第七章-数组

一维数组

一维数组的定义

一维数组：我们把物理上前后相邻、类型相同、表示一个类别的一系列有序数据集合称之为一维数组。

一维数组中每一个变量称为数组元素，变量的个数称为数组长度或者数组容量。

一维数组的定义方式为：

【存储类型】数据类型 数组名【数组长度】；

例如：int a[30];

说明：

（1） 数据类型可以是基本数据类型也可以是构造数据类型。

（2） 数组名是合法的标识符

（3） 数组长度在C语言中 必须定义时就确定，是定值常量，也可以是常量表达式，但不能是左值。即数组元素个数是固定的。

（4） 数组元素的下标从0 开始

（5） 定义的数组元素在内存中是按顺序连续存放的

（6） 存储类型有auto和static类型

（7） C语言对数组下标越界不做检查，编程时要自己注意

（8） 一般数组不能整体输入输出，只能对数组某个元素进行输入和输出

一维数组的赋值

类似于对普通变量进行赋值，格式如下：

数据类型 数组名[数组元素个数] = {初始值列表};

一般有如下几种赋值方式：

（1） 数组全部赋值

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

（2） 当数组元素赋初值个数等于数组长度时，可以不指定长度。

```
int b[] = {1, 1, 2, 4, 5};

// 这等价于 int b[5] = {1, 1, 2, 4, 5};
```

（3） 数组元素不初始化，其值为随机数；但是如果定义数组类型前有static时，其数组元素不赋初值时，系统默认全是0

int a[5]，后续没有赋初值时，a[0]~a[4]是随机数

static int a[5]，后续没有赋初值时，a[0]~a[4]都是0

（4） 部分初始化。

int a[5] = {1, 2, 3}; 赋值元素个数少于数组长度，缺少的位置补0

（5） 当数组长度 < 赋值数组个数时，报错

一维数组的引用

数组元素引用的格式如下：

数组名[下标]；

说明：

（1） 下标是从0开始的，并且下标只能是整型数据或者表达式。

（2） 下标不能越界，越了界会死。

（3） 数组元素只能单个引用，不能一次性引用整个数组，尤其是不能用数组名代替数组中的全部元素。

二维数组的定义

二维数组的定义方式为：

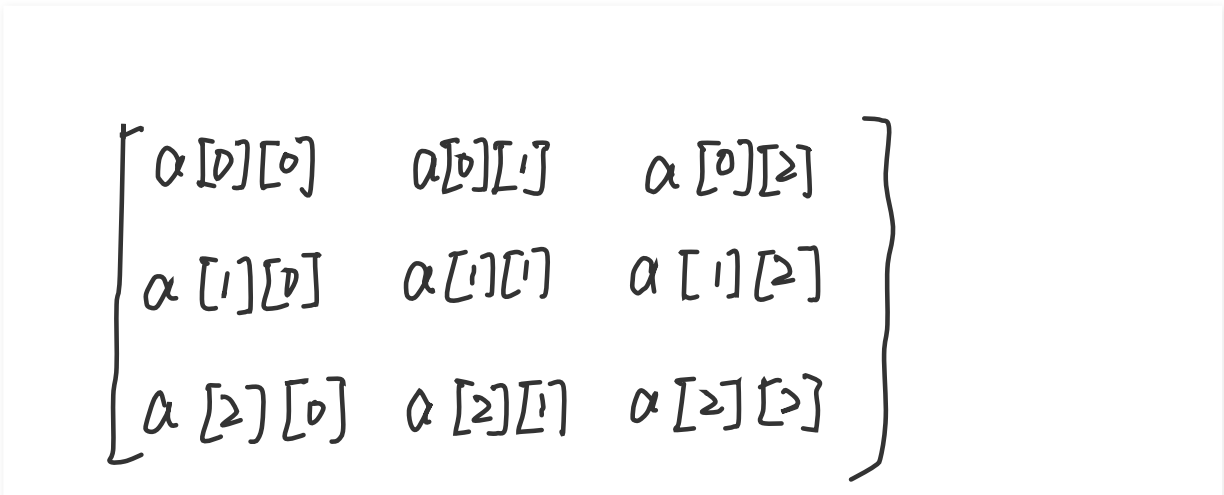
数据类型 数组名[常数1][常数2];

多维数组的定义方式为：

数据类型 数组名[常数1][常数2]……[常数n];

二维数组的存储与表示

设int a[3][3];则其对应的数组为：



可以看出二维数组以行为优先，在a[3][43中，第一个3表示的是行，第二个3表示的是列。二维数组在内存中的存储顺序也是以行优先，即一行一行存。上面数组的存储顺

列。二维数组在内存中的存储顺序也是以行优先，即一行一行存储。上面数组的存储顺序为：

a[0][0], a[0][1], a[0][2], a[1][0], a[1][1], a[1][2], a[2][0], a[2][1], a[2][2]

