

## 数组的初始化

一维数组的定初始化。

数组是一组具有相同类型的变量的集合。

起的名字称为数组名，另一个就是数组元素的每个数据项称为数组元素。

C语言称为下标，实现对数组元素的访问。

## 数组的下标和查询

先定义数组的基类型，即数组中元素的类型。下标的个数表面数组的维数

C语言中数组的下标都是从0开始的。

在定义数组时不能使用变量定义数组的大小，可以是定义的常量。

## 二维数组的初始化和定义处理：

二维数组的定义及其初始化处理，

类型 数组名[元素的个数]；

数组中内存的处理和求和是通过多个维度的加和和求解来进行求解的

使用一个定义的常量进行可以更好的统一整个数组的长度和大小

使用需要重新的定义和处理，有利于提高程序的可移植性。

数组第二维度的长度的声明永远不能省略。

## 向函数传递一维数组不要

若要把一个数组传递给一个函数，那么只要使用不带方括号的数组名，作为函数实参调研函数即可。

数组代表数组第一个元素的地址，因此用数组名作函数实参实际上是将数组的首地址传递给被调参数。

数组名代表第一个地址，就是首地址。

最好是以负数的作为输入结束的标记值。用着标记控制的循环，标记循环的控制和解决。

## 排序和查找处理

排序的处理和查找的处理

交换法进行排序的处理

交换法排序；

```
for (i = 0; i<maxsize-1;i++)
```

```
for(j = i+1;j<maxsize;j++)
```

```
{
```

```
判断的条件
进行循环的置换数据处理
}
```

## 查找处理

查找的方法和处理：

线性查找又称为顺序查找，的和折半查找

线性查找是在查找键值，折半查找法是在已有正确的排序的情况下进行查找，其的速度比顺序查找快很多，折半查找也称为对分查找

首先选取数组的中间值的元素，将其与查找键进行比较，从而提高下标或降低上标，从而递归求得其的值得大小。

向函数传递二维数组，传递时与处理一维的数组的方式没有什么区别

```
#include<stdio.h>
#define maxsize 10
int main()
{
    int a[maxsize];
    int i;
    for(i = 0;i<maxsize;i++)
    {
        a[i] = i-1;
        printf("%d",i);
    }
    return 0;
}
```

///二维数组及其函数的产生过程

```
#include<stdio.h>
#define maxx 5
#define maxy 5

void indef(int a[maxx][maxy])
{
```

```

int i;
int j;
for(i = 0;i<maxx;i++)
{
    for(j = 0;j<maxy;j++)
    {
        a[i][j] = 4;
        printf("  %d",a[i][j]);
    }
    printf("\n");
}

int main()
{
    int a[maxx][maxy];
    int i,j;
    printf("输入的矩阵的结构: \n");
    for(i = 0;i<maxx;i++)
    {
        for(j = 0;j<maxy;j++)
        {
            a[i][j] = i-1+j-2;
            printf("  %d",i);
        }
        printf("\n");
    }
    printf("处理过程中的矩阵的结构: \n");
    indef(a);
    printf("处理后的矩阵的结构: \n");
    for(i = 0;i<maxx;i++)
    {
        for(j = 0;j<maxy;j++)
        {
            a[i][j] = 4;
            printf("  %d",a[i][j]);

```

```

        }
        printf("\n");
    }
    return 0;
}

```

//排序算法及数组的初始化处理

```
#include<stdio.h>
```

```
#define maxsize 10
```

```
void paixu(int a[maxsize])
```

```

{
    int i;
    int j;
    int temp;
    for(i = 0;i<maxsize-1;i++)
    {
        for(j = i+1;i<maxsize;j++)
        {
            if(a[i]>a[i+1])
            {
                temp = a[i];
                a[i] = a[i+1];
                a[i+1] = temp;
            }
        }
    }
}

```

```
int main()
```

```

{
    int a[maxsize];
    int i;
    for(i = 0;i<maxsize;i++)
    {

```

```

        a[i] = 10-i;
        printf("%d", i);
    }

    printf("\n");

    paixu(a);
    for(i = 0; i<maxsize; i++)
    {
        printf("%d", a[i]);
    }

    printf("\n");
    return 0;
}

```

需要注意的是，mid这个变量定义出来会是一个特别好的一个变量使得这个中间值更加清晰。同事也提醒我们，一个量出现3次，就有必要将其定义成一个变量。

```

while(low<=high)
{
    mid = (high+low)/2;
    if(x>num[mid])
        low = mid+1;上浮下标
    else if (x<num[mid])
        high = mid-1;下浮上标
    else
        return mid;得到返回值
}

```