# 目录

[目录 1](#_Toc520304641)

[第5章 C语言-数组 2](#_Toc520304642)

[5.1 C语言数组概述 2](#_Toc520304643)

[5.1.1 何谓数组 2](#_Toc520304644)

[5.1.2 数组的特征 2](#_Toc520304645)

[5.2 一维数组定义 2](#_Toc520304646)

[5.2.1 一维数组如何定义 2](#_Toc520304647)

[5.2.2 一维数组的定义与初始化 3](#_Toc520304648)

[5.3 一维数组使用 3](#_Toc520304649)

[5.3.1 一维数组使用方法 3](#_Toc520304650)

[5.4 字符数组 5](#_Toc520304651)

[5.5 字符串与一维数组 5](#_Toc520304652)

[5.6 冒泡排序算法 8](#_Toc520304653)

[5.6.1 冒泡排序算法作用 8](#_Toc520304654)

[5.6.2 冒泡排序如何操作 8](#_Toc520304655)

[5.7 二维数组 12](#_Toc520304656)

[5.7.1 二维数组定义与初始化 13](#_Toc520304657)

[5.7.2 二维数组的使用 14](#_Toc520304658)

# C语言-数组

## C语言数组概述

基本数据类型：整型、字符型、浮点型 🡪 本身存在的，直接使用 int a; char b; float c;

构造数据类型：数组类型、结构体类型、公用体类型、枚举型、链表 🡪 本身没有🡪构造出来才能用

整型：int

整型定义一个变量：int a；//定义了一个整型变量a

数组类型本身是没有，要构造出来才有

数组类型：数据类型 [长度] int [3] char [5]

数组定义一个数组变量：int [3] a; int a[3]; 🡪 定义变量的格式：数据类型 变量名

数组类型定义一个变量：数组类型 变量名；数据类型 [长度] 变量名；🡪 数据类型 数组名[长度]；int [5] a;

### 何谓数组

数组：大量具有相同数据类型的集合 集合：很多数据一起

换句话说：就是由多个相同类型变量，并且它们地址是连续组合而成；int a[50];

//存50个学生数学成绩(double)

double a1…a50; //麻烦，代码量大

数组就很好的解决这个问题

double [50] a; 🡪 double a[50];

### 数组的特征

1. 数据都是相同数据类型
2. 地址连续
3. 使用前，大小已经确定好了

## 一维数组定义

### 一维数组如何定义

整型：int

整型定义一个变量：int a;定义了一个整型变量a

int [5] a; char a[50] float a[50]//定义了元素个数为50，元素类型是float类型的空间

数组类型：数据类型 [长度]

数据类型：可以基本数据类型、构造数据类型、指针类型 决定了每个元素空间的类型

[长度]：数组里面元素的个数，只能是整型常量表达式(正整数)元素个数为正整数

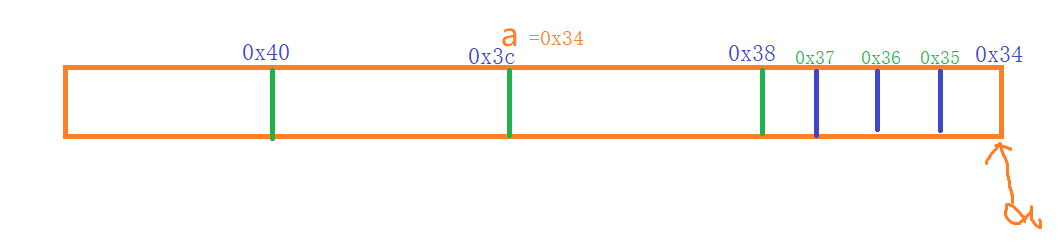
数组类型定义一个变量：数组类型 变量名；定义了一个数组变量

数据类型 [长度] 变量名；🡪 数据类型 数组名[长度]；

数组名：

1. 满足标识符的命名规则，具有含义
2. 数组名是首元素地址

int a[4];



举例

1. int a[5]; ok
2. int a[2.5]; no 长度:正整数
3. int a[2.5+1.5]; no
4. int a[‘a’]; //int a[97]; ok
5. int a[“a”]; //no
6. int b = 5;

int a[b]; //no

1. #define N 5

int a[N];

