

Chapter7 数组

warm up:

数组：相同类型数据元素的集合

- 数组大小必须是确定的
- 一个数组中每一个数据元素的数据类型是相同的，不允许出现不同类型的数据

1、一维数组

1.1 定义

数据类型 数组名[整型常数表达式]

ex:

```
int a[6];double v[8];
```

- C语言新规定，可以传入一个参数，记为n，把n给一个数组，即int a[n]的形式。是完全合法的！
- 整型表达式呈形 8%6 。就是不能有变量在里边

1.2 初始化

【法一】先定义数组：

之后再一个一个的初始化（可以采用循环）

```
for(int i=0;i<n;i++)  
    a[i]=i;
```

tip:使用本方法赋值，若有元素没有被赋值，那么这个元素将被赋予**随机值**

【法二】在定义数组的时候顺便初始化：

```
---使用花括号{}  
int a[8]={1,2,3,4}
```

tip：使用本方法赋值，若有元素没有被赋值（花括号内数字少于元素个数），那么其他元素将被赋予**0**

tip：以上两种方法，如果初值的类型和数组元素的类型不一致，C编译系统把初值类型转换为数组元素的类型！

ex: int a[]={1.5,2.3,4,5,6}

meaning1:这个数组有5个元素，即为int a[5]

meaning2:第一个元素是1，第二个元素是2

1.3 常用操作（含冒泡排序、选择排序）

1. 数组元素的插入（按大小从小到大）
 - 核心：找到位置了之后，先把最后一个元素往后挪，从后往前挪元素
2. 元素的删除
 - 核心：找到位置了之后，先把这个位置元素后面那个元素挪到当前位置，行前往后挪元素
3. 元素的排序！！

3.1冒泡排序

想法:

- 比较相邻两个元素，如果前大后小，则交换位置。否则不变
- 两层循环。外层循环 $n-1$ 次，内层循环 $n-1-i$ 次
 - 外层循环：一共有 $n-1$ 个元素，每一次循环确定了一个元素的位置（从元素大的开始确定）
 - 内层循环：通过相邻比较+换位置》找到最大、次大、第三大...的元素》并把他们排到最后去

【文件夹chapter7中的mao_pao.c】

```
#include <stdio.h>
int main()
{
    int num;
    scanf("%d",&num);
    int str[num]; // 创建了数组
    for(int i=0;i<num;i++)
        scanf("%d",&str[i]); // 给数组赋值
    for (int i=0;i<num;i++)
        for (int j=0;j<num-1-i;j++) // 可以举一个例子，就知道j是到多大就可以不用比较啦！
        {
            // 当然，如果嫌麻烦的话，让j小于num-1就可以啦，只不过多比较了几次
            if(str[j]>str[j+1])
            {
                int temp;
                temp=str[j];
                str[j]=str[j+1];
                str[j+1]=temp;
            }
        }
    for (int i=0;i<num;i++)
        printf("%d ",str[i]);
}
```

3.2选择排序

想法:

- 找到数组元素中最小的元素，把它放到前面的位置去！
- 即第 i 轮选择结果是把数组中第 i 小的元素的值放到了第 i 个元素的位置上，直到选择完成
- 两层循环：比冒泡好理解啦

【文件夹chapter7中的xuan_ze.c】

```
#include <stdio.h>
int main()
{
    int num;
    scanf("%d",&num);
    int str[num];
    for(int i=0;i<num;i++)
        scanf("%d",&str[i]);
    for (int i=0;i<num;i++)//从第一个位置开始，寻找是哪个是最小元素，并且和第一个位置换
    {
        int min=i;
        for(int j= i+1;j<num;j++)//内层循环用来找到最小元素，找到了之后，就用外层循环换位置！
        {
            //【bug: j必须能找到所有i之后的元素！】
            if (str[min]>str[j])
                min=j;
        }
        //把min的位置和i的位置进行互换
        int temp=str[i];
        str[i]=str[min];
        str[min]=temp;
    }
    for(int i=0;i<num;i++)
        printf("%d ",str[i]);

}
```

2、二维数组

2.1定义

理解：特殊的一维数组，它的每一个元素又都是一个一维数组

内存：按行存放，先存放第一行的元素。基地址：a[0][0]。同理多维数组，从最后一个维度开始存储

2.2初始化

法一：花括号

按照数组元素的排列顺序一次对各元素赋予初值

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

tip:如果在定义的时候，对它的全部元素提供初始值，则该数组的**行标**可以不指定，但是**列表**不能缺少！

ex:

```
int a[ ][3]={1,2,3,4,5,6};//会自动识别行标为2呢
```

法二：分行+花括号

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

tip:如果在定义的时候，某一行的给定个数小于规定个数，后面的元素自动设置为0！

法三：两层循环+一个一个元素赋值喽

3、字符数组

在标准C语言中没有专门的字符串类型变量，所以一般用字符数组来存放字符串。

一个字符型的一维数组可以用来存放一个字符串，一个字符型的二维数组可以用来存放多个字符串！

3.1定义

ex: char c[5];

