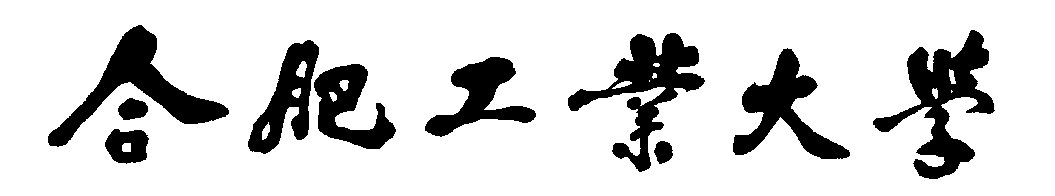
****

计算机与信息学院

数据结构实验报告

|  |  |
| --- | --- |
| 专 业 班 级 | 计算机科学与技术21-1 |
| 学生姓名及学号 | 杨程锦 2021214710 |
| 课程教学班号 |  |
| 任 课 教 师 | 胡学刚 |
| 实验指导教师 | 李培培 |
| 实验地点 | C201 |
| 2021 ~2022 学年第 二 学期 | |

说 明

实验报告是关于实验教学内容、过程及效果的记录和总结，因此，应注意以下事项和要求：

1．每个实验单元在4页的篇幅内完成一份报告。“实验单元”指按照实验指导书规定的实验内容。若篇幅不够，可另附纸。

2、各实验的**预习部分的内容是进入实验室做实验的必要条件**，请按要求做好预习。

3．实验报告要求：**书写工整规范，语言表达清楚，数据和程序真实。理论联系实际，认真分析实验中出现的问题与现象，总结经验。**

4．参加实验的每位同学应独立完成实验报告的撰写，其中程序或相关的设计图纸也可以采用打印等方式粘贴到报告中。严禁抄袭或拷贝，否则，一经查实，按作弊论取，并取消理论课考试资格。

5．实验报告作为评定实验成绩的依据。

实验序号及名称：实验 一 单链表实验

实验时间∶ 年 月 日

|  |
| --- |
| 预习内容 |
| 一、实验目的和要求∶  （1）理解线性表的链式存储结构。  （2）熟练掌握动态链表结构及有关算法的设计。  （3）根据具体问题的需要，设计出合理的表示数据的链  表结构，并设计相关算法。 |
| 二、实验任务∶  **设计算法实现下列问题的求解。**  **<1>求链表中第i个结点的指针（函数），若不存在，则返回NULL。**  实验测试数据基本要求：  第一组数据：链表长度n≥10，i分别为5，n，0，n+1，n+2  第二组数据：链表长度n=0，i分别为0，2  **<2>在第i个结点前插入值为x的结点。**  实验测试数据基本要求：  第一组数据：链表长度n≥10，x=100, i分别为5,n,n+1,0,1,n+2  第二组数据：链表长度n=0，x=100，i=5  **<3>删除链表中第i个元素结点。**  实验测试数据基本要求：  第一组数据：链表长度n≥10，i分别为5,n,1,n+1,0  第二组数据：链表长度n=0， i=5  **<4>在一个递增有序的链表L中插入一个值为x的元素，并保持其递增有序特性。**  实验测试数据基本要求：  链表元素为 （10,20,30,40,50,60,70,80,90,100）,  x分别为25，85，110和8  **<5>将单链表Ｌ中的奇数项和偶数项结点分解开，并分别连成一个带头结点的单链表，然后再将这两个新链表同时输出在屏幕上，并保留原链表的显示结果，以便对照求解结果。**  实验测试数据基本要求：  第一组数据：链表元素为 （1,2,3,4,5,6,7,8,9,10,20,30,40,50,60）  第二组数据：链表元素为 （10,20,30,40,50,60,70,80,90,100）  **<6>求两个递增有序链表L1和L2中的公共元素，并以同样方式连接成链表L3。**  实验测试数据基本要求：  第一组  第一个链表元素为 （1, 3, 6, 10, 15, 16, 17, 18, 19, 20）  第二个链表元素为 （1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 18, 20, 30）  第二组  第一个链表元素为 （1, 3, 6, 10, 15, 16, 17, 18, 19, 20）  第二个链表元素为 （2, 4, 5, 7, 8, 9, 12, 22）  第三组  第一个链表元素为 （）  第二个链表元素为 （1, 2, 3, 4, 5, 6, 7, 8, 9, 10） |
| 三、实验准备方案，包括以下内容：  （硬件类实验：实验原理、实验线路、设计方案等）  （软件类实验：所采用的核心方法、框架或流程图及程序清单）  链表的实现：提前写好了list.h文件，用node保存data和next指针，初始化头结点，然后执行插入操作。  **任务一：求链表中第i个结点的指针**    **任务二：i个结点前插入值为x的结点**    Append():    **任务三：删除链表中第i个元素结点。**      **任务四：在一个递增有序的链表L中插入一个值为x的元素，并保持其递增有序特性。**    **任务五**：**将单链表Ｌ中的奇数项和偶数项结点分解开**  构造两个链表分别存储原链表的奇数项和偶数项    **任务六: 求两个递增有序链表L1和L2中的公共元素，并以同样方式连接成链表。** |

