

大组训练营作业01

目录


题目与要求
使用说明&所存在问题
整体设计思路分析
代码实现&测试结果（还是以原代码为主要）
思考&小结

题目与要求

- 题目：实现单链表ADT、实现双向链表ADT、周记一篇
- - 1.单链表奇偶调换（例：input: 1 -> 2 -> 3 -> 4 , output: 2 -> 1 -> 4 -> 3)
 - 2.找到单链表的中点
 - 3.判断链表是否成环
 - 4.反转链表（递归和非递归）
- 要求：按照项目工程结构开发，要有良好的交互设计、用户输入处理、规范的代码风格。
周记要使用Markdown语法，按照规定格式书写。
多写注释，让师兄师姐更方便的读懂你们的代码！

使用说明&所存在问题

- 链表系统功能界面
 - - 单链表系统功能界面
 - 双链表系统功能界面
- 1、所有界面选择和输入值都有防止输入错误，从而崩溃的处理。
 - - 都是用atoi（）处理int型数据。
 - 你会看到很多这样的提示：（字符当0处理）
 - 2、使用时都会有简单的提示，按提示输入即可。
 - 3、没有文件管理，退出即是删除。
 - 4、存在一个警告无法解决。

 C28182 取消对 NULL 指针的引用。“slow”包含与“fast”相同的 NULL 值。 大组训练营作业... Release.cpp 351

- 5、c++可能不是很熟练，代码可能偏向c（其实我也不知道怎么才算是c++）。

整体设计思路分析

- 首先是定义好单链表、双向链表的结构，做好框架。
- - 利用class函数包含所有想要实现的函数，存在于Headers.h文件中。
 - 在main函数中做好接口，调用。
 - 在Release.cpp文件中实现各个函数。
 - 同时在设计时注意可能存在的问题，反复检测，画图做分析。
 - 最后完成所有函数，进行优化，检测问题。
 - 没问题则返回检查，在需要的地方加上注释。
 - 完成后，则是功能扩写，会的都写上去。

各功能模块设计

- 1、单链表创建功能，双链表创建功能。
 - 简单的带头结点链表的创建
 - 2、插入数据功能
 - 单链表是利用查找数据，返回位置的函数，返回需想要的位置，从而实现前后插入。
 - 双链表则是利用其特性，可以找前面的结点，直接实现前插入。
 - 3、删除数据功能
 - 单链表和双链表删除几乎一样。找到数据，修改指向，释放。
 - 4、遍历功能
 - 循环输出，单链表和双链表一样。
 - 5、删除整个链表
 - 用两个指针，一个个释放。
 - 6、单链表奇偶调换
 - 需要一个带头结点的链表。
 - 利用三个指针指向1、2、3结点，然后排序前两个奇偶结点。
 - 头结点接回2结点（因为交换后，2在前面了），在将指向头结点的指针指向1。
 - 这样重复循环，到NULL就结束。
 - 7、单链表中点
 - 利用快慢指针（2:1），当快指针到最后，慢指针刚刚好到中。
 - 8、单链表判断成环
 - 同意利用快慢指针，两个指针如果在某时间一样，说明成环，否则不成环。
 - 9、单链表反转（递归和非递归）
 - 这里同时实现了三种方法
 - 迭代反转链表
 - 利用三个指针，第一个指向空，其他两个分别是1、2结点。
 - 第二个指向第一个，然后三个指针后移，循环这样操作即可。
 - 递归反转链表
 - 递归到最后，从后面两个两个的修改指向。从而反转。
 - 头插反转链表
 - 相当于创建一个新的链表，然后将原链表的数据一个个头插入新链表。
-

