C语言程序设计上机实验报告

实验（1-7）数组

一:实验目的与技术

1. 熟练掌握一维数组、二维数组的定义、赋值、输入、输出方法。
2. 熟练掌握一维数组及一维数组元素储存方式。
3. 熟练掌握一维数组和二维数组的使用以及在函数中的传递。
4. 熟练掌握字符数组和字符串函数的使用
5. 实验步骤与内容：

7-1：输入并运行以下程序，分析程序运行结果。

#include<iostream.h>

void main()

{

int i ,a[5];

a[0]=2,a[1]=3,a[2]=5,a[3]=8,a[4]=10;

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

cout<<”a[“<<i<<”]=”<<a[i]<<endl;

}

分析：

Int a[5]：定义一个数组名为a的整形数组，该数组可以存储5个整数。[ ]中的数值称为数组下标，下标从0开始到数组长度减1，因此数组a的下标从0到4。

7-2: 输入并运行以下程序，分析程序运行结果。

#include<iostream.h>

void main()

{

int max,i,j,k;

int a[ ]={5,8,7,1,2,11,10};/\*数组元素初始化，将8个整数存放到数组a中\*/

max=a[0],k=0;/\*默认第0个数组元素为最大值，并记录下标\*/

for(i=1;i<7;i++)

if(max<a[i]) {max=a[i];k=i;}/\*用max保存当前最大值，用k保存当前最大值的下标\*/

cout<<”position is”<<k+1<<endl;

cout<<”value is”<<max<<endl;

}

分析：

1. 对全部元素赋初值时，可以不指定数组长度，编译系统自动根据初值个数来决定数组长度。
2. 一般数组的操作用循环来实现。

7-3：程序实现：从键盘输入一个数，将其插入到一个降序数组中，保持数组仍然按降序排列。将程序补充完整。

#include<iostream.h>

#include<iomanip.h>

void main()

{

int data,temp,k;

int a[9]={50,25,15,10,8,4,2,-10};

cout<<”Enter a data:”;

cin>>data;

\_\_\_\_\_\_\_\_\_\_\_\_;

for(k=8;\_\_\_\_\_;k--）

if(a[k]>a[k-1])

{

\_\_\_\_\_\_\_\_\_\_;

a[k]=a[k-1];

a[k-1]=temp;

}

else \_\_\_\_\_\_\_\_\_\_;

for(k=0;\_\_\_\_\_\_\_\_\_;k++)

cout<<setw(5)<<a[k];

cout<<”\n”;

}

7-4：输入并运行以下程序，分析程序运行结果。

#include<iostream.h>

#include<iomanip.h>

void main()

{

int a[5][5]={{1,2,3,4,5},{6,7,8,9,10},{11,12,13,14,15},{16,17,18,19,20},{21,22,23,24,25}};

int i,j;

cout<<”打印输出二维数组的值：”<<endl;

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

{ for(j=0;j<5;j++)

cout<<setw(5)<<a[i][j];/\*按行输出二维数组\*/

cout<<”\n”;

}

cout<<”输出主对角线上的值：”<<endl;

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

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

ff(i==j) {cout<<setw(3)<<a[i][j];break;}/\*输出矩阵主对角线上的值\*/

cout<<endl;

}

分析：

给二维数组赋初值有两种形式。第一种按元素的存储顺序赋值，如：“int a[2][2]={1,2,3,4};”；

第二种按行赋值，如：“int a[2][2]={{1,2],{3,4}};”。

7-5：函数FUN的功能是：把形参a所指数组中的偶数按原顺序依次存放a[0],a[1],a[2]......，把奇数从数组中删除，偶数个数通过函数值返回。例如：若a所指数组中的数组最初排列为：9,1,4,2,3,6,5,8,7，删除奇数后a所指数组中的数据为：4,2,6,8，返回值为4。

#include<iostream.h>

#include<iomanip.h>

#define N 9

int fun(int a[],int n)

{

int i,j;

j=0;

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

if(\_\_\_\_1\_\_\_==0)

{\_\_\_\_\_2\_\_\_\_\_=a[i];j++;}

return\_\_\_3\_\_\_\_;

}

void main()

{

int b[N]={9,1,4,2,3,6,5,8,7},i,n;

cout<<”\n The original data:”<<endl;

for (i=0;i<N;i++) cout<<setw(4)<<b[i];

cout<<”\n”;

n=fun(b,N);

cout<<”\n The number of even:”<<endl;

cout<<”\n The even:”<<endl;

for(i=0;i<n;i++) cout<<setw(4)<<b[i];

cout<<”\n”;

}

7-6：输入并运行以下程序，分析程序运行结果。

#include<iostream.h>

void main()

{

char a[6]=”BASIC”;

cout<<a<<endl;

cout<<a[0]<<”,”<<a[1]<<”,”<<a[2]<<endl;

}

分析：

1. 数组可以用字符串进行初始化“char a[6]=”BASIC:”;”，其结果如下图所示。不能将字符串直接赋给数组，如“a[6]=”BASIC”;”和“a=”BASIC”;”都是错误的。

