|  |
| --- |
| 哈尔滨工业大学(深圳) |
| **《数据结构》实验报告** |
|  |
| 实验一  线性结构及其应用  学 院: 计算机科学与技术   |  |  | | --- | --- | | 姓 名: | 苏亦凡 | | 学 号: | 200111229 | | 专 业: | 计算机科学与技术 | | 日 期: | 2021-03-30 | |

# 一、问题分析

（1）存储链表：用头插法将新节点插入链表。

（2）反转链表：反转两个链表的节点顺序。

（3）链表交点：寻找交叉链表的交点。

# 二、详细设计

## 2.1 设计思想

（1）储存链表：创建节点并从输入获取节点数值域的值，将这个节点连接在头指针与首元结点之间。由于给出的输入总是升序的，此链表在存储后成绩将是降序的。

（2）反转链表：令p指针始终指向它初始指向的节点。依次摘取此节点的next指针指向的节点并用（1）中的方法插入链表，直至p指针指向的节点的next指针为空。

（3）链表交点：遍历两个链表，获取两链表的长度，做出两链表的差值n，令两个操作指针指向两链表的头，较长链表的指针向前移动n步，之后两指针同时以1为步长前进，直到两指针值相同。

## 2.2 存储结构及操作

(1) 存储结构：单向链表

typedef struct node

{

char StuID[11];

int Grade;

struct node \*next;

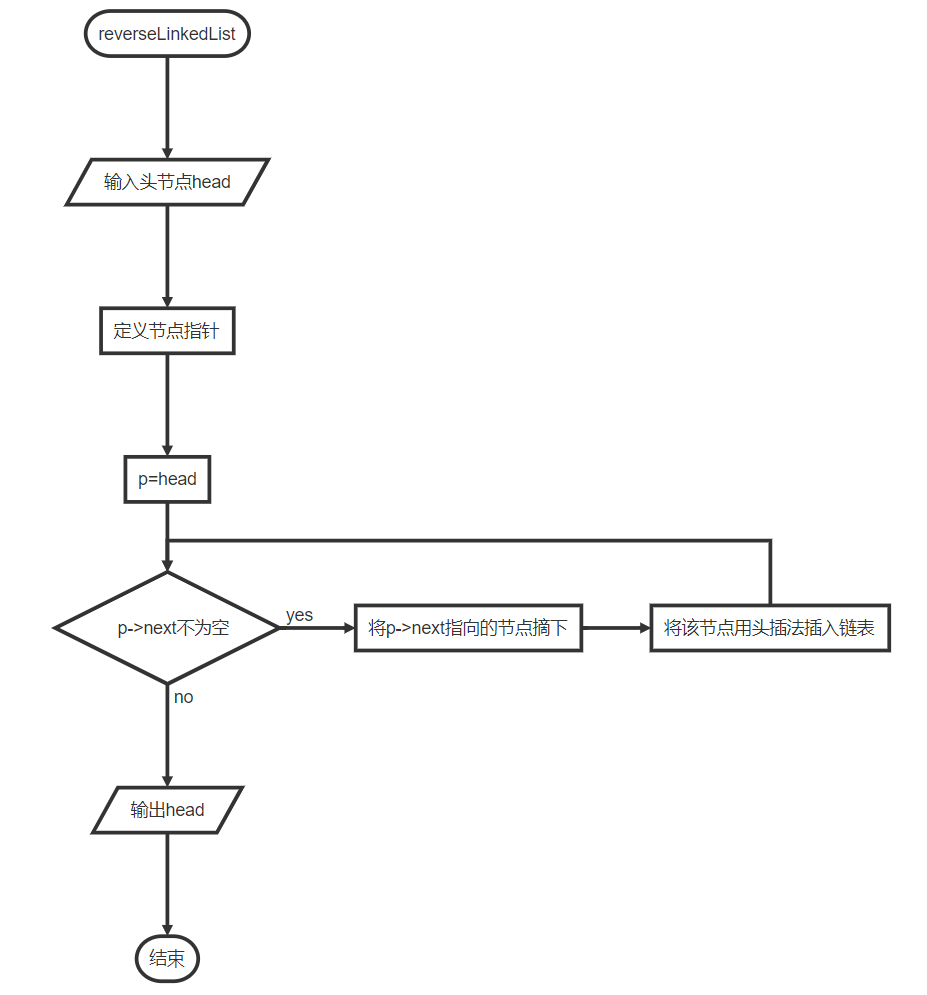
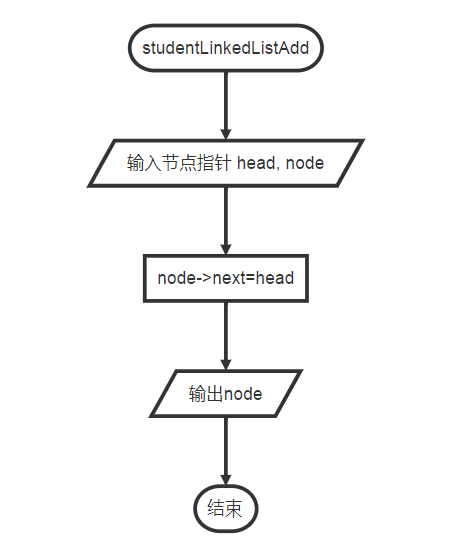
} StudentLinkedListNode;

