学号 P21714001 专业 计算机英才班 姓名 刘峰

实验日期 **2019.11.18**  教师签字 成绩

实验报告

【实验名称】   **C语言实验四**

【实验目的】

1. 掌握一维数组和二维数组的定义、赋值和输入输出的方法；
2. 掌握字符数组和字符串函数的使用；
3. 掌握与数组有关的算法（特别是排序算法）。

【实验原理】

(1) 一维数组的定义:类型说明符 数组名[常量表达式];

(2) 二维数组的定义:类型说明符 数组名[常量表达式1] [常量表达式2];

(3) 选择法排序的过程：

a）首先通过n-1次比较，从n个数中找出最小的， 将它与第一个数交换—---第一趟选择排序，结果最小的数被安置在第一个元素位置上

b）再通过n-2次比较，从剩余的n-1个数中找出关键字次小的记录，将它与第二个数交换—--第二趟选择排序

c）重复上述过程，共经过n-1趟排序后，排序结束

【实验内容】

1. 用选择法将任意8个整数按由小到大的顺序排列

#include<iostream>

using namespace std;

//初始化8个整数

int array[8];

int temp\_max;

int temp;

void initializeArray(){

    cout<<"请输入8个整数："<<endl;

    for(int i=0;i<8;i++){

        cin>>array[i];

    }

}

void sort(){

    for(int i=0;i<8;i++){

        for(int j=i+1;j<8;j++){

            temp\_max=i;

            if(array[temp\_max]>array[j])

            {

                temp\_max=j;

            }

            if(temp\_max!=i){      //上述是找到最大值的索引值，下面要让索引值对应的数值交换

                temp=array[i];

                array[i]=array[temp\_max];

                array[temp\_max]=temp;

            }

        }

    }

}

void display(){

cout<<"排序后的结果是："<<endl;

    for(int i=0;i<8;i++){

        cout<<array[i]<<" ";

    }

    cout<<endl;

}

int main(){

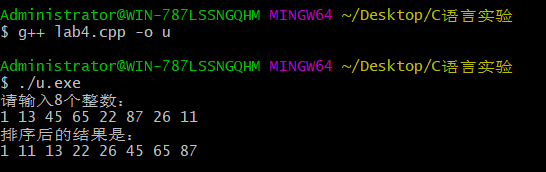
    initializeArray(); //初始化数组

    sort(); //对数组排序

    display(); //显示排序后的数组结果

    return 0;

}



1. 有15个数按升序存放在一个数组中，输入一个数，要求用折半查找法找出该数是数组中第几个元素的值。如果该数不在数组中，则输出“无此数”。这15个数用赋初值的方法在程序中给出。要找的数用scanf函数输入。（P166习题9）

#include<iostream>

#include<cstdio>

#include<algorithm>

using namespace std;

int array[15]={13,32,43,54,56,76,78,89,13,25,36,37,75,86,90};//按照题意，这15个数按赋初值方法，我自己给出

int num,start=0,End=14,middle;

int boundDown=array[start],boundUp=array[End];

void reduceHalf(){

    sort(array,array+15);

    cout<<"排序后的数是： ";            //利用STL库将乱序数值递增排序

    for(int i=0;i<15;i++){

        cout<<array[i]<<" ";

    }

    cout<<endl<<"请输入您要查找的那个数："<<endl;

    scanf("%d",&num); //要求用scanf输入

    for(;start<=End;)

    {

        middle=(start+End)/2;

        if(num<array[middle]){

            End=middle-1;

        }else if(num>array[middle]){

            start=middle+1;

        }else if(array[middle]==num){

            cout<<"恭喜您，找到了！  待查找数就是"<<array[middle]<<endl;

            break;

        }

    }

    if(array[middle]!=num)

    {

        cout<<"没找到！";

    }

      if(num<boundDown){

        cout<<"您输入的数太小，请重新输入"<<endl;

    }else if(num>boundUp){

        cout<<"您输入的数太大，请重新输入"<<endl;

    }

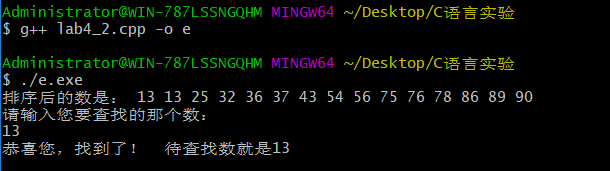
}

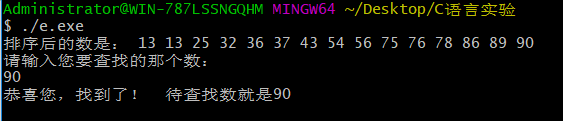
int main(){

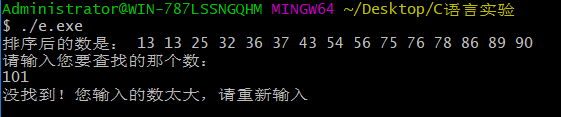
    reduceHalf();