地址 数组变量名

a B a[0]

a+1 A a[1]

a+2 S a[2]

a+3 I a[3]

a+4 C a[4]

a+5 \0 a[5]

（2）系统存储字符串是会在末尾自动追加一个字符’\0’，因此将字符串赋给字符数组是要充分考虑到该字符数组的长度是否大于字符串长度。

7-7：输入并运行以下程序，分析程序运行结果。

#include<iostream.h>

#include “string”

void main()

{

char a[][8]={“sQL”,”Foxpro”,”Unix”,”ASP”};

char str[8];

int i,k,j;

cout<<”排序前字符串：”<<endl;

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

cout<<a[i]<<endl;

/\*用选择排序法将数组a中的字符串按升序排列\*/

for(i=0;i<=2;i++)

{

k=i;

for(j=i+1;j<=3;j++)

If(strcmp(a[k],a[j]>0) k=j;/\*比较字符串大小，将小者的位置保存到k中\*/

if(k!=i)

{

strcpy(str,a[k]);

strcpy(a[k],a[i]);

strcpy(a[i],str);

}

}

cout<<”排序后字符串：”<<endl;

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

cout<<a[i]<<endl;

}

分析：

“char a[][8]={“sQL”,”Foxpro”,”Unix”,”ASP”};”定义了一个二维字符数组。a[0]是第0个字符串的首地址，a[1]是第1个字符串的首地址，a[i]是第i个字符串的首地址。其存储形式见下图

地址

a[0] S Q L \0

a[1] F o x p r o \0

a[2] U n i x \0

a[3] A S PS \0

7-8：通过自定义函数，统计数组中字母，数字以及其他字符的个数并输出。把程序补充完整（sum1存放字母的个数，sum2存放数字字符的个数，sum3存放其他字符的个数）

#include<iostream.h>

#include”string”

Void count(char a[])

{

Int sum1=0,sum2=0,sum3=0;

Int i;

For(i=0;a[i]!=’\0’;i++)

If(a[i]>=’a’&&a[i]<=’z’||a[i]>=’A’&&a[i]<=’Z’) sum1++;

Else if(\_\_\_\_\_\_\_) \_\_\_\_\_\_ ;

Else\_\_\_\_\_\_\_;

Cout<<”sum1=”<<sum1<<”,”<<”sum2=”<<sum2<<”,”<<”sum3=”<<sum3<<endl;

}

Void main()

{

Char string[81];

Cout<<”input string:”<<endl;

\_\_\_\_\_\_\_\_\_\_-;

Count(string);

}

分析：

当形参为数组时，对应的实参为数组名，传递的是数组的首地址，此时实参数组与形参数组共享同一段内存单元，形参数组元素的改变会影响实参数组元素的改变。

7-9：请编程计算下列给出的二维数组周边元素之和。

1 3 5 7 9

2 9 9 9 4

6 9 9 9 8

1 3 5 7 0

7-10：请编写函数fun,该函数的功能是移动字符串中的内容。移动的规则如下：把第1到第m个字符，平移到字符串的最后，把第m+1到最后的字符移到字符串的前部。

例如：字符串中原有的内容为ABCDEFGHIJK，m的值为3，移动后，字符串中的内容应该是DEFGHIJKABC。

7-11：一个合法的身份证号码由17位地区、日期编号和顺序编号加1位校验码组成。

校验码的计算规则如下：

首先对前17位数字加权求和，权重分配为：{7,9,10,5,8,4,2,1,6,3,7,9,10,5,8,4,2}；然后将计算的和对11驱魔得到的值Z；最后按照以下关系对应Z值与校验码M值：

Z：0 1 2 3 4 5 6 7 8 9 10

M：1 0 X 9 8 7 6 5 4 3 2

现在给定一些身份证号码，请你验证校验码的有效性，并输出有问题的号码。

输入格式：

输入第一行给出正整数N（<=100）时输入的身份证号码的个数。随后N行，每行给出1个18位身份证号码。

输出格式：

按照输入的顺序每行输出1个有问题的身份证号码。这里并不检验前17位时候合理，只检查前17位是否全为数字且最后1位校验码计算准确。如果所有号码都正常，则输出“All passed”。

输入样例1：

4

320124198808240056

12010X198901011234

110108196711301866

37070419881216001X

输出样例1：

12010X198901011234

110108196711301866

37070419881216001X

输出样例2：

2

320124198808240056

110108196711301866

输出样例2：

All passed

1. 实验小结

通过实验，我更加理解了老师在课堂上讲的内容。对数组能够理解并应用，知道了数组的输入输出用for循环来处理。前面的题目是老师在教室上课的时候都提到过，也肯定是我上课不认真听讲，很多关于数组的基本处理都不会。还好上机课认真听课，并且按部就班的操作，再加以理解，突然觉得数组比之前学的什么函数之类的更好学。不过上机题11我真的是连题目都没有读懂啊！我把老师的参考答案抄在了书上，在我课后再写一遍的时候发现差距还是大，有点挫败...

班级：15机设4班

学号：115040100418

姓名：高燕兵