14.         sum=0;
15.     }
16.
17.     average =(v[0]+v[1]+v[2])/3;
18.     printf("Math: %d\nC Language: %d\nEnglish: %d\n", v[0], v[1],
    v[2]);
19.     printf("Total:%d\n", average);
20.
21.     return 0;
22. }

```

运行结果：

Math: 72

C Language: 73

English: 81

Total:75

对于二维数组初始化赋值还有以下说明

1) 可以只对部分元素赋初值，未赋初值的元素自动取 0 值。例如：

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

是对每一行的第一列元素赋值，未赋值的元素取 0 值。赋值后各元素的值为：

1 0 0

2 0 0

3 0 0

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

赋值后的元素值为：

0 1 0

0 0 2

3 0 0

2) 如对全部元素赋初值，则第一维的长度可以不给出。例如：

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

可以写为：

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

3) 数组是一种构造类型的数据。二维数组可以看作是由一维数组的嵌套而构成的。设一维数组的每个元素都又是一个数组，就组成了二维数组。当然，前提是各元素类型必须相同。根据这样的分析，一个二维数组也可以分解为多个一维数组。C 语言允许这种分解。

如二维数组 `a[3][4]`，可分解为三个一维数组，其数组名分别为：`a[0]`、`a[1]`、`a[2]`。

对这三个一维数组不需另作说明即可使用。这三个一维数组都有 4 个元素，例如：一维数组 `a[0]` 的元素为 `a[0][0]`, `a[0][1]`, `a[0][2]`, `a[0][3]`。必须强调的是，`a[0]`, `a[1]`, `a[2]` 不能当作下标变量使用，它们是数组名，不是一个单纯的下标变量。