### 一维数组的定义与初始化

1. 数组的完全初始化，对数组的每个元素一一赋值

int a[3] = {1, 2, 3}; //系统默认从第0个元素开始赋值

//根据数据类型和元素个数来分配空间

1. 数组的未完全初始化，未赋值的元素，系统默认赋值0

int a[3] = {1, 2}; 1 2 0 //系统默认从第0个元素开始赋值 char str[20] = {0};

1. 数组进行初始化时，可以省略长度

int a[] = {1, 2, 3}; //根据大括号里面的元素个数和元素类型来分配空间

1. 如果函数内定义数组，未进行初始化时，元素的值为垃圾值

int a[5]; //元素的值就为随机值

1. 数组元素清零

int a[3] = {0};//利用与未初始化的原理

1. 如果在使用的过程中，需要赋一个相同的值，怎么赋？

int a[3];//3个空 a[0] a[1] a[2]

1

a[3] = 1;//越界

只能一个一个赋值 利用for循环

## 一维数组使用

1. 格式： 数组名[下标]； 只有定义时叫长度，其它都叫下标

下标的范围：0 – 长度-1 int a[50]; a[0] – a[49]

1. 对数组的使用

操作数组，本质就是操作数组元素空间

数组名的类型：数据类型 \* int a[4] int \* char a[4] char \*

每一个元素空间都没有名字，所以不能使用直接操作，只能间接操作(地址)

\*(a+0) 可以访问第0个元素 \*(a+0) == a[0]

\*(a+1)可以访问第1个元素 \*(a+1) == a[1]

\*(a+2)可以访问第2个元素

\*(a+3)可以访问第3个元素

课堂练习1：

将100以内的素数用数组保存起来，再将数组里面的值打印出来

## 字符串与一维数组

字符串：由多个字符和’\0’组成

存字符串的两种方式  
char a[6] = “hello”;//字符数组 是 字符串

a[0] = ‘g’;

char \*p = “hello”; //字符串存储在常量区(只读)

p[0] = ‘g’;

char a[10] = “hello”;//字符串存储在栈区(可读可写)

a[0] = ‘g’;

