

**实验报告**

**学院名称： 信息产业学院**

**专业名称： 计算机科学与技术**

**课程名称： 数据结构**

**班 级： 211060104 学号：21106010413**

**学生姓名： 刘钟泽**

**指导老师： 高俊**

2023年4月26日

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 学生姓名 | 刘钟泽 | | 学号 | 21106010413 | 实验成绩 |  |
| 实验项目名称 | | 用顺序表的建立和查找 | | | | |
| 实验地点 | | 博学楼c207 | | 实验日期 | 4月26日 | |
| 实验目的和要求  1. **实验目的：** 2. 掌握建立顺序表的基本方法。 3. 理解和掌握顺序表元素查找算法。 4. 掌握顺序表的插入、删除算法的思想和实现。 5. 实验要求**：** 6. 能够掌握顺序表的原理及其数据结构定义，并通过C语言编写程序实现基本操作； 7. 能够熟练运用数组的相关操作，理解数组在顺序表中的应用； 8. 能够熟悉顺序表的插入、删除、查找等基本操作原理，并能够编写程序解决相应问题； 9. 能够结合实际问题，将顺序表的实现运用到实际问题中，提高解决实际问题的能力； 10. 能够编写实验报告，充分总结实验内容和实验过程，反思实验结果，归纳实验心得。  **实验内容**  1. 建立一个顺序表，要求从键盘输入 10 个整数，并将该顺序表的元素从屏幕显示出来。 2. 编写查找函数，在上面的顺序表中查找其中一个元素，如果找到，返回该元素在顺序表中的位置和该元素的值，否则提示无此元素。要求被查找元素从键盘输入。 3. 编写插入和删除函数，由用户输入待插入元素及插入位置，将完成插入后的 4. 注意：其中顺序表的数据是随机产生，请考虑如何设计从键盘输入。 5. **实验过程**   定义了一个具有 MAXSIZE 个元素的结构体 variable SqList，其中 data 数组用于存储序号表的元素，而 length 则表示数组中实际存储的元素数量；   1. 设计 InitList 函数，用于初始化序号表，将 length 初始化为 0； 2. 设计 CreateList 函数，用于利用用户从键盘输入的值来进行序号表的初始化； 3. 设计 DispList 函数，用于从屏幕显示顺序表中的所有元素； 4. 设计 SearchList 函数，用于查找顺序表中是否存在指定元素； 5. 设计 InsertList 函数，用于在顺序表中插入指定元素； 6. 设计 DeleteList 函数，用于从顺序表中删除指定元素；  **实验源代码** #include <stdio.h>  #include <stdlib.h>  #define MAXSIZE 100  typedef struct {      int data[MAXSIZE];      int length;  } SqList;  void InitList(SqList \*L)  {      L->length = 0;  }  void CreateList(SqList \*L)  {      int i;      printf("请输入10个整数：\n");      for(i = 0; i < MAXSIZE; i++) {          scanf("%d", &L->data[i]);          L->length++;      }  }  void DispList(SqList L)  {      int i;      printf("顺序表中的元素为：\n");      for(i = 0; i < L.length; i++) {          printf("%d ", L.data[i]);      }      printf("\n");  }  int SearchList(SqList L, int key)  {      int i;      for(i = 0; i < L.length; i++) {          if(L.data[i] == key) {              printf("查找成功！该元素在顺序表中的位置为：%d，该元素的值为：%d\n", i + 1, L.data[i]);              return i;          }      }      printf("查找失败！顺序表中不存在该元素！\n");      return -1;  }  void InsertList(SqList \*L, int pos, int key)  {      int i;      if(pos < 1 || pos > L->length + 1) {          printf("插入位置不合法！\n");          return;      }      if(L->length == MAXSIZE) {          printf("顺序表已满，无法插入！\n");          return;      }      for(i = L->length - 1; i >= pos - 1; i--) {          L->data[i + 1] = L->data[i];      }      L->data[pos - 1] = key;      L->length++;      printf("插入成功！\n");  }  void DeleteList(SqList \*L, int pos)  {      int i;      if(pos < 1 || pos > L->length) {          printf("删除位置不合法！\n");          return;      }      for(i = pos; i < L->length; i++) {          L->data[i - 1] = L->data[i];      }      L->length--;      printf("删除成功！\n");  }  int main()  {      SqList L;      int choice, key, pos;      InitList(&L);        printf("请选择需要进行的操作：\n");      printf("1. 创建顺序表\n");      printf("2. 查找元素\n");      printf("3. 插入元素\n");      printf("4. 删除元素\n");      printf("5. 显示顺序表\n");      printf("0. 退出程序\n");        while(1) {          printf("请输入操作编号：");          scanf("%d", &choice);          switch(choice) {              case 1:                  CreateList(&L);                  break;              case 2:                  printf("请输入需要查找的元素：");                  scanf("%d", &key);                  SearchList(L, key);                  break;              case 3:                  printf("请输入待插入元素的值：");                  scanf("%d", &key);                  printf("请输入待插入位置：");                  scanf("%d", &pos);                  InsertList(&L, pos, key);                  break;              case 4:                  printf("请输入待删除位置：");                  scanf("%d", &pos);                  DeleteList(&L, pos);                  break;              case 5:                  DispList(L);                  break;              case 0:                  exit(0);              default:                  printf("输入有误，请重新选择操作！\n");          }      }      return 0;  } 实验结果 | | | | | | |