《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

1）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。

2）数据结构设计

```
typedef int ElemType;
typedef struct
{
    ElemType data[MAX];
    int length;
}SeqList;
```

3）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
   a) 写入顺序表的第 length 位
   b) length 自增
   c) 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
   a) 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
   a) 元素前移
4. length 自减
5. 输出 length

4）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。

5）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)

2. 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！
```

3. **上机时遇到的问题**
   a) **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
   b) **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

4. 程序代码

seqList1.c
seqList1.h

二、ex1-2

1. 问题分析

a) 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。

b) 数据结构设计

同上

c) 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
   a) 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
   a) 元素后移
3. 将 x 插入
4. length 自增
5. 输出 length

d) 健壮性设计

在插入前判断数组是否递增

e) 性能分析

时间复杂度：O(n)
空间复杂度：O(1)

2. 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
```

3. 上机时遇到的问题

a) **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

4. 程序代码

seqList2.c
seqList2.h

三、ex1-3

1. 问题分析

a) 题意理解

给定顺序表，将数据反序存放

b) 数据结构设计

同上

c) 关键算法思路

i. 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
   a) 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
   a) 将 L2 表 i 位置的元素赋至 L 表 i 位置
4. 输出 length

ii. 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
   a) i 和 length-i-1 的元素互换位置
2. 输出 length

d) 健壮性设计

暂无

e) 性能分析

i. 方法一

时间复杂度 O(n)
空间复杂度 O(1)

ii. 方法二
时间复杂度 O(n)
空间复杂度 O(n)

2. 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
```

3. 上机时遇到的问题

a) **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

4. 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。