《数据结构与算法》实验报告

实验一：顺序表

**教 师：潘晔**

**学 生：梁书恺**

**学 号：2022040906023**

**时 间：9.18**

**地 点：科B119**

**一、ex1-1**

1. **问题分析**
2. **题意理解**

创建一个顺序表，用键盘对其进行数据读入、插入、删除等操作。

1. **数据结构设计**

typedef int ElemType;

typedef struct

{

    ElemType data[MAX];

    int length;

}SeqList;

1. **关键算法思路（画图或伪语言说明）**

创建：

输入：顺序表指针

输出：顺序表长度length

1. 键盘读入一个数字
2. 当数字不等于-1且length不超过max
   1. 写入顺序表的第length位
   2. length自增
   3. 读入一个数字
3. 输出length

插入：

输入：顺序表指针

输出：顺序表长度length

1. 键盘读入插入元素x
2. 键盘读入插入位置i
3. 出错处理
4. 从length-1向前遍历到i-1
   1. 元素后移
5. 在i-1处插入x
6. length自增
7. 输出length

删除：

输入：顺序表指针

输出：顺序表长度length

1. 键盘读入删除位置i
2. 错误处理
3. 从i遍历至length
   1. 元素前移
4. length自减
5. 输出length
6. **健壮性设计（说明出错处理方式）**

使用统一出错处理函数，在可能出错的位置进行判断，向出错处理函数传入出错代码，函数进行处理。

1. **性能分析（分析时间和空间复杂度）**

时间复杂度：O(n)

空间复杂度：O(1)

1. **测试数据与运行结果截图（设计合理全面的测试样例）：**

（最大长度MAX为10）

正常测试：

请输入顺序表中的元素，以-1结束

16 24 36 11 58 -1

顺序表中的元素为：16 24 36 11 58

请输入要插入的元素：87

请输入要插入的位置：3

顺序表中的元素为：16 24 87 36 11 58

请输入要删除的位置：2

顺序表中的元素为：16 87 36 11 58

插入、删除的边界测试：

请输入顺序表中的元素，以-1结束

16 24 36 11 58 -1

顺序表中的元素为：16 24 36 11 58

请输入要插入的元素：87

请输入要插入的位置：1

顺序表中的元素为：87 16 24 36 11 58

请输入要删除的位置：6

顺序表中的元素为：87 16 24 36 11

顺序表满时插入测试：

请输入顺序表中的元素，以-1结束

16 24 36 11 58 43 69 72 91 8 -1

顺序表中的元素为：16 24 36 11 58 43 69 72 91 8

请输入要插入的元素：56

请输入要插入的位置：4

顺序表已满，无法插入！

位置不合法测试：

请输入顺序表中的元素，以-1结束

16 24 36 11 58 -1

顺序表中的元素为：16 24 36 11 58

请输入要插入的元素：87

请输入要插入的位置：7

插入位置不合法！

1. **上机时遇到的问题**
   1. **问题现象：**exit()函数报错

**原因：**找不到函数定义

**解决办法：**包含<stdlib.h>头文件

* 1. **问题现象：**插入、删除时操作第一个、最后一个元素出现错误

**原因：** 边界条件设置有误

**解决办法：**修改边界条件

1. **程序代码**

seqList1.c

seqList1.h

**二、ex1-2**

1. **问题分析**
   1. **题意理解**

给定递增有序线性表，将元素x插入至表中。使其保持递增有序。

* 1. **数据结构设计**

同上

* 1. **关键算法思路**

输入：顺序表指针，插入元素x

输出：顺序表长度length

1. 将i从0遍历至length
   1. 如果x小于顺序表i位置的元素则跳出循环
2. 从length向前遍历至i
   1. 元素后移
3. 将x插入
4. length自增
5. 输出length
   1. **健壮性设计**

在插入前判断数组是否递增

* 1. **性能分析**

时间复杂度：O(n)

空间复杂度：O(1)

1. **测试数据与运行结果**

正常测试：

顺序表中的元素为：1 4 9 16 25 36 49 64 81 100

顺序表是递增的

请输入要插入的元素：50

顺序表中的元素为：1 4 9 16 25 36 49 50 64 81 100

边界测试：

顺序表中的元素为：1 4 9 16 25 36 49 64 81 100

顺序表是递增的

请输入要插入的元素：0

顺序表中的元素为：0 1 4 9 16 25 36 49 64 81 100

重合测试：

顺序表中的元素为：1 2 3 4 5 6 7 8 9 10

顺序表是递增的

请输入要插入的元素：5

顺序表中的元素为：1 2 3 4 5 5 6 7 8 9 10

非递增测试：

顺序表中的元素为：10 9 8 7 6 5 4 3 2 1

顺序表不是递增的

1. **上机时遇到的问题**
   1. **问题现象：**exit()函数报错

**原因：**找不到函数定义

**解决办法：**包含<stdlib.h>头文件

1. **程序代码**

seqList2.c

seqList2.h

**三、ex1-3**

1. **问题分析**
   1. **题意理解**

给定顺序表，将数据反序存放

* 1. **数据结构设计**

同上

* 1. **关键算法思路**
     1. 在原表逆序

输入：顺序表指针

输出：顺序表长度length

1. 建立新表L2
2. 将i从0遍历至length
   1. 将L表i位置的元素赋至L2的length-i-1位置
3. 将i从0遍历至length
   1. 将L2表i位置的元素赋至L表i位置
4. 输出length
   * 1. 额外使用一张新表

输入：顺序表指针

输出：顺序表长度length

1. 将i从0遍历至length的一半
   1. i和length-i-1的元素互换位置
2. 输出length
   1. **健壮性设计**

暂无

* 1. **性能分析**
     1. 方法一

时间复杂度O(n)

空间复杂度O(1)

* + 1. 方法二

时间复杂度O(n)

空间复杂度O(n)

1. **测试数据与运行结果**

偶数项：

顺序表中的元素为：1 4 9 16 25 36 49 64 81 100

逆置第一次

顺序表中的元素为：100 81 64 49 36 25 16 9 4 1

逆置第二次

顺序表中的元素为：1 4 9 16 25 36 49 64 81 100

奇数项：

顺序表中的元素为：1 2 3 4 5 6 7 8 9

逆置第一次

顺序表中的元素为：9 8 7 6 5 4 3 2 1

逆置第二次

顺序表中的元素为：1 2 3 4 5 6 7 8 9

1. **上机时遇到的问题**
   1. **问题现象：**顺序表只创建第一项

**原因：**sizeof()函数错误将指针作为参数

**解决办法：**将数组名作为参数

1. **程序代码**

seqList3.c

seqList3.h

**小结体会**

本次的实验内容为顺序表，主要进行了顺序表的创建、插入、删除、逆序等操作。本次实验较为简单，无论是数据的结构还是元素的操作都比较容易实现。在本次上机实验中，我的问题主要出现在边界条件上，出现了插入、删除第一个、最后一个元素出现问题的情况。在以后的实验中，要仔细考虑边界条件，减少问题的产生。