**南京师范大学中北学院**

**《数据结构》**

**实**

**验**

**报**

**告**

**姓 名: 钟玮**

**学 号: 85213241**

**班 级: 计算机二班**

**日 期: 2024年3月4日**

**顺序表的基本操作2**

1. **实验目的**

**1、掌握线性表的定义；**

**2、掌握线性表的基本操作，如建立、插入、查找、删除及合并等。**

1. **实验内容**

**1、在顺序线性表L中删除第i个元素，并用e返回其值；**

**2、在顺序线性表L中查找第1个值与e相等元素的位序；**

**3、已知顺序线性表La和Lb的元素按值非递减排列，归并La和Lb得到新的顺序线性表Lc，Lc的元素也按值非递减排列。**

1. **实验主要代码**

**可以截图并做简要解释，如：**

1. **在顺序线性表L中删除第i个元素，并用e返回其值**

**Status ListDelete\_Sq(SqList &L, int i, ElemType &e)**

**{**

**// i值不合法**

**if (i < 1 || i > L.length)**

**return ERROR;**

**ElemType \*p = &(L.elem[i - 1]); // p为被删除元素的位置**

**e = \*p; // 被删除元素的值赋给e**

**ElemType \*q = L.elem + L.length - 1;**

**// 被删除元素之后的元素左移**

**for (++p; p <= q; ++p)**

**\*(p - 1) = \*p;**

**--L.length; // 表长减1**

**return OK;**

**}**

1. **在顺序线性表L中查找第1个值与e相等元素的位序**

**int LocateElem\_Sq(SqList &L, ElemType e)**

**{**

**int i = 1; // i的初值为第1个元素的位序**

**ElemType \*p = L.elem;**

**// 从第1个元素开始，依次与e比较**

**while (i <= L.length && \*p != e)**

**{**

**i++;**

**p++;**

**}**

**if (i <= L.length)**

**return i;**

**else**

**return 0;**

**}**

1. **已知顺序线性表La和Lb的元素按值非递减排列，归并La和Lb得到新的顺序线性表Lc，Lc的元素也按值非递减排列**

**void MergeList\_Sq(SqList La, SqList Lb, SqList &Lc)**

**{**

**int i = 0, j = 0, k = 0;**

**Lc.length = La.length + Lb.length;**

**Lc.elem = (ElemType \*)malloc(Lc.length \* sizeof(ElemType)); // 分配内存**

**// 合并La和Lb到Lc**

**while (i < La.length && j < Lb.length)**

**{**

**if (La.elem[i] <= Lb.elem[j])**

**{**

**Lc.elem[k++] = La.elem[i++];**

**}**

**else**

**{**

**Lc.elem[k++] = Lb.elem[j++];**

**}**

**}**

**// 如果La还有剩余元素，复制到Lc**

**while (i < La.length)**

**{**