1. 字符数组不一定是字符串

字符串不一定是字符数组

char a[5] = {‘h’, ‘e’, ‘l’, ‘l’, ‘o’};//是字符数组 不是字符串 字符数组并不是字符串

char a[6] = {‘h’, ‘e’, ‘l’, ‘l’, ‘o’};//字符数组 字符串 字符数组也是字符串

char a[5] = “y”; //是字符串

char a[] = {‘h’, ‘e’, ‘l’, ‘l’, ‘o’}; //不是

char a[] = {‘h’, ‘e’, ‘l’, ‘l’, ‘o’, ‘\0’};//是字符串

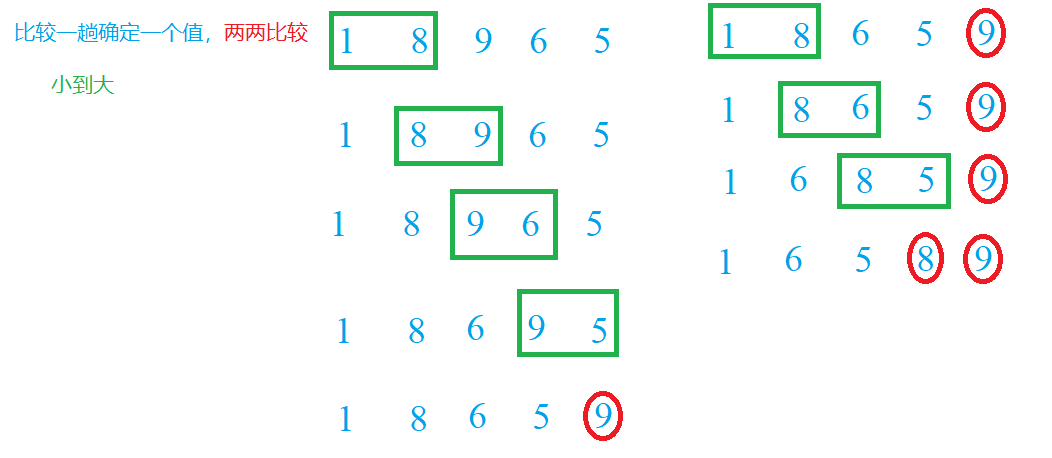
## 冒泡排序算法

### 冒泡排序算法作用

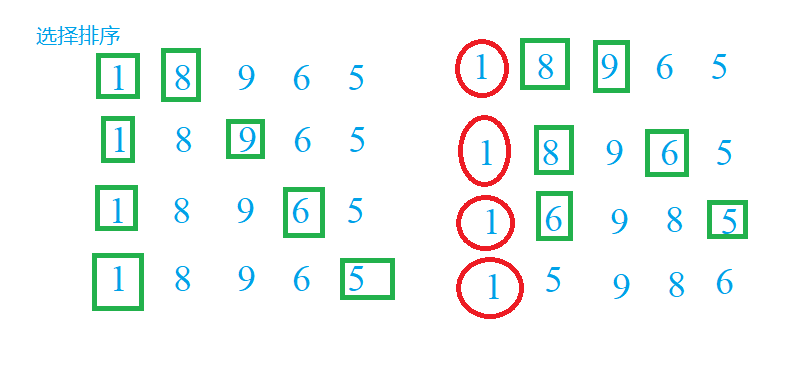
作用：对数据进行排序。小到大，大到小

### 冒泡排序如何操作

1 8 9 6 5



选择排序：固定一个数，一一进行比较



练习：用代码实现出来

## 二维数组

int a[5]; 数组名就是首元素地址

数组名的类型：元素类型 \* a:int \*

char b[5]; b：char \*

float c[5]; c: float \*

int a[5]; 元素类型：去掉数组名[长度]；

数组类型：去掉数组名 int [5];

### 二维数组定义与初始化

数组类型：数据类型 [长度] 一维数组

二维数组：数组数组

数组类型：数据类型 [长度1][长度2]

长度1：行长度

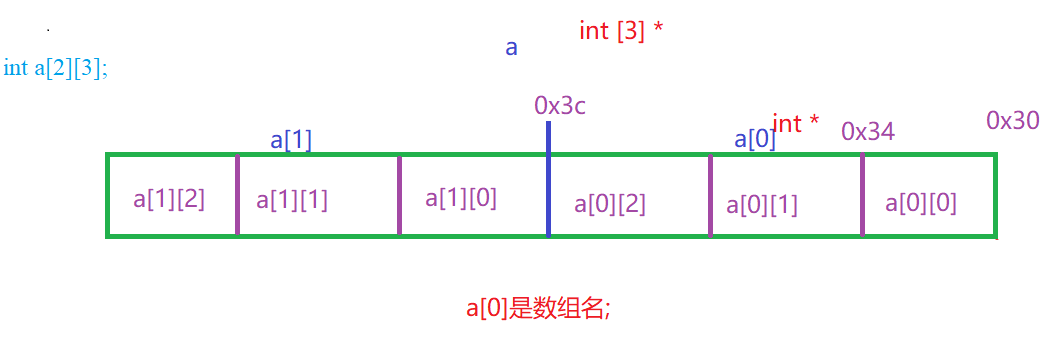
长度2：列长度

二维数组定义变量的格式：数组类型 变量名； 数据类型 [长度1][长度2] 变量名

通常的写法：数据类型 数组名[行长度][列长度]

int a[2][3];//二维数组的元素有2个，二维数组的元素元素有6个

元素类型：int [3];



初始化：

1. 二维数组的完全初始化，按照先行后列的顺序进行赋值

int a[2][3] = {{1, 2, 3 }, {4, 5, 6}}; //按行赋值

int a[2][3] = {1, 2, 3, 4, 5, 6}; //不按行赋值

完全初始化，按行赋值和不按行赋值效果一样

1. 二维数组的未完全初始化，未赋值的元素，系统默认赋值0

int a[2][3] = {{1, 2 }, {3}}; //按行赋值

1 2 0

3 0 0

int a[2][3] = {1, 2, 3}; //不按行赋值

1 2 3

0 0 0

1. 当二维数组初始化，行长度可以省略，但是列长度不能省略

int a[][3] = {1, 2, 3, 4}; //根据列长度，确定行长度

### 二维数组的使用

1. 引用格式

数组名[行下标][列下标]； 0 – 长度减一

\*(数组名[行下标]+ 列下标)；\*(a[0]+1)

\*(\*(数组名+行下标)+列下标)；\*(\*(a+0)+1)

1. 课堂练习
2. 输入6个数，用二维数组的形式存储，再打印出来
3. 1 2 3 转置为 1 4 利用3行2列的数组

4 5 6 2 5

3 6

1. 1 2 3 转置为 1 4 7 不需要

4 5 6 2 5 8

7 8 9 3 6 9