**数据结构实验报告**

|  |  |  |  |
| --- | --- | --- | --- |
| **第3次实验** | **学号：20191060064** | **姓名：白文强** | **得分：** |

**一、实验目的**

1、复习结构体、指针

2、掌握链表的创建、遍历操作

3、了解函数指针

**二、实验内容**

1、每个学生的成绩信息包括：学号、语文、数学、英语、总分、加权平均分；采用链表存储若干学生的成绩信息；输入学生的学号、语文、数学、英语成绩；计算学生的总分和加权平均分（语文占30%，数学占50%，英语占20%）；输出学生的成绩信息。

2、可以在链表末尾追加新学生的成绩信息；可以根据学号，删除该学生的成绩信息。

3、可以根据学号或总分，升序排序学生的成绩信息。

4、数据结构MOOC的第二周编程实训题第2题：2-2 单链表逆转

**三、数据结构及算法设计**

1、在头结点后插入n个带有学生信息的节点

2、单链表尾插法、利用strcmp函数找到输入的学号所在的节点，进行删除

3、每次固定一个节点，与后面的节点依次比较，若此节点小于后面的节点，就交换，固定节点后移，再开始一轮

4、单链表逆转算法

**四、核心程序代码（给出必要注释）**

1、

**void** **InsertList(LinkList** **\*L)**

**{**

**LinkList** **p** **=** **(LinkList)malloc(sizeof(LinkNode));**

**static** **LinkList** **tail;//静态定义一个尾指针**

**GetInput(&p);//给新节点赋值**

**if((\*L)->next** **==** **NULL)//如果是空链表**

**{**

**(\*L)->next** **=** **p;**

**p->next** **=** **NULL;**

**}**

**else//不是空链表**

**{**

**p->next** **=** **NULL;**

