《数据结构》课程实验报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 专业名称 | 计算机 | 年级 | 2017 | 班级 | 2 |
| 学生姓名 | 王汝芸 | 学号 | 201711010202 | 指导教师 | 郑志华 |
| 实验题目 | 实验1 顺序表的建立和运算 | | | 提交时间 | 2018．3.1 |

一、实验目的和要求

例如：（1）熟悉C语言的上机环境，进一步掌握C语言的结构特点。

（2）掌握线性表的顺序存储结构的定义及基本运算

二、实验环境

Visual Studio2017

三、实验内容及实施

采用较为简单的线性表；宏定义N为数组长度，方便后期修改；快速排序法；\*没按顺序实现问题

**实验1：顺序表的建立及运算**

实验要求：

1、建立一个顺序表，输入n个元素并输出；

2、查找线性表中的最大元素并输出；

3、在线性表的第i个元素前插入一个正整数x；

4、删除线性表中的第j个元素；

5、将线性表中的元素按升序排列；

\*6、将线性表中的元素就地逆序（只允许用一个暂存单元）；

**模块图**

建立n个元素的顺序表

查找最大元素

插入一个正整数

删除第j个元素

排序

1、建立一个顺序表，输入n个元素并输出；

2、查找线性表中的最大元素并输出；

3、在线性表的第i个元素前插入一个正整数x；

4、删除线性表中的第j个元素；

5、将线性表中的元素按升序排列；

\*6、将线性表中的元素就地逆序

**【源程序】**

/\*

1、建立一个顺序表，输入n个元素并输出；√

2、查找线性表中的最大元素并输出；√

3、在线性表的第i个元素前插入一个正整数x；√

4、删除线性表中的第j个元素；√

5、将线性表中的元素按升序排列；√

\*6、将线性表中的元素就地逆序（只允许用一个暂存单元）；√

\*/

#include<stdio.h>

#define N 10

//快速排序函数

void Quick\_sort(int a[], int left, int right)

{

int i = left, j = right;

int temp;

int pivot;//基准点

pivot = a[(left+right) / 2]; //基准点位置在中间

while (i <= j)

{

//从左到右找到大于等于基准点的元素

while (a[i] < pivot)

{

i++;

}

//从右到左找到小于等于基准点的元素

while (a[j] > pivot)

{

j--;

}

//如果i<=j，则互换

if (i <= j)

{

temp = a[i];

a[i] = a[j];

a[j] = temp;

i++;

j--;

}

}

//函数递归

if (left < j)

{

Quick\_sort(a, left, j);

}

if (i < right)

{

Quick\_sort(a, i, right);

}

}

//求最大值函数

int Max(int a[])

{

int max, i;

max = a[0];

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

{

if (a[i] >= max)

{

max = a[i];

}

}

return max;

}

int main()

{

int a[N];

int i;

//a赋值

printf("请输入%d个数:\n",N);

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

{

printf("请输入第%d个数:\n", i + 1);

scanf\_s("%d", &a[i]);

}

//输出a

printf("--------------------\n");

printf("\t这%d个数是：",N);

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

{

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

}

printf("\n--------------------\n");

int length;

length = sizeof(a) / sizeof (a[0]);

//原地逆序

int t;

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

{

t = a[i];

a[i] = a[N-1 - i];

a[N-1 - i] = t;

}

//输出逆序后结果

printf("\t逆序后结果为：");

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

{

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

}

printf("\n--------------------\n");

//求最大元素

int m;

m = Max(a);

printf("\t最大数是%d：", m);

printf("\n--------------------\n");

//快速排序

Quick\_sort(a, 0, length - 1);

//输出排序结果

printf("\t排序后的结果是：\n");

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

{

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

}

printf("\n--------------------\n");

//删除第j个元素

int j;

printf("\t删除第j个元素，请输入j：");

scanf\_s("%d", &j);

for (j; j < N; j++)

{

a[j-1] = a[j];

}

for (i = 0; i < N-1 ; i++)

{

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

}

printf("\n--------------------\n");

//在第i元素前插入x

printf("\t在第i个元素前插入正整数x\n请输入i：");

scanf\_s("%d", &i);

printf("请输入x：");

printf("\n--------------------\n");

int x;

scanf\_s("%d", &x);

j = i;

//顺序后移

for (i=0; i < N-j; i++)

{

a[N - 1 - i] = a[N - 2 - i];

}

a[j-1] = x;

printf("\t插入后结果为：");

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

{

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

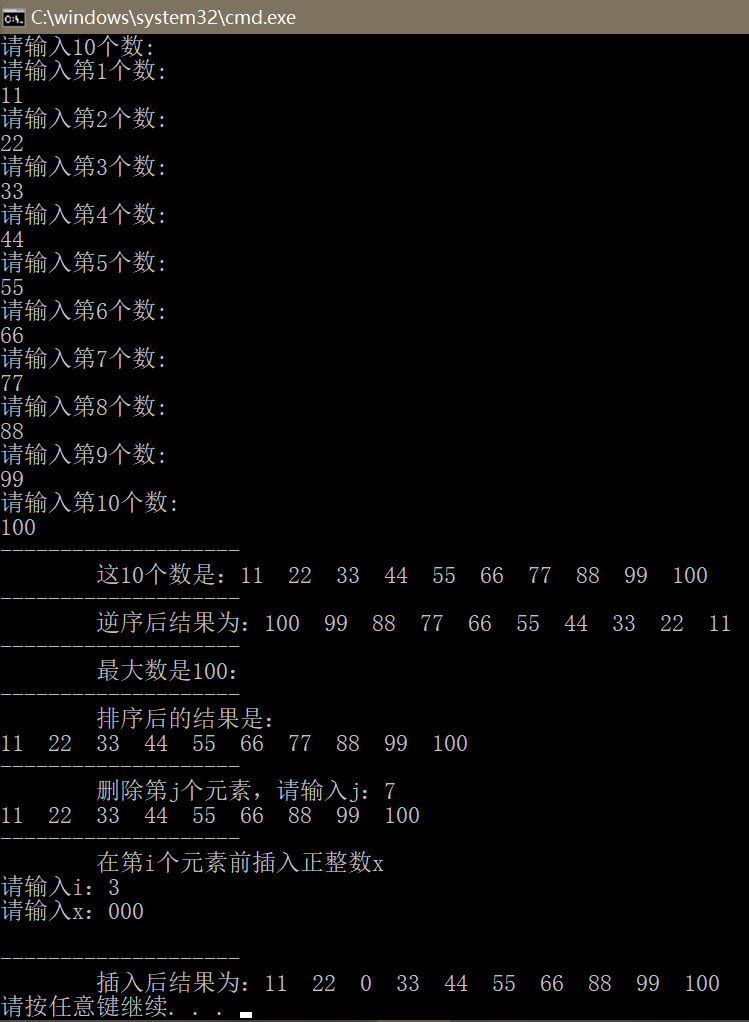
}

printf("\n");

return 0;

}

**四、实验结果 (程序的执行结果)**



**五、实验讨论（可选）**

无