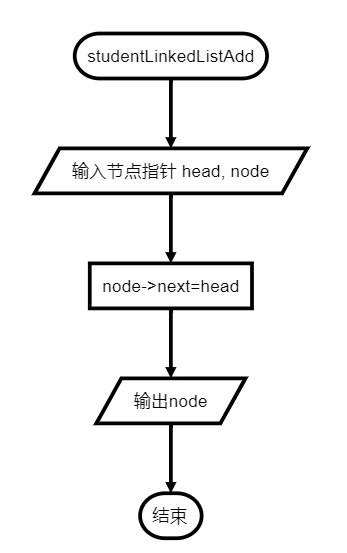
(2) 涉及的操作（一般为自定义函数，可不写过程，但要注明该函数的含义。）

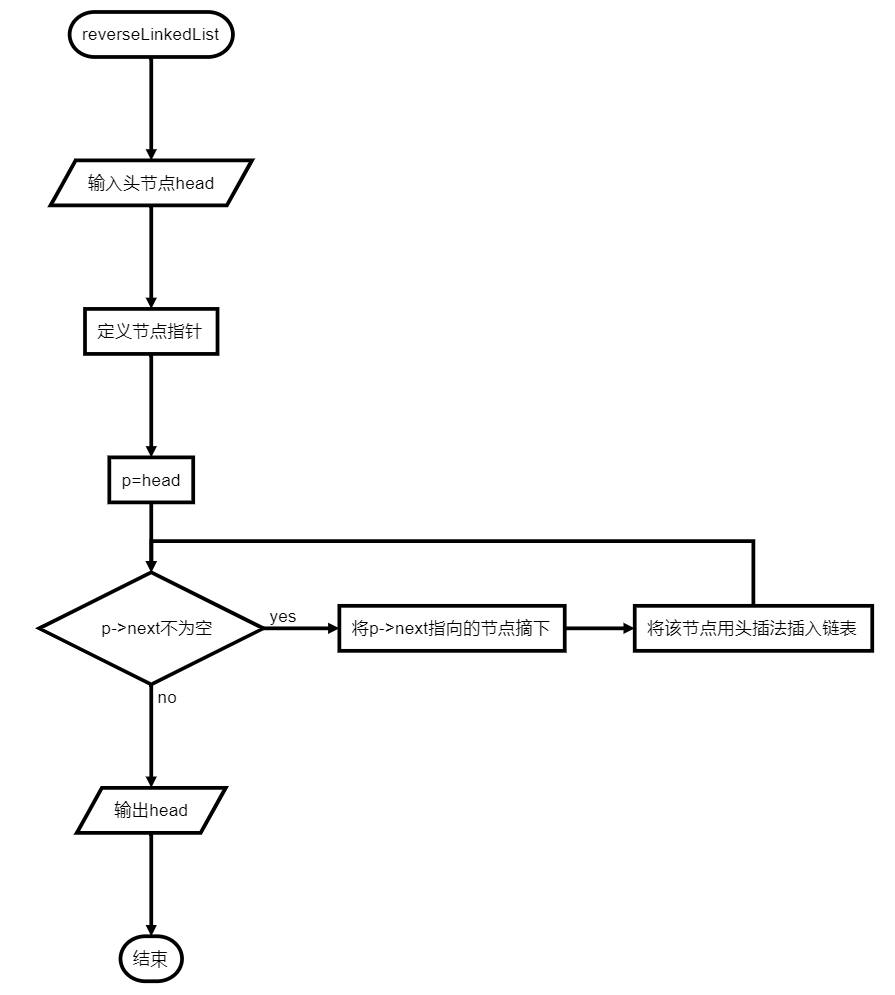
1. void printLinkedListNode(StudentLinkedListNode \* node);
2. void outputStudentLinkedList(StudentLinkedListNode\* head);
3. StudentLinkedListNode\* studentLinkedListCreate(char student\_id[], int grade);
4. StudentLinkedListNode\* studentLinkedListAdd(StudentLinkedListNode\* head, StudentLinkedListNode\* node);
5. StudentLinkedListNode\* reverseLinkedList(StudentLinkedListNode\*head);
6. StudentLinkedListNode\* findCrossBeginNode(StudentLinkedListNode\* class1, StudentLinkedListNode\* class2);

