

软件学院综合性、设计性实验报告

学院： 软件学院 专业： 智能应用 年级/班级： 大二/2 班
2024—2025 学年第一学期

课程名称	数据结构	指导教师	张磊
学号姓名	2328624035 李陈歆烨		
实验地点	向真楼 403	实验时间	9 月 11 日
项目名称	线性表的顺序存储结构	实验类型	综合性

一、实验目的

- (1) 初始化一个线性表；
- (2) 创建一个包含 n 个正整数值的线性表（线性表的长度 n 和表中元素的值随机输入）；
- (3) 将一个数 x 插入在第 i 个元素前（x 和 i 在程序运行时输入）；
- (4) 删除第 i 个元素（i 在程序运行时输入），并在删除结束后输出删除元素的值；
- (5) 查找指定元素 e 是否在线性表中存在，若存在返回此元素的位序，否则返回 0；
- (6) 查找指定位置元素的值并输出；
- (7) 输出线性表中所有元素。

用主函数调用你所编写的函数，并在使线性表有所变化的每一步输出线性表的内容，以验证你编程序的正确性。

二、实验仪器或设备

计算机 C++DEV 程序

三、实验说明（设计方案）

- (1) 要求编一菜单，根据选项逐个调用各函数；
- (2) 所编程序要具有一定的健壮性，即：在插入删除时要考虑表空、表满、位置是否合法等情况。
- (3) 当输入数据非法时，程序也能适当的做出反应，而不致于出现莫名其妙的结果。
- (4) 根据老师提供的代码，根据用户需求编写主程序完成整个代码

四、实验步骤（包括主要步骤、代码分析等）

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define LIST_INIT_SIZE 100 // 线性表存储空间的初始分配量

typedef int ElemType; // 假设 ElemType 为 int 类型

typedef struct {
    ElemType *elem; // 存储空间基址
```

```

    int length; // 当前长度
} SqList;

// 1. 初始化线性表
void InitList(SqList *L) {
    L->elem = (ElemType *)malloc(LIST_INIT_SIZE *
sizeof(ElemType)); // malloc 的语法是：指针名 = (数据类型*) malloc (长度),
(数据类型*) 表示指针.
    if (!L->elem) exit(0); // 存储分配失败
    L->length = 0;
}

// 2. 创建一个包含 n 个正整数值的线性表
void CreateList(SqList &L, int n) {
    L.length = n;
    for (int i = 0; i < n; i++) {
        L.elem[i] = rand() % 100 + 1; // 随机生成 1 到 100 之间的整数
    }
}

// 3. 将一个数 x 插入在第 i 个元素前
void Insert(SqList &L, int x, int i) {
    if (i < 1 || i > L.length + 1 || L.length == LIST_INIT_SIZE) {
        printf("插入位置不合法或线性表已满\n");
        return;
    }
    for (int j = L.length; j >= i; j--) {
        L.elem[j] = L.elem[j - 1];
    }
    L.elem[i - 1] = x;
    L.length++;
}

// 4. 删除第 i 个元素，并输出删除元素的值
int Delete(SqList &L, int i) {
    if (i < 1 || i > L.length) {
        printf("删除位置不合法\n");
        return -1;
    }
    int deletedValue = L.elem[i - 1];
    for (int j = i; j < L.length; j++) {
        L.elem[j - 1] = L.elem[j];
    }
    L.length--;
}

```

```

        return deletedValue;
    }

// 5. 查找指定元素 e 是否在线性表中存在
int Locate(SqList L, int e) {
    for (int i = 0; i < L.length; i++) {
        if (L.elem[i] == e) {
            return i + 1; // 返回元素的位序
        }
    }
    return 0; // 元素不存在
}

// 6. 查找指定位置元素的值并输出
void PrintElement(SqList L, int i) {
    if (i < 1 || i > L.length) {
        printf("位置不合法\n");
        return;
    }
    printf("元素的值是: %d\n", L.elem[i - 1]);
}

// 7. 输出线性表中所有元素
void PrintList(SqList L) {
    for (int i = 0; i < L.length; i++) {
        printf("%d ", L.elem[i]);
    }
    printf("\n");
}

int main() {
    srand((unsigned int)time(NULL)); // 设置随机数种子
    SqList L;
    InitList(&L);
    int n, x, i, e, pos;

    // 创建线性表
    printf("请输入线性表的长度: ");
    scanf("%d", &n);
    CreateList(L, n);

    // 插入元素
    printf("请输入要插入的数和位置 (i x): ");
    scanf("%d %d", &i, &x);

```

```

Insert(L, x, i); //insert(插入)

// 删除元素
printf("请输入要删除的位置: ");
scanf("%d", &i);
int deleted = Delete(L, i);
if (deleted != -1) {
    printf("删除的元素是: %d\n", deleted);
}

// 查找元素
printf("请输入要查找的元素: ");
scanf("%d", &e);
pos = Locate(L, e);
if (pos) {
    printf("元素%d 在线性表中的位置是: %d\n", e, pos);
} else {
    printf("元素%d 不在线性表中\n", e);
}

// 查找指定位置的元素
printf("请输入要查找的位置: ");
scanf("%d", &i);
PrintElement(L, i);

// 输出线性表
printf("线性表中的所有元素是: ");
PrintList(L);

// 释放内存
free(L.elem);
return 0; //当内存不再使用时, 应使用 free() 函数将内存块释放。
}

```

五、结果分析与总结

```
C:\Users\LENOVO\Desktop\实验 X + v
请输入线性表的长度：6
请输入要插入的数和位置 (i x)：2 5
请输入要删除的位置：3
删除的元素是：100
请输入要查找的元素：2
元素2不在线性表中
请输入要查找的位置：7
位置不合法
线性表中的所有元素是：25 5 64 76 41 76

-----
Process exited after 16 seconds with return value 0
请按任意键继续. . .
```

根据用户页面提示输入线性表长度 删除 查找 显示所有元素，结果正确

2024 年 9 月 12 日