

深圳大学实验报告

课程名称： 面向对象程序设计

实验项目名称： 实验一 数组排序

学院： 医学院

专业： 生物医学工程

指导教师： 李乔亮、邓云

报告人： 陈焕鑫 学号： 2016222042 班级： 生工 2 班

实验时间： 2018.9.13

实验报告提交时间： 2018.9.26

教务部制

实验题目：

- 1、编写函数对长度为 N 的数组进行从小到大排序。函数定义为：

```
void SortArray(int *array, int len);
```

- 2、在 Main 函数中，定义数组：

```
int a[5];
```

```
int b[10];
```

从键盘输入数组中个元素的值，并调用 SortArray 函数对数组 a, b 分别排序，在排序前后分别打印数组元素进行比较。

实验内容：

在实验设计的过程中，总共设计了五种排序的方法，分别是冒泡排序 (bubbleSortArray)，选择排序 (selectSortArray)，快速排序 (quickSortArray)、希尔排序 (shellSortArray) 和插入排序 (insertSortArray)。定义一个名为 MyArray 的类，该类是由总长度为 100，类型为 int 的数组构成的线性表，用来存放从键盘输入的数据。在这个类中，对流插入和流提取操作符进行了重载，使得可以直接使用运算符“<<”和“>>”对 MyArray 类对象进行操作。该类在构造函数中对数组进行初始化，将数组的元素全部置为 0，同时将 len 置为 0；清空数组函数的操作与初始化一样。

在 main 主函数中，声明了一个 MyArray 类变量 test，以及两个用来接收用户输入的 char 类型 code 和 keepon。进入程序后，会首先请求输入一个整数，每输入一个整数按下回车之后，会询问是否继续输入，如果继续输入，请按“y”，如果放弃输入，可以按 y 以外任意字符键，程序会自动输出未排序的数组元素。每次输入完一个数据就会提示数组中已有数据的数目。当放弃继续输入数据之后，就会自动调用一种排序方法（在 main 函数中，需要使用哪一种排序方法，就取消这一行的注释，并将其他排序函数注释掉）对数组进行排序，然后输入数组数据元素的数目以及排好序的数组。至此，程序将会询问是否要输入新的数组，如果是请输入“y”，否则输入 y 以外任意字符，并回车，程序结束。

实验环境与程序代码:

实验环境：win7 系统下的 Visual C++ 6.0

```
#include <iostream>

using namespace std;

#define ARRAY_SIZE 100    //定义数组的大小

void bubbleSortArray(int *array, int len);        //冒泡排序
void selectSortArray(int *array, int len);        //选择排序
void quickSortArray(int *array, int low, int high); //快速排序
void shellSortArray(int *array, int len);         //希尔排序
void insertSortArray(int *array, int len);        //插入排序

class MyArray{
    friend ostream& operator << (ostream &out, MyArray &obj); //重载流插入运算符
    friend istream& operator >> (istream &in, MyArray &obj); //重载流提取运算符
public:
    int seq[ARRAY_SIZE]; //对象的数组，数组长度为 ARRAY_SIZE
    int len;              //数组已使用长度

    MyArray()              //构造函数
    {
        for (int i = 0; i < ARRAY_SIZE; i++) //利用循环对数组所有元素进行初始化
        {
            this->seq[i] = 0;
        }
        len = 0;          //初始化后使用长度为 0
    }

    void clearArray(void) //清空数组
    {
        for (int i = 0; i < this->len; i++) //利用循环对数组所有元素进行初始化
        {
            this->seq[i] = 0;
        }
        len = 0;          //初始化后使用长度为 0
    }
};

ostream& operator << (ostream &out, MyArray &obj) //重载流插入运算符
{
    if (obj.len == 0) //如果对象数组里面并没有内容
    {
        out << "There is nothing in the array!" << endl; //输出错误信息
    }
}
```

```

        return out;          //返回
    }

    for (int i = 0; i < obj.len; i++) //逐个输出对象中的成员数组的元素
    {
        out << obj.seq[i] << " ";
    }

    return out;          //返回
}

istream& operator >> (istream &in, MyArray &obj) //重载流提取运算符
{
    in >> obj.seq[obj.len];    //将新元素加入对象数组中
    while(cin.fail())          //输入引发 cin 异常
    {
        cout << "Error. Please input again: "; //输出错误信息
        cin.clear();              //先清除异常状态
        cin.ignore(999, '\n');    //再清理缓冲区，清理 999 个字符或者碰到'\n'

        in >> obj.seq[obj.len];    //再次接受新元素
    }
    obj.len++;                  //数组长度加 1

    return in;
}

/*****
* 函数名称: main
* 函数功能: 主函数
* 输入参数: void
* 输出参数: void
* 返回值: int
*****/
int main()
{
    MyArray test;              //声明一个 MyArray 对象 test
    char code;                  //用于接收是否放弃输入元素的标志 (y 为继续输入, n 为结束输入)
    char keepon;

    do{
        do{
            cout << "Please enter a integer:";
            cin >> test;
            cout << "Now there are " << test.len << " in the array";

