数组：

1、定义：

一组数据类型相同且在内存上连续存储的数据集合

注：① 数据类型相同 ==》每个元素都是同一种格式，且每个元素在内存上所占空间大小相同

② 内存上连续存储 ==》元素的存储地址是相连的

③ 数据集合 ==》有多个数据

一维数组：

2、格式：

存储类型 数据类型 数组名[数组大小];

例：auto int arr [8];

注：① 存储类型表示数组中的内容的存储位置，一般默认为auto可以省略

② 数据类型表示数组中每个元素的数据类型，可以是基本数据类型，也可以是构造数据类型

③ 数组名表示数组整体名称，符合标识符命名规则，也表示数组首元素地址

④ 数组大小表示数组中最多可以保存的数据个数，只能是常量

3、初始化

① 全部初始化 --- 一个萝卜一个坑

int arr[10] = {10,20,30,40,50,60,70,80,90,100};

② 部分初始化 --- 坑比萝卜多

int arr[10] = {10,40,60,70,80,100};

③ 省略初始化 --- 坑的数量由萝卜决定

int arr[] = {10,20,50,70,80,90};

注：数组可以和变量一样初始化，也可以不初始化，但是若使用省略初始化格式定义数组，则数组必须初始化

4、数组成员调用

int arr[10] = {10,20,30,40,50,60,70,80,90,100};

float brr[6] = {12.3,45.43,121.23,4.56,53.42,1.25};

格式：数组名[下标];

注：下标范围为 0 - (N - 1)，N表示数组大小

练习：将具有10个整型元素的数组倒序并输出，不开辟新数组空间

5、数组大小计算

sizeof(数组名) / sizeof(数据类型)

sizeof(数组名) / sizeof(数组名[0])

**笔试原题：使用宏函数实现数组大小计算**

#define VAL(a) sizeof(a)/sizeof(a[0])

二维数组：

（2）格式：

存储类型 数据类型 数组名[行数][列数];

例：auto int arr[3][4];

注：① 列数表示一行最多有几个元素

② 数组大小 = 行数 \* 列数;

③ 存储方式：按行存储，上一行填满，接着填充下一行

（3）初始化：

① 全部初始化

普通初始化：int arr[3][4] = {10,20,30,40,50,60,70,80,90,100,110,120};

按行初始化：int arr[3][4] = {{10,20,30,40},{50,60,70,80},{90,100,110,120}};

② 部分初始化

普通初始化：int arr[3][4] = {10,40,50,70,90,120};

按行初始化：int arr[3][4] = {{10,40},{50,70,90}};

③ 省略初始化

普通初始化：int arr[][4] = {10,40,50,70,90,120};

按行初始化：int arr[][4] = {{10,40},{50,70,90},{120}};

（4）数组元素访问：

格式：数组名[行下标][列下标];

注：行下标范围为 0 - (M - 1)，M表示数组行数

列下标范围为 0 - (N - 1)，N表示数组列数

作业：

1. 从终端输入10个数，将最大值放在第一个数，最小值放在最后一个数，其他数不动(不能使用排序)

#include<stdio.h>

int main()

{

int arr[10]={1,2,35,6,8,9,3,0,5,4},i,max,min;

int flag1,flag2;

for(i=0;i<10;i++)

{

scanf("%d",&arr[i]);

}

max=min=arr[0];

for(i=0;i<10;i++)

{

if(arr[i]>max)

{

max=arr[i];

flag1=i;

}

}

for(i=0;i<10;i++)

{

if(arr[i]<min)

{

min=arr[i];flag2=i;

}

}

while(flag1>0)

{

arr[flag1]=arr[flag1-1];

flag1--;

}

while(flag2<10)

{

arr[flag2]=arr[flag2+1];

flag2++;

}

arr[0]=max; arr[9]=min;

for(i=0;i<10;i++)

{

printf("%d\n",arr[i]);

}

return 0;

}2、打印斐波那契数列前10项（使用数组）

1 1 2 3 5 8 13 21 34

#include<stdio.h>

int main()

{

int arr[10]={1,1},i,t;

for(i=2;i<10;i++)

{

arr[i]=arr[i-1]+arr[i-2];

}

for(i=0;i<10;i++)

{

printf("%d\n",arr[i]);

}

return 0;

}

3、输出杨辉三角前10行

4、有两个递增数组，将其合并为一个数组后，新数组仍然保持递增,不能重新开辟新的空间(不能使用排序)

#define M 4

#define N 3

#include<stdio.h>

int main()

{

int a[M+N]={2,4,6,7},b[N]={1,3,5};

int i,j,k,t=0,c;

int end1=a[M-1],end2=b[N-1];//a3 b2

if(end1>end2)

{//往a放

for(i=0;i<M+N;i++)

