**新疆政法学院**

**数据结构与算法**

**实验报告**

班 级： 计算计科学与技术4班

学 号： 2124030160

姓 名： 孙久猛

指导老师： 张家琦

学 期： 2022-2023第一学期

|  |  |  |  |
| --- | --- | --- | --- |
| **实验题目** | 实验一：练习有序数列的数据插入 | | |
| **实验时间** | **10:00** | **实验地点** | **东教0411** |
| **实验成绩** |  | **实验性质** | **□验证性 √设计性 □综合性** |
| 教师评价：  □算法/实验过程正确； □源程序/实验内容提交 □程序结构/实验步骤合理；  □实验结果正确； □语法、语义正确； □报告规范；  其他：  评价教师签名： | | | |
| 一、实验目的  1.理解有序数列数据插入的实现步骤及思想；  2.掌握顺序存储方式存放数据时的数据插入；  3.在顺序存储的基础上，理解并练习以链式存储方式存储数据的数据插入。  4、通过对比学习数组和链表插入数据时所用方法的不同，加深对链表的理解，为后面学习内容奠定基础。 | | | |
| **二、实验项目内容（实验题目）**  （一）有序数组的插入  1、通过建立数组来模拟数据的顺序存储。  2、对建立的数据依次插入按序列排好的数据。  3、插入要插入的数据。  4、输出新的数据序列。  （二）递增的整数序列链表的插入  1、建立一个链表。  2、对建立的链表依次插入按序列排好的数据。  3、插入要插入的数据。  4、输出新的数据序列。  **三、源程序（实验步骤/实验过程/算法）**  // Dome\_01.c 本次的数组插入数据，利用了空位填补前后比较。    #include<stdio.h>  void main() {  int a[11]={2,3,4,5,7,8,9,11,34,56};  int i,j,x;  printf("请输入需要插入的数据：\n");  scanf("%d",&x);  for(i=9; i>=0; i--) { //令i=9倒序，共11个数字，有9个空来填充  if(a[i]>x) //if判断当数组数大于输入的数值  a[i+1]=a[i]; //令近两项相等  else  break; //否则跳出  }  a[i+1]=x; //将输入的数赋值给数组中的下一位置的项  for(i=0;i<11;i++)  printf("%d ",a[i]); //输出结果  }  // Dome\_02.c 建立了空链表，将链表的数组中插入数值    #include<stdio.h> /\* EOF(=^Z或F6),NULL \*/  #include<math.h> /\* floor(),ceil(),abs() \*/  #define TRUE 1  #define FALSE 0  #define OK 1  #define ERROR 0  typedef int Status;  // Status是函数的类型,其值是函数结果状态代码，如OK等  typedef int ElementType;  typedef struct PtrToNode  {  ElementType data;  struct PtrToNode \*prior,\*next; //定义指针头尾  }PtrToNode,\*List;  Status InitList(List \*L)  { // 产生空的双向循环链表L  \*L=(List)malloc(sizeof(PtrToNode)); //赋值给指针链表  if(\*L)  {  (\*L)->next=(\*L)->prior=\*L;  return OK;  }  else  return OVERFLOW;  }  void ListTraverse(List L,void(\*visit)(ElementType))  {  // 由双链循环线性表L的头结点出发,正序对每个数据元素调用函数visit()  List p=L->next; // p指向头结点  while(p!=L)  {  visit(p->data);  p=p->next;  }  printf("\n");  }  void vd(ElementType c) //ListTraverse()调用的函数(类型一致)  {  printf("%d ",c);  }  Status ListInsert(List L, ElementType X) {  List Pre, Tmp;  Pre = L; // 前驱结点从头结点开始  while(Pre->next) {  if(X < Pre->next->data)  break; // 找到第一个比x大的结点  else  Pre = Pre->next;  }  Tmp =(List)malloc(sizeof(PtrToNode));  Tmp->data = X; //建立x结点  Tmp->next = Pre->next;  Pre->next = Tmp; // 将x插入Pre后一个位置  return OK;  }  void main()  {  List L; //创建链表  int i,n,j,d;  ElementType e;  InitList(&L);  printf("初始化链表依次输入：1,2,3,4,5,6,7,8,9\n");  for(i=1;i<=9;i++) //中共有9个结点，  ListInsert(L,i); //在第i个结点之前插入i  ListInsert(L,8); //在第8个结点之前插入8  ListTraverse(L,vd); //正序输出  }  **四、运行结果**  //Dome\_01    //Dome\_02 | | | |

日 期： 2022年9月28日