```

```

    cout << endl << "Do you still want to write?(y/n) ";
    cin >> code;
}while(code == 'y'); //如果用户输入 y, 继续等待用户的输入

cout << "The Array is like this: ";
cout << test << endl; //输出用户输入结束后的数组元素
//int a[] = {9,8,7,6,5,4,3,2,1};
//quickSortArray(test.seq, 0, test.len - 1); //快速排序
shellSortArray(test.seq, test.len); //希尔排序
//insertSortArray(test.seq, test.len); //插入排序
//bubbleSortArray(test.seq, test.len); //冒泡排序
//selectSortArray(test.seq, test.len); //选择排序

cout << "The length of the array is " << test.len << " in the end" << endl; //输出最终数
组的长度

cout << "The array after sorting is : " << test << endl; //输出排序后的数组元素

test.clearArray();

cout << "Do you want to creat a new array?(y/n) ";
cin >> keepon;
}while(keepon == 'y');

cout << "Thank you!";

return 0;
}

/*****
* 函数名称: bubbleSortArray
* 函数功能: 冒泡排序
* 输入参数: array: 数组; len: 数组长度
* 输出参数: void
* 返回值: void
*****/

void bubbleSortArray(int *array, int len)
{
    for(int i = 0; i < len - 1; i++)
    {
        int change = 1; //是否已具有顺序的标志

        for (int j = 0; j < len - 1 - i; j++)
        {
            if (*(array+j) > *(array+j+1)) //小数往上（前）浮，大数往下（后）沉
            {

```

```

        change = 0;                //发生过交换，表明元素还没排好序
        int temp = *(array+j);
        *(array+j) = *(array+j+1);
        *(array+j+1) = temp;
    }
}

if(change)    //如果循环中没有发生交换了，证明已有序，跳出循环
{
    break;
}
}

/*****
* 函数名称: selectSortArray
* 函数功能: 选择排序
* 输入参数: array: 数组; len: 数组长度
* 输出参数: void
* 返回值: void
*****/

void selectSortArray(int *array, int len)
{
    for(int i = 0; i < len - 1; i++)
    {
        int mini = 0;
        //从第 i 个开始，在第 i 个后面寻找最小值
        for(int j = i + 1, mini = i; j < len; j++)
        {
            if(*(array + j) < *(array + mini))
            {
                //找到更小的值了，将索引赋给 mini
                mini = j;
            }
        }
        //如果第 i 个元素后面有比 i 更小的值，即 mini 不等于 i
        if(mini != i)
        {
            //将更小的值和第 i 个元素值交换
            int temp = *(array + i);
            *(array + i) = *(array + mini);
            *(array + mini) = temp;
        }
    }
}