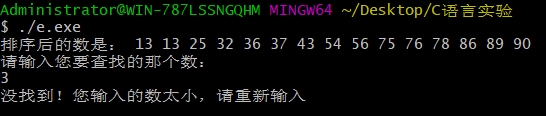
    return 0;

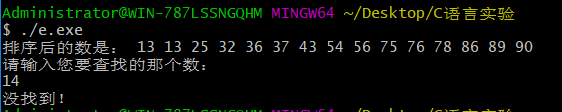
}











1. 对于某二维数组:

(1)找出每行的最大值;

(2)找出每列的最大值;

(3)找出二维数组的“鞍点”，即该位置上的元素在该行上最大，且在该列上最小。也可能没有鞍点。(P166习题8) 应当至少准备两组测试数据：

①二维数组有鞍点

9 80 205 40

90 -60 96 1

210 -3 101 89

②二维数组没有鞍点

9 80 205 40

90 -60 196 1

210 -3 101 89

45 54 156 7

用scanf函数从键盘输入数组各元素的值，检查结果是否正确。题目并未指定二维数组的行数和列数，程序应能处理任意行数和列数的数组。

如果已指定了数组的行数和列数，可以在程序中对数组元素赋初值，而不必用scanf函数。请同学们修改程序以实现之。

选)将两个字符串连接起来，不要用strcat函数。（P166习题13）

选) 任意输入一个英文句子(单词之间用空格隔开),编程输出其中每一个单词(每个单词一行).

选) 任意输入一个英文句子(单词之间用空格隔开),编程输出其中最长的单词.

#include<iostream>

#include<iomanip>

#include<vector>

using namespace std;

//指定二维数组的行数和列数

int arrayA[3][4]={{9,80,205,40},{90,-60,96,1},{210,-3,101,89} };

int arrayB[4][4]={{9,80,205,40},{90,-60,96,1},{210,-3,101,89},{45,54,156,7}};

//int tempmaxA[3],tempminA[4],tempmaxB[4],tempminB[4];

int flag=1,tempfindmax[3],tempfindmin[4];

void findMaxvalue(){

        cout<<"-------------------------------矩阵A----------------------------"<<endl;

        cout<<"arrayA:"<<endl;

        for(int i=0;i<3;i++){

            for(int j=0;j<4;j++){

                cout<<setw(4)<<arrayA[i][j];

            }

            cout<<endl;

        }

        int temp\_max;

        //int tempSaddle;

        for(int i=0;i<3;i++){

            temp\_max=arrayA[i][0];

            for(int j=1;j<4;j++){

            if(temp\_max<arrayA[i][j])

                {

                    temp\_max=arrayA[i][j];

                }

            //tempmaxA[i]=temp\_max;

            }

            cout<<i+1<<"行的最大值分别是："<<temp\_max<<"  "<<endl;

        }

        cout<<endl;

        for(int i=0;i<4;i++){

            temp\_max=arrayA[0][i];

            for(int j=1;j<3;j++){

                if(temp\_max<arrayA[j][i])

                {

                    temp\_max=arrayA[j][i];

                }

            }

            //tempminA[i]=temp\_max;

            cout<<i+1<<"列的最大值分别是："<<temp\_max<<"  "<<endl;

        }

        cout<<endl;

        cout<<"-------------------------------矩阵B----------------------------";

        cout<<endl;

        cout<<endl<<"arrayB:"<<endl;

        for(int i=0;i<4;i++){

            for(int j=0;j<4;j++){

                cout<<setw(4)<<arrayB[i][j];

            }

            cout<<endl;

        }

         for(int i=0;i<4;i++){

            temp\_max=arrayB[i][0];

            for(int j=1;j<4;j++){

            if(temp\_max<arrayB[i][j])

                {

                    temp\_max=arrayB[i][j];

                }

            }

            //tempmaxB[i]=temp\_max;   //tempmaxB存储的是每一行的最大值

            cout<<i+1<<"行的最大值分别是："<<temp\_max<<"  "<<endl;

        }

        cout<<endl;

        for(int i=0;i<4;i++){

            temp\_max=arrayB[0][i];

            for(int j=1;j<4;j++){

                if(temp\_max<arrayB[j][i])

                {

                    temp\_max=arrayB[j][i];

                }

            }

            //tempminB[i]=temp\_max;

            cout<<i+1<<"列的最大值分别是："<<temp\_max<<"  "<<endl;

