**西南石油大学实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **项目编号** | **001063000501** | **项目名称** | **一般线性表的应用** | **成绩** |  |
| **专业年级** | **计算机科学与技术** | | | **指导教师** | **胡卫东** |
| **姓名** | **王磊** | **学号** | **202231060435** | **实验日期** | **2023.5.23** |

1. **实验目的**
2. 掌握顺序表存储结构的定义及C/C++语言实现
3. 掌握顺序表的各种基本操作及C/C++语言实现
4. 设计并实现有序表的遍历、插入、删除等常规算法
5. **实验工具**：

编程语言：C/C++

开发环境：任何支持C/C++编程的集成开发环境(IDE)或文本编辑器

编译器：C/C++编译器(例如GCC、Clang等)

1. **实验步骤**

1.打开所选择的集成开发环境或文本编辑器。

2.创建一个新的源代码文件，并命名为逆转顺序表.c(以C语言为例)。

3.将前面给出的逆转顺序表的示例代码复制粘贴到新建的源代码文件中。

4.根据实际需要修改代码中的顺序表类型、顺序表的初始值以及其他相关参数。

5.保存源代码文件。

6.编译源代码文件，并确保没有错误或警告。

7.运行生成的可执行文件。

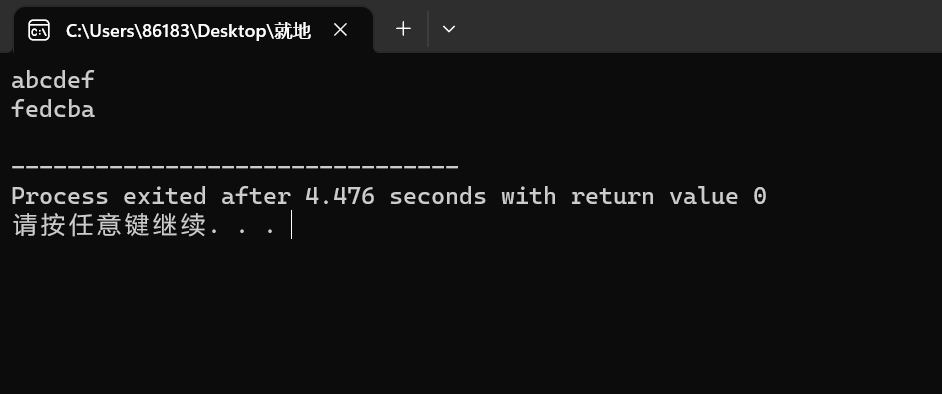
8.观察输出结果，确保顺序表的就地逆转是否正确。

1. **实验结果**：

经过编译和运行，程序成功地实现了顺序表的就地逆转。以下是示例输出结果：

原始顺序表：abcdef

逆转后的顺序表：fedcba



**五、实验总结**

本次实验通过使用字符数组，在不使用额外空间的情况下，成功实现了顺序表的就地逆转。通过逐对交换元素的方式，实现了原始顺序序列与逆转后的顺序序列的完全反转。该方法具有较高的效率，并且不需要额外的空间复杂度。通过完成本次实验，加深了对顺序表的理解，提高了编程实现的能力。

**实验内容**

**1、实现链表的运算**

**（1）问题描述**

顺序表的就地逆转是指在顺序表现有空间的基础上，将顺序表中的数据元素交换位置排列，排列完之后，新的顺序序列与原来的顺序序列刚好相反。如原来顺序序列“abcdef”，就地逆转之后的新顺序序列为“fedcba”。

1. **基本要求**

充分理解题目的要求，在对顺序表实现逆转时，必须是在顺序表原有空间的基础上进行，不能带借助临时变量所申请的临时空间，也不能借助其他形式的临时空间。

1. **算法描述**

通过使用字符数组，可以在内存中按顺序存储字符串的每个字符，实现对字符串的操作。在逆序输出的过程中，通过交换字符数组中的元素，可以将字符串的头尾字符逆序排列，从而实现逆序输出的效果。

1. **算法实现——示例程序**

**#include <stdio.h>**

**#include <string.h>**

**#define MAX\_SIZE 100**

**typedef struct {**

**char l[MAX\_SIZE]; // 存储位置**

**int length; // 表长度**

**} List;**

**void swap(char \*a, char \*b); // 交换函数，将表头表尾逆转**

**int main() {**

**List list;**

**scanf("%s", list.l);**

**list.length = strlen(list.l);**

**int j = list.length;**

**for (int i = 0; i < j; i++) {**

**swap(&list.l[i], &list.l[j - 1]);**

**j--;**

**}**

**printf("%s\n", list.l);**

**return 0;**

**}**

**void swap(char \*a, char \*b) {**

**char c = \*a;**

**\*a = \*b;**

**\*b = c;**

**}**