**tail->next** **=** **p;**

**}**

**tail** **=** **p;**

**}**

**typedef** **struct** **student**

**{**

**char** **ID[20];**

**double** **Chinese;**

**double** **Math;**

**double** **English;**

**double** **sum;**

**double** **weight\_average;**

**struct** **student** **\*next;**

**}** **LinkNode,** **\*LinkList;**

**printf("请输入人数\n");**

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

**for(i** **=** **0;** **i** **<** **n;** **i++)**

**{**

**InsertList(&L);**

**}//输入一开始的学生信息**

2、

**void** **InsertList(LinkList** **\*L)//尾插法**

**{**

**LinkList** **p** **=** **(LinkList)malloc(sizeof(LinkNode));**

**static** **LinkList** **tail;//静态定义一个尾指针**

**GetInput(&p);//给新节点赋值**

**if((\*L)->next** **==** **NULL)//如果是空链表**

**{**

**(\*L)->next** **=** **p;**

**p->next** **=** **NULL;**

**}**

**else//不是空链表**

**{**

**p->next** **=** **NULL;**

**tail->next** **=** **p;**

**}**

**tail** **=** **p;**

**}**

**void** **Delete(LinkList** **\*L,** **char** **input[])//删除**

**{**

**LinkList** **current** **=** **\*L;**

**LinkList** **previous** **=** **NULL;**

**while(current** **!=** **NULL** **&&** **strcmp(input,** **current->ID))//直到找到匹配的学号或者找不到该学号**

**{**

**previous** **=** **current;**

**current** **=** **current->next;**

**}**

**if(current** **==** **NULL)//未找到**

**{**

**printf("未找到此学号\n");**

**return;**

**}**

**else**

**{**

**previous->next** **=** **current->next;**

**free(current);//释放内存**

**}**

**}**

3、

**void** **LinkSort(LinkList** **\*L)**

**{**

**LinkNode** **temp;**

**LinkList** **pf** **=** **(\*L)->next;//第一个有元素的节点**

**LinkList** **pb;**

**if(pf** **==** **NULL)**

**{**

**printf("空链表，无需排序\n");**

**}**

**else** **if(pf->next** **==** **NULL)**

**{**

**printf("只有一个元素，无需排序:\n");**

**}**

**else**

**{**

**while(pf** **!=** **NULL)//pf先固定**

**{**

**pb** **=** **pf->next;**

**while(pb** **!