        }

}

void findSaddle(){

        cout<<"----------------------找鞍点-----------------------------------"<<endl;

         for(int i=0;i<3;i++){

             tempfindmax[i]=arrayA[i][0];

            for(int j=0;j<4;j++){

                if(tempfindmax[i]<arrayA[i][j]){

                    tempfindmax[i]=arrayA[i][j];      //最大值放在tempfindmax  3

                }

            }

         }

          for(int i=0;i<4;i++){

             tempfindmin[i]=arrayA[0][i];

            for(int j=0;j<3;j++){

                if(tempfindmin[i]>arrayA[j][i]){     //最小值放在tempfindmin数组中  4

                    tempfindmin[i]=arrayA[j][i];

                }

            }

         }

        for(int i=0;i<3;i++){

            for(int j=0;j<4;j++){

                if(arrayA[i][j]==tempfindmax[i] && arrayA[i][j]==tempfindmin[j])

                {cout<<" 鞍点已经找到： "<<arrayA[i][j];

                flag=1;

                break;

                }

            }

        }

        if (flag==0)

            cout<<"鞍点没找到";

}

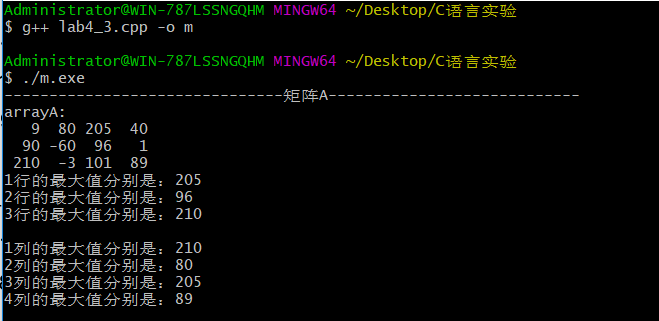
int main(){

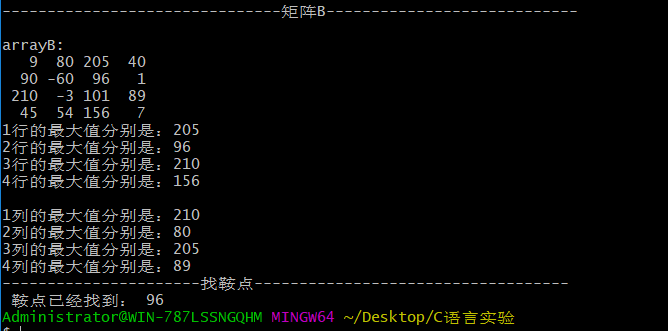
    findMaxvalue();

    findSaddle();

    return 0;

}





#include<iostream>

#include<iomanip>

#include<vector>

using namespace std;

//指定二维数组的行数和列数

int arrayA[3][4];

int arrayB[4][4];

//int arrayA[3][4]={{9,80,205,40},{90,-60,96,1},{210,-3,101,89} };

//int arrayB[4][4]={{9,80,205,40},{90,-60,96,1},{210,-3,101,89},{45,54,156,7}};

//int tempmaxA[3],tempminA[4],tempmaxB[4],tempminB[4];

int flag=1,tempfindmax[3],tempfindmin[4];

void inputArray(){

    cout<<"请输入矩阵A："<<endl;

    for(int i=0;i<3;i++){

        for(int j=0;j<4;j++){

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

        }

    }

    cout<<"请输入矩阵B："<<endl;

    for(int i=0;i<4;i++){

        for(int j=0;j<4;j++){

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

        }

    }

}

void findMaxvalue(){

        cout<<"-------------------------------矩阵A----------------------------"<<endl;

        cout<<"arrayA:"<<endl;

        for(int i=0;i<3;i++){

            for(int j=0;j<4;j++){

                cout<<setw(4)<<arrayA[i][j];

            }

            cout<<endl;

        }

        int temp\_max;

        //int tempSaddle;

        for(int i=0;i<3;i++){

            temp\_max=arrayA[i][0];

            for(int j=1;j<4;j++){

            if(temp\_max<arrayA[i][j])

                {

                    temp\_max=arrayA[i][j];

                }

            //tempmaxA[i]=temp\_max;

            }

            cout<<i+1<<"行的最大值分别是："<<temp\_max<<"  "<<endl;

        }

        cout<<endl;

        for(int i=0;i<4;i++){

            temp\_max=arrayA[0][i];

            for(int j=1;j<3;j++){

                if(temp\_max<arrayA[j][i])

                {

                    temp\_max=arrayA[j][i];

                }

            }

            //tempminA[i]=temp\_max;

            cout<<i+1<<"列的最大值分别是："<<temp\_max<<"  "<<endl;

        }

        cout<<endl;

        cout<<"-------------------------------矩阵B----------------------------";

        cout<<endl;

        cout<<endl<<"arrayB:"<<endl;

        for(int i=0;i<4;i++){

            for(int j=0;j<4;j++){

                cout<<setw(4)<<arrayB[i][j];

            }

            cout<<endl;