{ //每次a中的每一个元素和b[0]比较

if(a[i]>=b[0])//比b[0]大的话

{

for(k=M+N-1;k>i;k--)//把b[0]放在a的前面，其他往后挪一个

{

a[k]=a[k-1]; //倒着挪，空第几个

}

a[i]=b[0];

t++;//计数

if(t>=N)

{

break;

}

for(j=0;j<N;j++)//b往前一个

{

b[j]=b[j+1];

}

}

}

for(k=0;k<M+N;k++)

{

printf("%d\n",a[k]);

}

}

else

{//b放

c=b[2];

for(i=0;i<M+N;i++)

{ //每次a中的每一个元素和b[0]比较

if(a[i]>=b[0])//比b[0]大的话

{

for(k=M+N-1;k>i;k--)//把b[0]放在a的前面，其他往后挪一个

{

a[k]=a[k-1]; //倒着挪，空第几个

}

a[i]=b[0];

t++;//计数

if(t>=N)

{

break;

}

for(j=0;j<3;j++)//b往前一个

{

b[j]=b[j+1];

}

}

}

a[M+N-1]=c;

for(k=0;k<M+N;k++)

{

printf("%d\n",a[k]);

}

}

return 0;

}

1. 选择题
2. 以下关于数组的描述正确的是 C 。

A. 数组的大小是固定的，但可以有不同的类型的数组元素。

B. 数组的大小是可变的，但所有数组元素的类型必须相同。

C. 数组的大小是固定的，但所有数组元素的类型必须相同。

D. 数组的大小是可变的，但可以有不同的类型的数组元素。

1. 在定义int a[10];之后，对a的引用正确的是 D 。

A. a[10] B. a[6.3] C. a(6) D. a[10-10]

1. 以下能正确定义数组并正确赋初值的语句是 D 。

A. int n=5,b[n][n]; B. int a[1][2]={{1},{3}};

C. int c[2][]={{1,2},{3,4}} D. int a[3][2]={{1,2},{3,4}}

1. 在执行int a[][3]={1,2,3,4,5,6};语句后，a[1][0]的值是 A 。

A. 4 B. 1 C. 2 D. 5

1. 读程序写结果
2. #include <stdio.h>

void main()

{

int a[8]={1,0,1,0,1,0,1,0},i;

for(i=2;i<8;i++)

a[i]+= a[i-1] + a[i-2];//a2=1+0+1=2 a3=0+0+2=2

//a4=1+2+2=5 a5=0+2+5=7 a6=1+7+5=13 a7=13+7+0=20

for(i=0;i<8;i++)

printf(“%5d”,a[i]);

}

//1 0 2 2 5 7 13 20

1. #include <stdio.h>

void main()

{

float b[6]={1.1,2.2,3.3,4.4,5.5,6.6},t;

int i;

t=b[0];//1.1

for(i=0;i<5;i++)

b[i]=b[i+1];//2.2 3.3 4.4 5.5 6.6 1.1

b[5]=t;

for(i=0;i<6;i++)

printf(“%6.2f”,b[i]);

}

//2.2 3.3 4.4 5.5 6.6 1.1

1. #include <stdio.h>

void main()

{ int p[7]={11,13,14,15,16,17,18},i=0,k=0;

while(i<7 && p[i]%2)

{ k=k+p[i]; i++;}

printf("k=%d\n",k);//11+13=24

}

1. void main()

{ int a[3][3]={1,3,5,7,9,11,13,15,17};

int sum=0,i,j;

for (i=0;i<3;i++)

for (j=0;j<3;j++)

{ a[i][j]=i+j;

if (i==j)

sum=sum+a[i][j];

}

printf("sum=%d",sum);

}

//0+1+2=3

1. void main()

{ int a[4][4],i,j,k;

for (i=0;i<4;i++)

for (j=0;j<4;j++)

a[i][j]=i-j;//

for (i=0;i<4;i++)

{ for (j=0;j<=i;j++)

printf("%4d",a[i][j]);

printf("\n");

}

}

//0 1 -1 2 1 0 3 2 1 0

1. 填空题
2. 构成数组的各个元素必须具有相同的 数据类型 。
3. 下面的程序是输出数组中最大元素的下标(p表示最大元素的下标)。

void main()

{

int i,p;

int s[]={1,-3,0,-9,8,5,-20,3};

for(i=0,p=0;i<8;i++)

if(s[i]>s[p]) p=i ;

printf(“%d\n”,p);

}

1. 输入20个数，输出他们的平均值，输出与平均值之差的绝对值最小的数组元素。

#include <stdio.h>

#include <math.h>

void main()

{

float a[20],pjz=0,s,t;

int i,k;

for(i=0;i<20;i++)

{

scanf(“%f”,&a[i]);

pjz+=a[i]/20；

}

s=fabs(a[0]-pjz);

t=a[0];

for(i=1;i<20;i++)

if( fabs(a[i]-pjz)<s )

{ s= fabs(a[i]-pjz;

t=a[i];

}

Printf(“%d %d”,pjz,s);

}