=** **NULL)**

**{**

**if(pf->sum** **>** **pb->sum)//如果前面大于后面**

**{**

**//把两个节点的内容互换**

**temp** **=** **\*pb;**

**\*pb** **=** **\*pf;**

**\*pf** **=** **temp;**

**//再把next域换回来**

**temp.next** **=** **pf->next;**

**pf->next** **=** **pb->next;**

**pb->next** **=** **temp.next;**

**}**

**pb** **=** **pb->next;**

**}**

**pf** **=** **pf->next;**

**}**

**}**

**}**

4、

**List** **Reverse(** **List** **L** **)**

**{**

**List** **p** **=** **NULL,** **q** **=** **L;**

**while(q** **!=** **NULL)**

**{**

**L** **=** **q->Next;** **// 保留下一个节点**

**q->Next** **=** **p;**

**p** **=** **q;**

**q** **=** **L;** **// 移动到下一个节点**

**}**

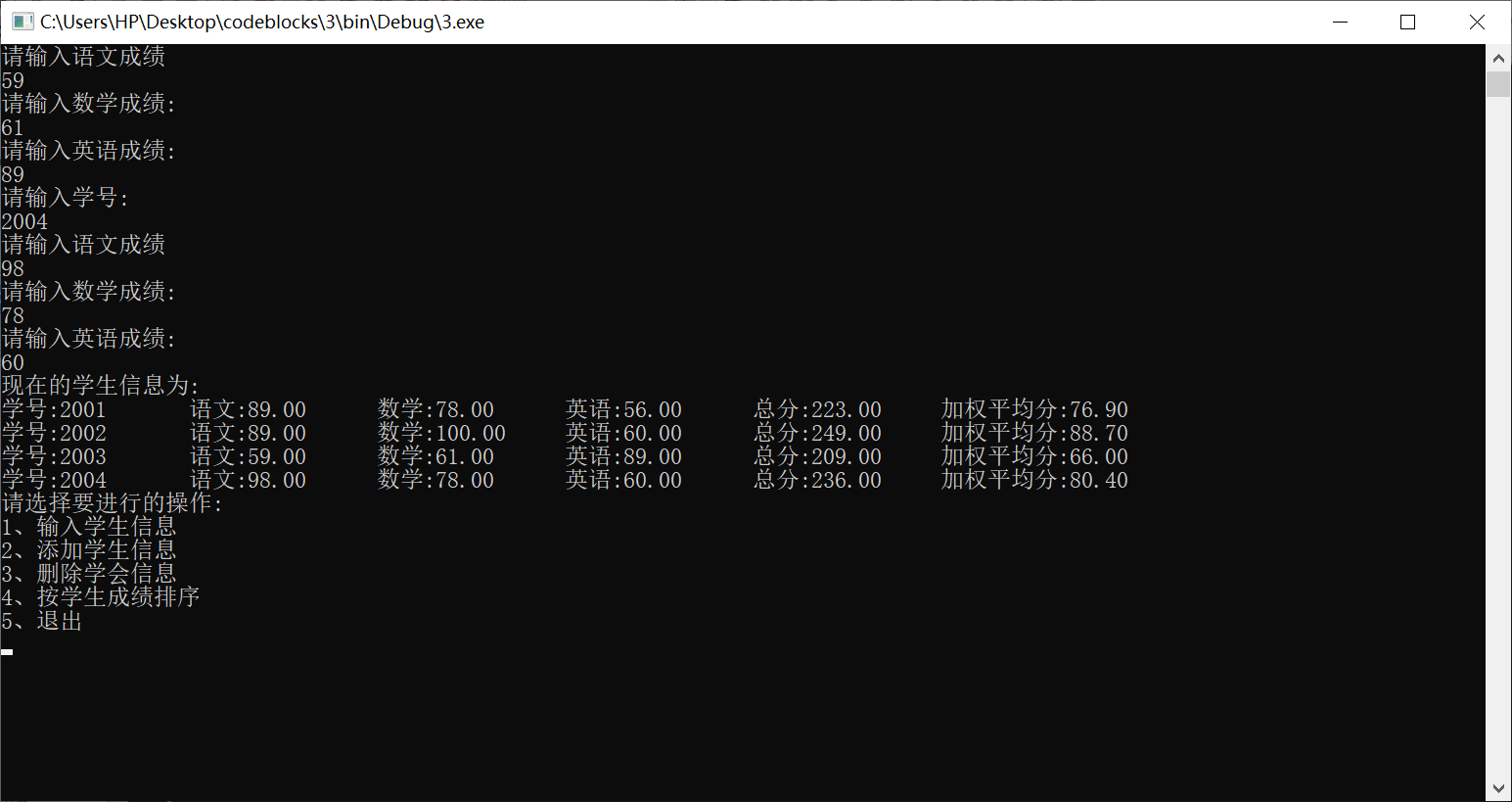
**L** **=** **p;** **// 指向逆置后的头节点**

**return** **L;**

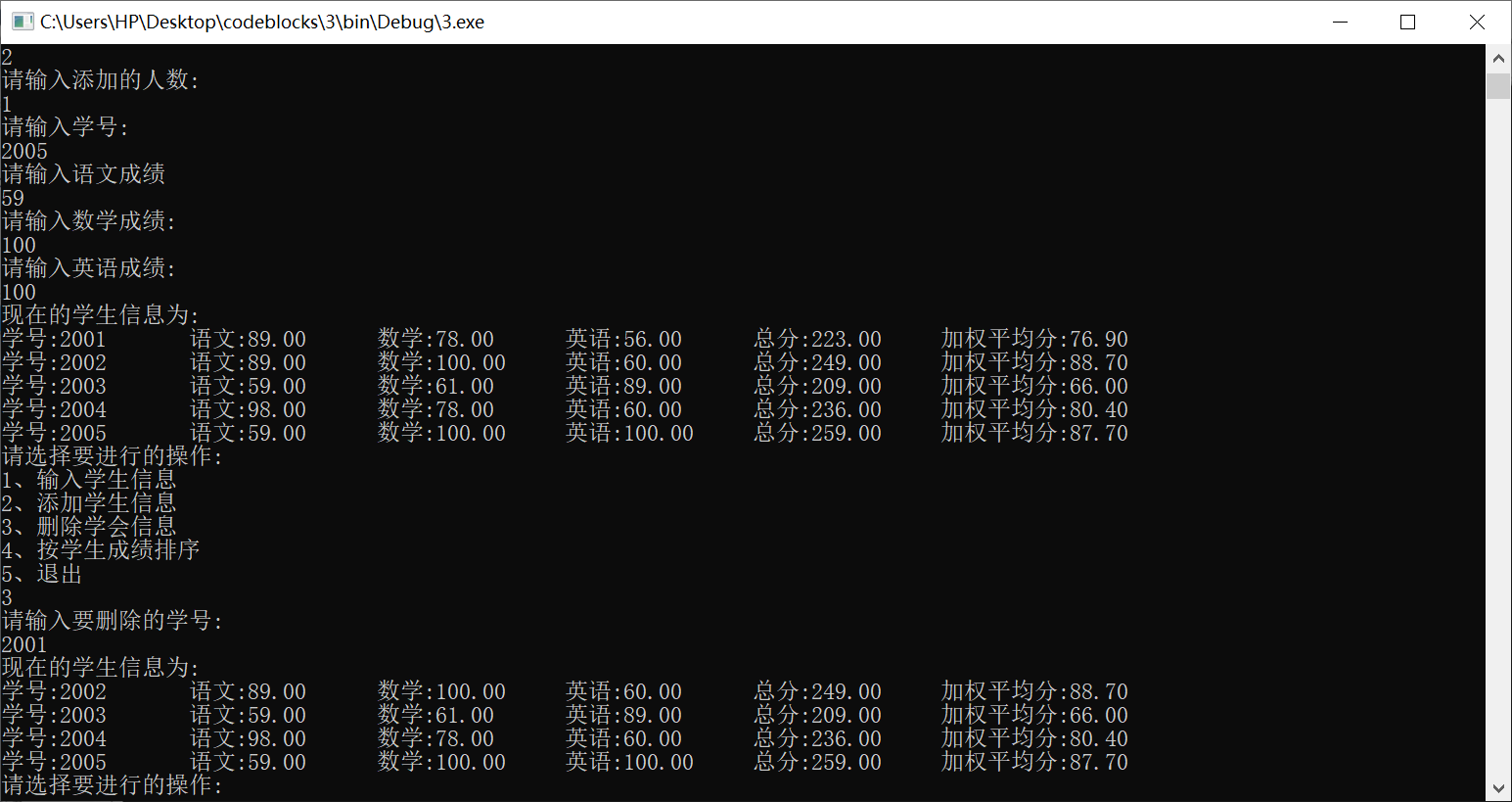
**}**

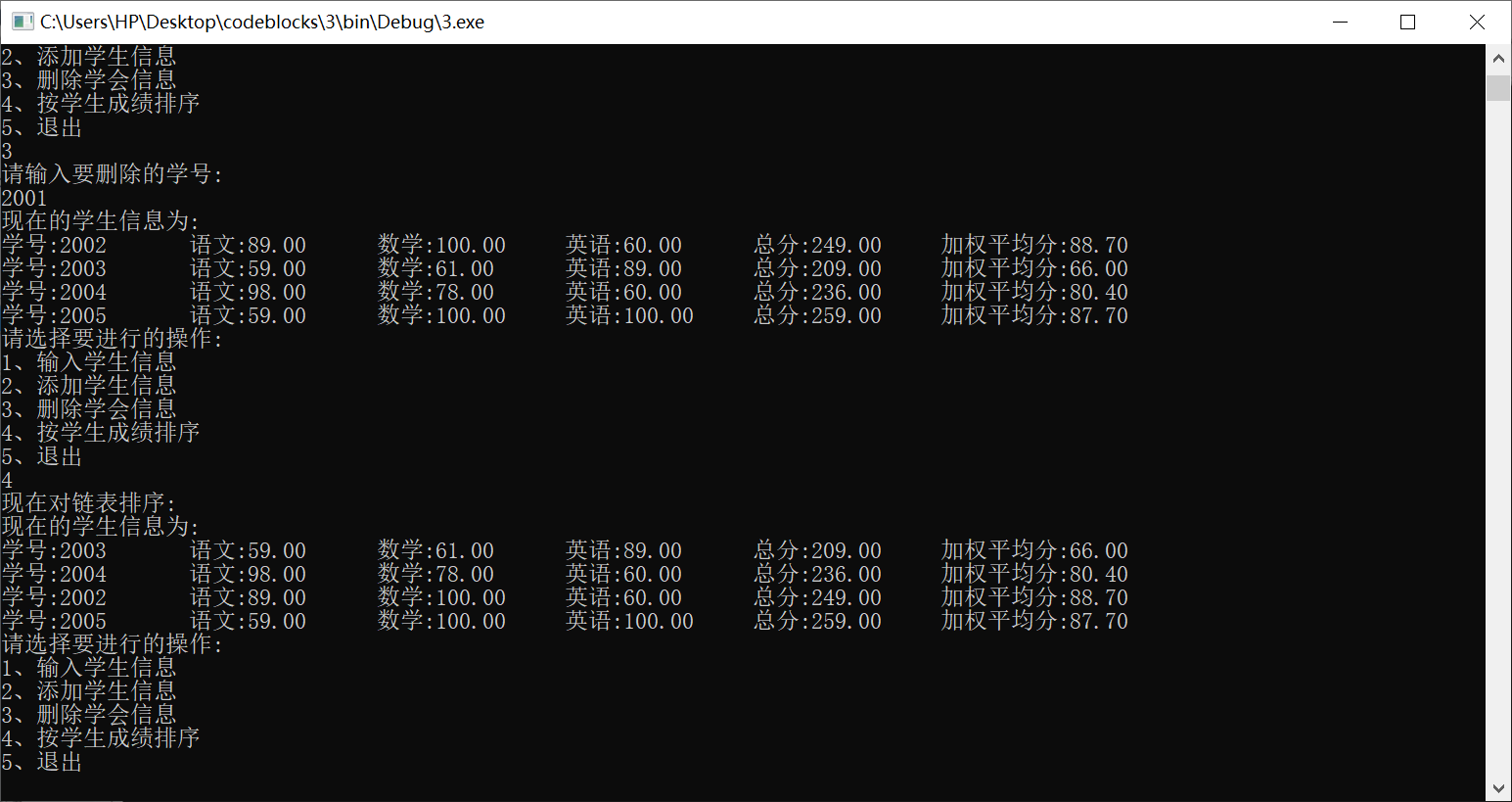
**五、测试及结果（给出测试用例及测试结果）**

1、



2、

3、



4、