        }

         for(int i=0;i<4;i++){

            temp\_max=arrayB[i][0];

            for(int j=1;j<4;j++){

            if(temp\_max<arrayB[i][j])

                {

                    temp\_max=arrayB[i][j];

                }

            }

            //tempmaxB[i]=temp\_max;   //tempmaxB存储的是每一行的最大值

            cout<<i+1<<"行的最大值分别是："<<temp\_max<<"  "<<endl;

        }

        cout<<endl;

        for(int i=0;i<4;i++){

            temp\_max=arrayB[0][i];

            for(int j=1;j<4;j++){

                if(temp\_max<arrayB[j][i])

                {

                    temp\_max=arrayB[j][i];

                }

            }

            //tempminB[i]=temp\_max;

            cout<<i+1<<"列的最大值分别是："<<temp\_max<<"  "<<endl;

        }

}

void findSaddle(){

        cout<<"----------------------找鞍点-----------------------------------"<<endl;

         for(int i=0;i<3;i++){

             tempfindmax[i]=arrayA[i][0];

            for(int j=0;j<4;j++){

                if(tempfindmax[i]<arrayA[i][j]){

                    tempfindmax[i]=arrayA[i][j];      //最大值放在tempfindmax  3

                }

            }

         }

          for(int i=0;i<4;i++){

             tempfindmin[i]=arrayA[0][i];

            for(int j=0;j<3;j++){

                if(tempfindmin[i]>arrayA[j][i]){     //最小值放在tempfindmin数组中  4

                    tempfindmin[i]=arrayA[j][i];

                }

            }

         }

        for(int i=0;i<3;i++){

            for(int j=0;j<4;j++){

                if(arrayA[i][j]==tempfindmax[i] && arrayA[i][j]==tempfindmin[j])

                {cout<<" 鞍点已经找到： "<<arrayA[i][j];

                flag=1;

                break;

                }

            }

        }

        if (flag==0)

            cout<<"鞍点没找到";

}

int main(){

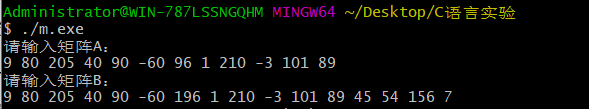
    inputArray();

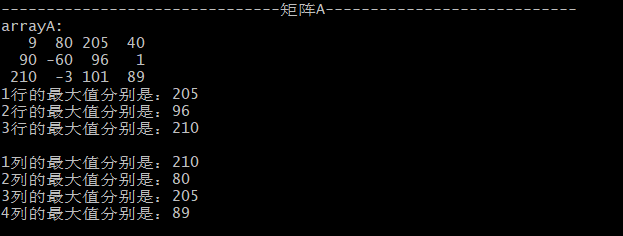
    findMaxvalue();

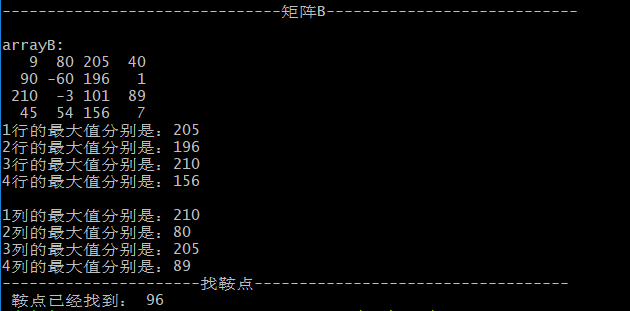
    findSaddle();

    return 0;

}







选)用冒泡法(又称起泡法)将任意8个整数按由小到大的顺序排列(参见教材P 143例)

#include<iostream>

using namespace std;

int num[8];

int temp;

void inputNum(){        //初始化数组

    cout<<"请输入8个整数："<<endl;

    for(int i=0;i<8;i++){

        cin>>num[i];

    }

}

void bubbleSort(){

    for(int i=0;i<8;i++){

        for(int j=0;j<8;j++){

            if(num[j]>num[j+1]){

                temp=num[j+1];

                num[j+1]=num[j];

                num[j]=temp;

            }

        }

    }

    cout<<"冒泡排序完成:"<<endl;

    for(int i=0;i<8;i++){

        cout<<num[i]<<" ";

    }

    cout<<endl;

}

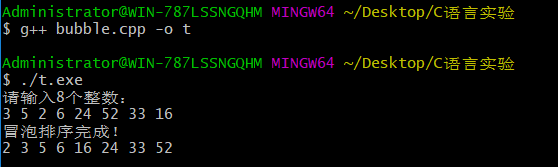
int main(){

    inputNum();

    bubbleSort();

    return 0;

}



【实验总结】

本次实验掌握了一维数组，字符数组以及与数组有关的排序算法，更加深入的了解了c语言以及c++模板库的使用方法。反复练习，日常巩固，有助于编程能力的提高。