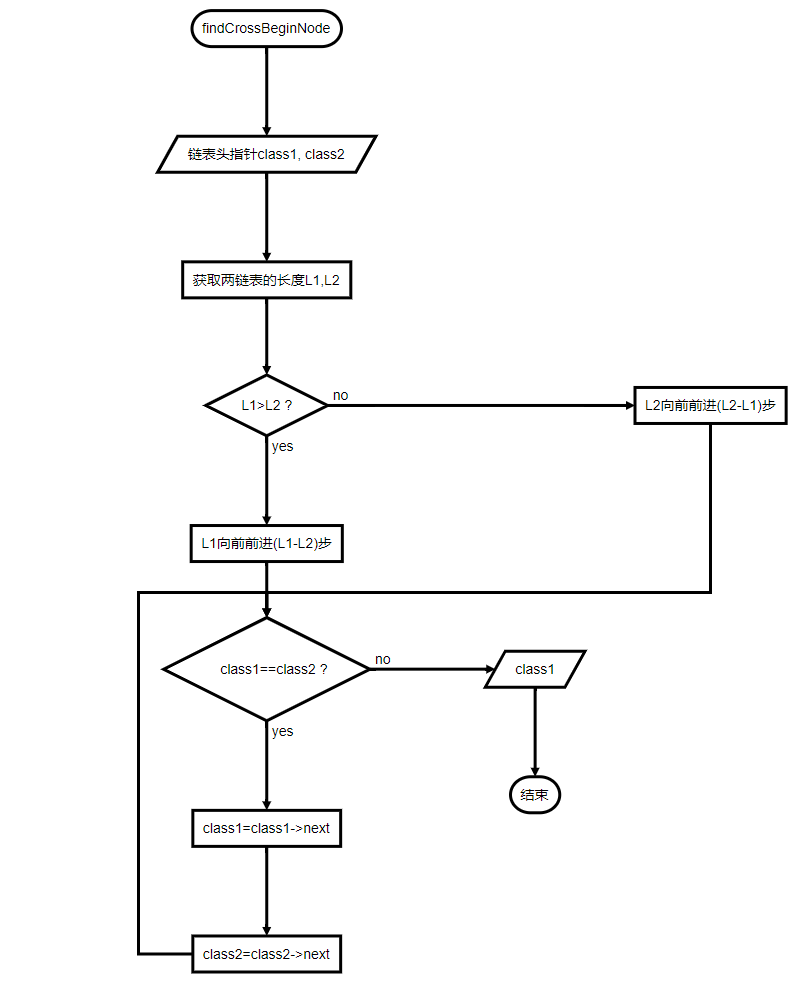
## 2.3 程序整体流程

画出整体流程，及核心算法流程。









# 三、用户手册

输入数据的方式：

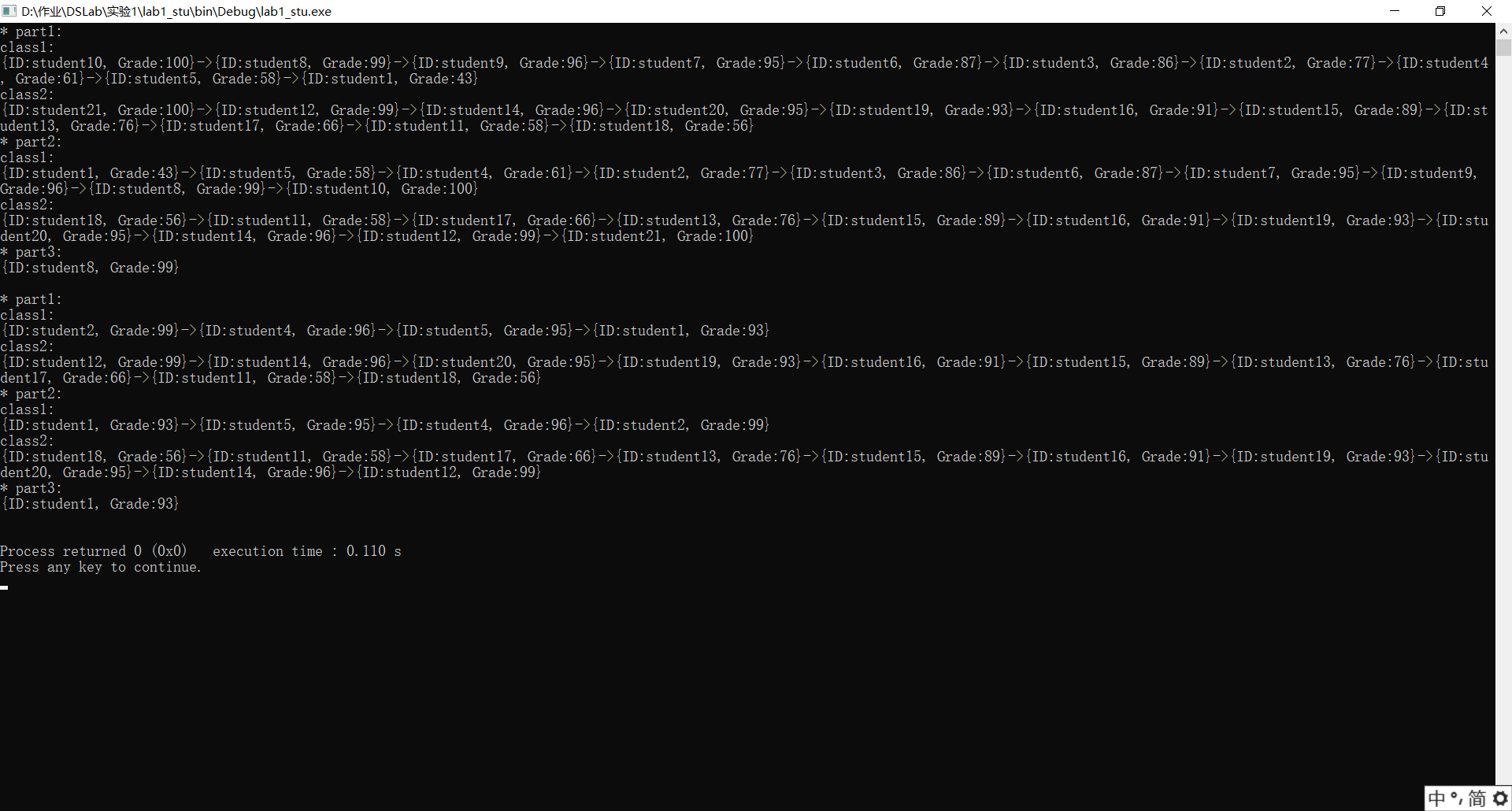
建立输入文件，相对路径为./gradeImport.in。

第一行的两个数字分别为两个班级的人数，用空格分隔。

接下去是每个同学的信息，格式为name score\n。

最后一行为链表的交叉位置，用空格分隔。

# 四、结果



# 五、总结

本实验涉及链表这一储存结构，涉及链表的遍历、删除节点、插入节点等基本操作。

在寻找链表交点时，我的程序在第二个测试数据时得到了错误的输出。经分析，原因是第二个测试数据涉及到了边界值，重新考虑了边界值，这个问题便得到了解决。这让我意识到测试边界值的重要性。

本次实验加深了我对链表的理解，锻炼了我对实际问题的分析以及程序的测试能力。