理解：

回到概念上：数组的概念就是在物理上前后相邻、类型相同、表示一个类别的一系列有序数据集合。一维数组a[5] = {1, 2, 3, 4, 5};中的类别就是整型数据，而二维数组就是a[3][3]中的类别就是一个个数组元素。数组也是一种数据类型，也可以作为数组的元素，如此套娃就有了多维数组的概念。

二维数组的初始化

1. 分行初始化
   》 二维数组全部元素初始化
   》》 int a[2][3] = {{1,2,3}, {4,5,6}};
   》 二维数组部分元素初始化
   》》 int a[2][3] = {{1,2}, {4}}; 缺位补0
   》 二维数组中第一维长度省略初始化
   》》 int a[][3] = {{1,2,3},{4,5}}; 这就和一维数组中数组长度和赋值数组元素长度相同时可以省略数组长度不写一样一样的。
2. 按元素排列顺序初始化
   》二维数组全部元素初始化
   》》 int a[2][3] = {1,2,3,4,5,6};
   》 二维数组部分元素初始化
   》》 int a[2][3] = {1,2,3,4}; 缺位补0
   》 二维数组中第一维长度省略初始化
   》》 int a[][3] = {1,2,3,4,5}; 缺位补0，二维中确定了每一行元素个数，根据赋值元素个数只能赋值两行上的数据，因此默认给数组一维给的值是2

二维数组的引用

```
数组名[行下标][列下标];
```

字符数组和字符串

字符数组

定义：用来存放字符数据的就是字符数组。

一维字符数组的定义格式：

字符数据类型 字符数组名[常数];

举例：char str[1000];

二维字符数组的定义格式：

字符数组类型 字符数组名[常数1][常数2];

举例：char ch[10][20];

字符数组的初始化

1. 逐字符初始化
   （1） 字符数组全部赋初值
   char ch[5] = {'H', 'e', 'l', 'l', 'o'};
   （2） 字符数组部分赋初值
   char ch[5] = {'H', 'e', 'l',}; //剩余元素自动赋值为空字符 ‘\0’

字符数组的引用

```
数组名[下标]  或者  数组名[下标1][下标2]...[下标n];
```

字符串的存储与结束

存储：C语言中字符串是存储在字符数组当中的。

结束：C语言会自动的在字符串的后面加上结束标志 ‘\0’ ，但这并不计入字符串的长度中去。

字符串和字符数组

```
# include <stdio.h>
# include <string.h>
int main(void){
    char cStr[] = {'H', 'e', 'l', 'l', 'o'};
    char sStr[] = "Hello"

    printf("cStr的存储长度的是%zd", sizeof(cStr));  //cStr的存储长度的是5
    printf("sStr的存储长度是%zd", sizeof(sStr));  //sStr的存储长度是6
}
```

注：sizeof()函数用来测试参数所占自己长度，由以上输出可以看到sStr所占长度要比cStru多一个字节。

字符数组的输入输出

（1） 逐个字符I/O：用格式%c输入或输出一个字符

（2） 整个字符串I/O：用格式%s，意思是对字符串进行格式化输入输出，要注意的是使用%s时，空格和回车都是结束符号。

常用的字符串处理函数

字符串处理函数包含在头文件string.h中，编译预处理为：

#include

字符串输出函数puts

其一般格式为：puts(字符数组名)；

字符串输入函数gets

其一般格式为：gets(字符数组名)；

字符串连接函数：strcat

其一般格式为：strcat（字符数组名1， 字符数组名2）；

字符串复制函数strcpy 和 strncpy

strcpy

一般格式：strcpy(字符数组名1，字符数组名2);

功能：将字符数组2复制到字符数组1中去，要求字符数组1 的长度要能够容纳字符数组2.

strncpy

一般格式：strncpy(字符数组名1， 字符数组名2，n)；

功能：将字符数组2中的前n个字符复制到字符数组1当中去替换字符数组1的前n个字符，但是n不能大于strlen(str2)

字符串长度测试函数

一般格式：strlen(字符数组)；

字符串比较函数strcmp

一般格式：strcmp(字符数组名1，字符数组名2)；

功能：判断两个字符串是否相等，原则是，对两个字符串的字符逐个比较（ASCII码比较），遇到不同字符或者\0停止，比较返回值如下：

较产，遇到不用字符或者\0停止。比较返回值如下：

字符串1 < 字符串2 == 》 返回（-1）

字符串1 > 字符串2 == 》 返回（1）

字符串1 = 字符串2 == 》 返回 （0）