**Lc.elem[k++] = La.elem[i++];**

**}**

**/ / 如果Lb还有剩余元素，复制到Lc**

**while (j < Lb.length)**

**{**

**Lc.elem[k++] = Lb.elem[j++];**

**}**

**}**

**// 添加一个函数来对顺序线性表进行排序**

**void SortList\_Sq(SqList &L)**

**{**

**for (int i = 0; i < L.length - 1; i++)**

**{**

**for (int j = i + 1; j < L.length; j++)**

**{**

**if (L.elem[i] > L.elem[j])**

**{**

**ElemType temp = L.elem[i];**

**L.elem[i] = L.elem[j];**

**L.elem[j] = temp;**

**}**

**}**

**}**

**}**

1. **main函数**

**int main()**

**{**

**SqList l;**

**ElemType e1;**

**ElemType e2;**

**ElemType e3;**

**int loc1;**

**int loc2;**

**int loc3;**

**lnitList\_Sq(l);**

**// 插入5个随机元素**

**for (int i = 1; i <= 5; i++)**

**{**

**ListInsert\_Sq(l, i, rand() % 5);**

**}**

**ListTraverse\_Sq(l);**

**printf("当前线性表长度是：%d\n", GetLength\_Sq(l));**

**printf("请输入需要删除的位置：\n");**

**scanf("%d", &loc2);**

**ListDelete\_Sq(l, loc2, e2);**

**ListTraverse\_Sq(l);**

**printf("请输入需要查找的元素值：\n");**

**scanf("%d", &e3);**

**loc3 = LocateElem\_Sq(l, e3);**

**printf("元素%d的位置是：%d\n", e3, loc3);**

**SqList La, Lb, Lc;**

**lnitList\_Sq(La);**

**lnitList\_Sq(Lb);**

**for (int i = 1; i <= 5; i++)**

**{**

**ListInsert\_Sq(La, i, rand() % 5);**

**}**

**for (int i = 1; i <= 5; i++)**

**{**

**ListInsert\_Sq(Lb, i, rand() % 5);**

**}**

**SortList\_Sq(La); // 对La进行排序**

**SortList\_Sq(Lb); // 对Lb进行排序**

**printf("A中元素有：\n");**

**ListTraverse\_Sq(La);**

**printf("B中元素有：\n");**

**ListTraverse\_Sq(Lb);**

**printf("合并后的元素有：\n");**

**MergeList\_Sq(La, Lb, Lc);**

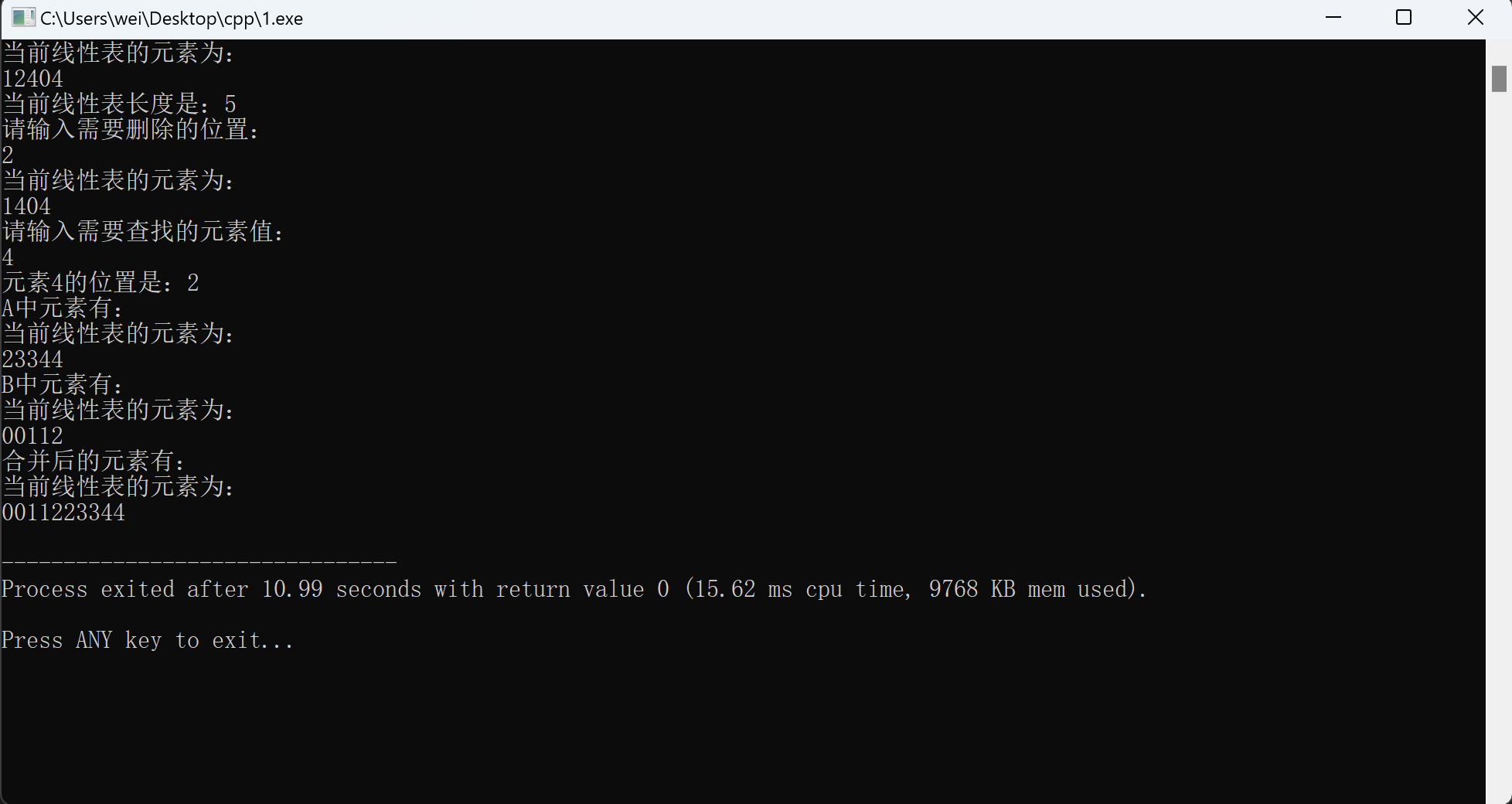
**ListTraverse\_Sq(Lc);**

**return 0;**

**}**

1. **实验结果**

**实验运行结果的截图，如main函数运行结果截图：**

****