代码实现&测试结果（还是以原代码为主要）

• 1、单链表创建&双向链表创建

```
//创建带头结点单链表
LinkList p, r;
*L = new Node;
r = *L;

for (int i = 1; i <= atoi(num.c_str()); i++)
{
    p = new Node;
    //r跟上
    r->next = p;
    r = p;
}
r->next = NULL;

//创建带头结点双向链表
Dul_LinkList p, r;
*L = new Dul_Node;
(*L)->prior = NULL; //前指为空

r = *L;
for (int i = 1; i <= atoi(num.c_str()); i++)
{
    p = new Dul_Node;
    //双双指向
    r->next = p;
    p->prior = r;

    r = p;
}
r->next = NULL;
```

```
请输入您的选择: 1
请输入你要构建的结点数(字符当0处理): 3
请输入第1数据: 11
请输入第2数据: 12
请输入第3数据: 13
请按任意键继续. . .
```

• 2、插入功能

```
//单链表插入
if (select == "1") //前插入
{
    r = new Node;
    r->data = atoi(Data.c_str()); //赋值
    if (place == 1) //处理第一位，并且要插入在前面的情况
    {
        r->next = (*L)->next;
        (*L)->next = r;
        break;
    }
}
```

```

    }
    for (int i = 0; i < place - 2; i++)//找到想要插入的数据的前一个数据的位置
(做后插)
    {
        p = p->next;
    }
    //插入
    r->next = p->next;
    p->next = r;
    break;
}
else if (select == "2")//后插入
{
    r = new Node;
    r->data = atoi(Data.c_str());//赋值
    for (int i = 0; i < place-1; i++)
    {
        p = p->next;
    }
    //插入
    r->next = p->next;
    p->next = r;
    break;
}

//双向链表插入
if (select == "1")//前插入
{
    r = new Du1_Node;
    r->data = atoi(Data.c_str());//赋值

    r->prior = p->prior;
    p->prior->next = r;
    r->next = p;
    p->prior = r;
    break;
}
else if (select == "2")//后插入
{
    r = new Du1_Node;
    r->data = atoi(Data.c_str());//赋值

    if (p->next == NULL)//处理在最后的情况
    {
        r->next = p->next;
        r->prior = p;
        p->next = r;
        break;
    }
    r->prior = p;
    r->next = p->next;
    p->next->prior = r;
    p->next = r;
    break;
}
}

```

- ○ 测试 (两个一样)

请输入您的选择: 2
请输入想要插入那个数据的前后(字符当0处理): 12
1、前面 2、后面
请选择插入在前, 还是在后: 1
请输入想插入的数据(字符当0处理): 13
插入成功!
请按任意键继续. . .

- 3、删除数据功能

```
//单链表删除
while (p->data != atoi(Data.c_str()))
{
    q = p;
    p = p->next;
    if (p == NULL)
    {
        cout << "\t没有找到您想要删除的数据" << endl;
        system("pause");
        system("cls");
        return ERROR;
    }
}
//进行删除
q->next = p->next;
delete p;

//双向链表删除
if (p->next == NULL)//处理最后的情况
{
    //让前面结点指向空
    p->prior->next = NULL;
    delete p;
}

//进行删除
p->prior->next = p->next;
p->next->prior = p->prior;
delete p;
```

- ◦ 测试 (两个一样)

请输入您的选择: 3
请输入你想删除的数据(字符当0处理): 11
删除成功!
请按任意键继续. . .

- 4、遍历功能 (两个一样实现)

```

LinkedList p = *L;
p = p->next;
cout << "\t";
while (p != NULL)
{
    cout << p->data << "->";
    p = p->next;
}
cout << "NULL" << endl;

```

- ○ 测试

请输入您的选择: 4
13->12->13->NULL
请按任意键继续. . .
请输入您的选择: 4
13<->12<->13<->NULL
请按任意键继续. . .