|  |
| --- |
| 实验内容 |
| 一、实验用仪器、设备：  Legion R9000p，cLion |
| 二、实验内容与步骤（过程及数据记录）：  任务一：  void LinkList::SearchNode(int i) {  if(i <= 0 || i > length){  cout << "i已越界，无法查找" << endl;  return;  }  int k = 1;  Node \*p;  p = head->next;  while(k != i){  p = p->next;  k++;  }  cout << p->data << endl;  return;  }  1.    2.    任务二：  void LinkList::InsertNode(int i, int x) {  if(i > length + 1 || i <= 0){  cout << "i已越界，无法插入" << endl;  return;  }  Node \*p;  p = head->next;  if(i == 1){  Node \*s = new Node();  s->data = x;  s->next = head->next;  head->next = s;  length++;  }else{  for(int j = 1; j < i - 1;){  p = p->next;  j++;  }  Node \*s;  s = new Node();  s->data = x;  s->next = p->next;  p->next = s;  length++;  }  }              任务三：  void LinkList::DeleteNode(int i) {  if(i > length || i <= 0){  cout << "i已越界，无法查找" << endl;  return;  }  Node \*p = head->next;  for(int j = 1; j < i - 1; j++){  p = p->next;  }  Node \*s = p->next;  p->next = p->next->next;  delete s;  length--;  }      。  任务四：  void LinkList::InsertNodeRule(int x) {  Node \*p;  p = head->next;  if(p->data > x){  Node \*s = new Node();  s->data = x;  head->next = s;  s->next = p;  }  for(int i = 1; i <= length; i++) {  if (p->next != NULL) {  if (p->next->data >= x && p->data < x) {  Node \*s;  s = new Node();  s->data = x;  s->next = p->next;  p->next = s;  length++;  return;  }else{  p = p->next;  }  }else{  Node \*s;  s = new Node();  s->data = x;  p->next = s;  s->next = NULL;  length++;  return;  }  }  }    任务五：  void LinkList::DisconnectList() {  Node \*p = this->head->next;  int length = this->GetLength();  LinkList first,second;  first.length = 0;  second.length = 0;  Node \*p1 = first.head;  Node \*p2 = second.head;  for(int i = 1; i <= length ; i++){  Node \*s = new Node();  s->data = p->data;  if(i % 2 != 0){  p1->next = s;  s->next = NULL;  p1 = p1->next;  p = p->next;  first.length++;  continue;  }else{  p2->next = s;  s->next = NULL;  p2 = p2->next;  p = p->next;  second.length++;  continue;  }  }  cout << "奇数项数列：";  first.TravelList();  cout << endl << "偶数项数列：";  second.TravelList();  }    任务六：  void LinkList::ConnectList() {  LinkList l1, l2;  l1.CreatLinkList();  l2.CreatLinkList();  Node \*p1 = l1.head->next;  Node \*p2 = l2.head->next;  Node \*p = this->head;  if (l1.length != 0 && l2.length != 0){  for (int i = 1; i <= l1.length + l2.length; i++) {  if (p1 == NULL && p2 != NULL){  Node \*s = new Node();  s->data = p2->data;  s->next = NULL;  p->next = s;  p2 = p2->next;  p = p->next;  this->length++;  }  if (p2 == NULL && p1 != NULL){  Node \*s = new Node();  s->data = p1->data;  s->next = NULL;  p->next = s;  p1 = p1->next;  p = p->next;  this->length++;  }  if(p2 == NULL && p1 == NULL){  break;  }  if (p1->data >= p2->data) {  Node \*s = new Node();  s->data = p2->data;  s->next = NULL;  p->next = s;  p2 = p2->next;  p = p->next;  this->length++;  } else {  Node \*s = new Node();  s->data = p1->data;  s->next = NULL;  p->next = s;  p1 = p1->next;  p = p->next;  this->length++;  }  }  }else if (l2.length == 0){  for(int i = 0; i < l1.length; i++){  Node \*s = new Node();  s->data = p1->data;  s->next = NULL;  p->next = s;  p1 = p1->next;  p = p->next;  this->length++;  }  }else if (l1.length == 0){  for(int i = 0; i < l2.length; i++){  Node \*s = new Node();  s->data = p2->data;  s->next = NULL;  p->next = s;  p2 = p2->next;  p = p->next;  this->length++;  }  }  this->TravelList();  } |

|  |
| --- |
|  |
| 三、实验结果分析、思考题解答∶  实验结果符合预期，成功实现了带头结点的单链表的构造以及一系列算法的实现，掌握了动态链表结构及有关算法的设计。 |
| 四、感想、体会、建议∶  这是第一个构造的链式的数据结构，插入操作出了很多的问题，前插后插也搞了很久，删除操作出现的问题更多，常常误删了头结点，后来加入了各种判断，程序逐渐严谨。以后包括树和图的构造很多时候都基于链式的插入之类的操作，现在看比较基础，但放在一开始真的挺让人掉头发的。 |
| 实验成绩∶  指导教师签名：  年 月 日 |