```

```

/*****
* 函数名称: quickSortArray
* 函数功能: 快速排序
* 输入参数: array: 数组; low: 较小的索引; high: 较大的索引
* 输出参数: void
* 返回值: void
*****/

void quickSortArray(int *array, int low, int high)
{
    if (low >= high)
    {
        return;
    }

    int first = low;
    int last = high;
    int key = array[first];

    while (first < last)
    {
        while(first < last && array[last] >= key)
        {
            --last;
        }

        array[first] = array[last];

        while(first < last && array[first] <= key)
        {
            ++first;
        }

        array[last] = array[first];
    }

    array[first] = key;
    quickSortArray(array, low, first - 1);
    quickSortArray(array, first + 1, high);
}

/*****
* 函数名称: InitTemplate
* 函数功能: 希尔排序
* 输入参数: array: 数组; len: 数组长度
* 输出参数: void
*****/

```

```

* 返回值: void
*****/

void shellSortArray(int *array, int len)
{
    if(len <= 1 || array == NULL)
    {
        return; //如果数组的长度小于等于 1 或者数组空了, 返回
    }

    for (int div = len / 2; div >= 1; div = div / 2) //每次都数组长度减半, 进行分组
    {
        for(int i = 0; i < div; i++) //对所分的每个组进行排序
        {
            for(int j = i; j < len - div; j += div)
            {
                for(int k = j; k < len; k += div)
                {
                    if (array[j] > array[k])
                    {
                        int temp = array[j];
                        array[j] = array[k];
                        array[k] = temp;
                    }
                }
            }
        }
    } //当 div 等于 1 时, 跳出循环

}

}

/*****

* 函数名称: insertSortArray
* 函数功能: 插入排序
* 输入参数: array: 数组; len: 数组长度
* 输出参数: void
* 返回值: void
*****/

void insertSortArray(int *array, int len)
{
    int temp;

    for (int i = 1; i < len; i++)
    {
        int j;
        temp = array[i]; //取出数组中第 i 个元素
    }

```



```

for(j = i - 1; j >= 0 && temp < array[j]; j--) //找到更大的数，在它之前插入
{
    array[j + 1] = array[j];                //元素后移，挪出适合 temp 的位置
}
array[j + 1] = temp;                        //将值插入适当的位置
}
}

```

实验结果与分析：

程序运行结果如下图所示：

```

H:\C++\LAB\LAB1>a.exe
Please enter a integer:16
Now there are 1 in the array
Do you still want to write?(y/n) y
Please enter a integer:98
Now there are 2 in the array
Do you still want to write?(y/n) y
Please enter a integer:45
Now there are 3 in the array
Do you still want to write?(y/n) y
Please enter a integer:12
Now there are 4 in the array
Do you still want to write?(y/n) y
Please enter a integer:17
Now there are 5 in the array
Do you still want to write?(y/n) n
The Array is like this: 16 98 45 12 17
The length of the array is 5 in the end
The array after sorting is : 12 16 17 45 98
Do you want to creat a new array?(y/n) y
Please enter a integer:19
Now there are 1 in the array
Do you still want to write?(y/n) y
Please enter a integer:43
Now there are 2 in the array
Do you still want to write?(y/n) y
Please enter a integer:99
Now there are 3 in the array
Do you still want to write?(y/n) y
Please enter a integer:2
Now there are 4 in the array
Do you still want to write?(y/n) y
Please enter a integer:55
Now there are 5 in the array
Do you still want to write?(y/n) y
Please enter a integer:50
Now there are 6 in the array
Do you still want to write?(y/n) y
Please enter a integer:36
Now there are 7 in the array
Do you still want to write?(y/n) y
Please enter a integer:465
Now there are 8 in the array
Do you still want to write?(y/n) y
Please enter a integer:258
Now there are 9 in the array
Do you still want to write?(y/n) y
Please enter a integer:1
Now there are 10 in the array
Do you still want to write?(y/n) n
The Array is like this: 19 43 99 2 55 50 36 465 258 1
The length of the array is 10 in the end
The array after sorting is : 1 2 19 36 43 50 55 99 258 465
Do you want to creat a new array?(y/n) n
Thank you!

```

指导教师批阅意见：

成绩评定：

指导教师签字：

年 月 日

备注：

- 注： 1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。