- 5、删除整个链表 (两个实现一样)

```

LinkedList p, q;
p = (*L)->next; //让p指向第一个结点
while (p)
{
    q = p->next; //保留下一个结点位置
    delete p; //释放
    p = q;
}
(*L)->next = NULL;

```

- 6、单链表奇偶调换

```

LinkedList pre = *L;
while (pre->next != NULL && pre->next->next != NULL)
{
    //创建三个指针指向1、2、3个结点
    LinkedList L1 = pre->next;
    LinkedList L2 = pre->next->next;
    LinkedList L3 = L2->next;

    //进行交换next指向,重新排序前两个奇偶
    L1->next = L3;
    L2->next = L1;
    //带头结点的链表接回
    pre->next = L2;

    //pre移位
    pre = L1;
}

```

- ○ 测试

请输入您的选择: 4

11->12->13->14->15->NULL

请输入您的选择: 4

12->11->14->13->15->NULL

- 7、单链表中点

```
LinkedList fast = (*L)->next; //快指针
LinkedList slow = (*L)->next; //慢指针
while (fast != NULL && fast->next != NULL)
{
    slow = slow->next; //慢指针移动一个位置
    fast = fast->next->next; //快指针移动两个位置
}
```

- 测试

请输入您的选择: 7

查找到的中值为: 14

- 8、判断链表是否成环

```
LinkedList fast = *L; //快指针
LinkedList slow = *L; //慢指针
while (fast != NULL && fast->next != NULL)
{
    slow = slow->next; //移动一
    fast = fast->next->next; //移动二
    if (fast == slow) //成环则相等
    {
        cout << "\t链表成环! " << endl;
        system("pause");
        system("cls");
        return SUCCESS;
    }
}
```

- 测试

请输入您的选择: 9

没有成环!

- 9、反转链表 (递归和非递归)

- 迭代反转链表

```
//第一个指针
LinkedList beg = NULL;
//第二个指针
LinkedList mid = (*L)->next;
//第三个指针
LinkedList end = (*L)->next->next;
//遍历
while (true)
```

```

{
    //修改mid的指向
    mid->next = beg;

    if (end == NULL)
    {
        break;
    }

    //后移
    beg = mid;
    mid = end;
    end = end->next;
}
//最后修改头指向
(*L)->next = mid;

```

◦ 递归反转链表

```

LinkedList head = (*L)->next;
//递归反转链表
(*L)->next = recursive_reverse(head);

LinkedList recursive_reverse(LinkedList head)
{
    //当只有一个结点，就返回
    if (head->next == NULL)
    {
        return head;
    }
    else
    {
        //递归到最后
        LinkedList new_head = recursive_reverse(head->next);
        //改变指向
        head->next->next = head;
        head->next = NULL;

        return new_head;
    }
}

```

◦ 头插反转链表

```

LinkedList head = (*L)->next;
//头插反转链表
LinkedList new_head = NULL; //相当于新链表
LinkedList temp = NULL; //配合head一个个头插入new_head

while(head != NULL)
{
    temp = head;

    //后移head，保留下一个结点位置
    head = head->next;

    //更改指向

```



```
temp->next = new_head;
new_head = temp; //跟上
}
//接回L中
(*L)->next = new_head;
```

- ○ 测试

请输入您的选择: 4

11->12->13->14->15->NULL

请输入您的选择: 4

15->14->13->12->11->NULL

思考&小结

总体来说呢，因为训练营的作业，让我不在感觉每天无所事事。对于这次作业呢，我的收获可以说是非常大的。特别是在做训练营的作业的时候，刚刚开始对很多题目还是不知道是什么，都是随着去思考，在草稿本上写写画画，在一遍遍的推导中，慢慢的学会了一种思维，同时也感到很奇妙，特别是看着自己曾经没想过的代码，居然在我的电脑中成功运行起来，那是一种满满的成就感。怎么说呢，训练营让我有了明确的目标，但也不可否认，是真的很辛苦，没有假期。不过呢，还是那句话，这也让我学到了许多知识，像链表的奇偶调换、链表反转、快慢指针等等。收获很大，我也期待着能在图形学上能完成的更好，学的更多。