字符数组之中每一个元素占用一个字节

3.2初始化

法一：赋值语句一个元素一个元素赋值

```
char c[3]={'a','b','c'};
```

法二：优选！！！使用字符串常量世界对字符数组初始化

```
char c[15]="Beijing";
```

tip:

- 如果提供初值的个数（字符个数）大于数组长度，则报语法错误！
- 如果提供初值的个数（字符个数）小于数组长度，则后面其余的元素自动定位空字符('\0')

重要规定

使用字符+花括号来初始化的时候，**定义的时候数组是多长就有多少的字符来接**

```
ex: char c[2]={'a','b'}; 【此时没有\0,但也是合法的】
```

使用字符串来初始化的时候，定义的时候数组是多长，只能有**长度-1**个字符元素可以被接受到值
因为系统对字符串常量的后面自动加一个'\0'来判定字符串结束！

```
ex: char c[5]="ouou"; 【此时有\0，也是合法的】
```

其中，\0代表的是ASCII码值为0的字符，不是一个可以显示的字符，而是一个“空操作符”，即什么也不干的意思。只作为字符串是不是结束的标志

ex: char c[10]="happy";//从第六个元素开始后面五个元素全都是空字符'\0'!!!

3.3字符数组输入输出

法一：逐个字符输出，使用格式符"%c"，输出一个字符

法二：整个字符串一次输入输出，使用格式符"%s"

ex:

```
char c[10];
```

```
scanf("%s",c);//此时用字符数组名字就可以啦，不需要取地址。字符数组本身就是引用的地址
```

```
printf("%s",c);
```

tip: 如果输入的长“文字”之中含有空格，则无法输入！

```
char str[13];
```

```
scanf("%s",str);
```

输入: Good morning

—————结果是只接收到了Good\0

3.4字符串处理函数

C语言中没有对字符串进行合并、比较和赋值的运算符，对字符串的处理过程被编写成库函数放在C的函数库之中，供人们使用时候调用

- 库函数并非C语言的组成部分，而是人们为方便将具有固定功能的模块编写成供大家使用的公共函数，是各种版本C自带的部分

1. puts(输出字符数组)

将一个字符串（以'\0'结束）输出到终端【花括号+字符赋值就不可以啦，没有'\0'呢！】

- tip：使用puts()函数输出的字符串中可以包含转义字符！

ex:

```
char str[]="beijing\nrui";
```

```
puts(str);
```

2. gets（输入字符数组）可以包含空格一起输入，直到遇到回车键才结束输入！

从终端输入一个字符串到字符数组中，并且得到一个返回值【不太关心这个返回值！】

比较：

```
char c1[80],c2[80];
gets(c1);
printf("%s",c2);
```

```
输入 I am a chinese.
c1就是 I am a chinese.
c2是 I
```

gets()可以包含空格一起输入，直到遇到回车键才结束输入！

scanf()遇到空格或者回车都可以结束输入

3. strcat()（连接两个字符串）

- 连接两个字符数组中的字符串，把字符串2接到字符串1之后，**结果放在字符串1中**

```
strcat(str1,str2);
```

tip:

字符串1的长度必须足够大，可以容纳后面新接入的字符串。否则就会报错
新字符串中，取消了字符串1后面的标志'\0'，保留了字符串2后面的标志'\0'

4. strcpy()(字符串复制函数)

- 把字符串2复制到字符数组1之中

```
strcpy(str1,str2);
```

其中str2可以直接是一个字符串常量，也可以是一个已经定义过的数组名

补充：

- 两种不合法的写法

```
char str1[15],str2[15];
```

```
str1=str2;//字符数组不能直接相互赋值
```

```
str1="beijing";//字符数组定义了之后就不能被字符串常量直接赋值了！
```

- strcpy()函数可以设定复制的数量

```
strcpy(str1,str2,3) //把str2中前3个字符复制到str1之中，这样str1中的前3个字符被取代！
```

5. strcmp()(【有返回值】字符串比较函数，得到函数返回值)

```
strcmp(str1, "China");
```

- 比较规则：从左到右逐个字符比较他们的ASCII码，直到遇到不相同的或结束标志'\0'为止。
如果字符串1=字符串2，函数值为0
字符串1>字符串2，返回值为正整数
字符串1<字符串2，返回值是负整数

6. strlen()(【有返回值】，返回不含'\0'的字符串长度)

7. strlwr()(把字符串中的大写字母转换成小写字母)

8.strupr()(把字符串中的小写字母转换成大写字母)

编程技巧：

①s1[i]!='\0';--算字符串的长度

②s1[i]-s2[i]是两个ASCII码相减