|  |
| --- |
| 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: | 宗晴 | | 学 号: | 200110513 | | 专 业: | 计算机科学与技术 | | 日 期: | 2021-04-01 | |

# 问题分析

实验原题是：学期结束，辅导员需要收集并整理两个班级同学的C语言课程成绩。请你为辅导员编写一个成绩录入统计程序，帮助辅导员更好地工作。同时，由于操作不慎，辅导员将将两个链表的后一部分交叉到了一起，要求编程找到交叉的第一个结点的位置。

转换成计算机要解决的问题就是，如何在计算机内按班级存储一组结构相同，均包含学生学号和分数信息的数据，并且学生人数未知（即要存储的数据规模未知）。该存储结构需要能够方便地进行成绩的管理与操作，由于涉及频繁的插入操作，且要求成绩按照降序存储，因此选择链式存储的线性表最为合适。每个班级以一个链表呈现，而班内每个学生的学号与分数信息则用一个节点呈现。要找到两个链表交叉的第一个节点，也就是要找到两个链表中第一个地址相同的节点。

# 二、详细设计

## 2.1 设计思想

用链表实现一个班级的成绩管理，其中每个节点代表一个学生的信息，包含该学生的学号和分数以及指向下一个学生的指针。

1、输出该表的成绩情况：用一个结构体指针访问链表中的每一个节点，用打印单个节点的函数输出当前节点的信息，并通过节点的next域访问下一个节点，直至指针指向表尾。

2、新建一个链表node并返回：用malloc申请一块节点内存，存放学号和分数信息，返回内存的地址。

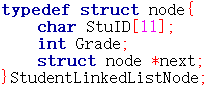
3、按照降序插入学生的成绩情况（输入总是升序的）：检查表是否为空，若为空表，则头指针直接指向该节点。否则，先判断该节点的分数是否大于第一个节点的分数，若成立，则将该节点的指针域指向第一个节点，再将头结点指向该节点，若不成立，则输出“输入未按升序排列”并退出。

4、反转链表：判断表是否为空，若为空，则输出相应的提示信息。判断表中是否只有一个节点，若是，则直接输出。否则用一个结构体指针指向第二个节点，第一个节点的指针域置为NULL，用另一个结构体指针指向当前节点的下一个节点，再将当前节点用头插法（上一点中已有介绍）插入第一个节点前，重复此过程直到插完最后一个节点。

5、找到相交的第一个结点：判断两表是否为空，若为空，则输出相应的提示信息。用两个指针分别遍历两个表，同时用len1, len2分别记录两个表的长度，循环结束时使两个指针分别指向两个表的最后一个节点，若两个节点的地址不同，则说明两个表不相交。否则，说明有相交节点，比较两表长度，将短表的指针指向该表的第一个节点，将长表指针移到与表尾相距短表表长的位置。两指针同时后移，判断当前两指针指向的节点的地址是否相同，找到地址相同的节点，将此节点返回。

## 2.2 存储结构及操作

(1) 存储结构（一般为自定义的数据类型，比如单链表，栈等。）



使用单链表的结构，每个节点代表一个学生的信息，包含该学生的学号和分数以及指向下一个学生的指针。

1. 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。）

* void outputStudentLinkedList (StudentLinkedListNode\* head)

参数：传入链表的头指针

功能：输出该表的成绩情况

返回值：无返回值

* StudentLinkedListNode\* studentLinkedListCreate (char student\_id[], int grade)

参数：传入用字符数组存放的学生学号，以及int型的学生分数

功能：新建一个链表的节点，包含传入的学号和分数信息

返回值：指向该新建节点的指针

* StudentLinkedListNode\* studentLinkedListAdd (StudentLinkedListNode\* head, StudentLinkedListNode\* node)

参数：传入链表的头指针，以及待插入的节点

功能：按照降序插入学生的成绩情况（由于输入的成绩总是升序的，所以只需使用头插法将新节点插入链表）

返回值：链表头指针

* StudentLinkedListNode\* reverseLinkedList(StudentLinkedListNode\*head)

参数：传入链表的头指针

功能：反转链表

返回值：链表头指针

* StudentLinkedListNode\* findCrossBeginNode(StudentLinkedListNode\* class1, StudentLinkedListNode\* class2)

参数：传入代表两个班级的两个链表的头指针

功能：找到两个链表相交的第一个结点

返回值：指向该相交节点的指针

## 2.3 程序整体流程



核心算法流程：

* void outputStudentLinkedList(StudentLinkedListNode\* head)函数：



* StudentLinkedListNode\* studentLinkedListCreate (char student\_id[], int grade)函数：



* StudentLinkedListNode\* studentLinkedListAdd (StudentLinkedListNode\* head, StudentLinkedListNode\* node)函数：



* StudentLinkedListNode\* reverseLinkedList(StudentLinkedListNode\*head)函数：



* StudentLinkedListNode\* findCrossBeginNode(StudentLinkedListNode\* class1, StudentLinkedListNode\* class2)函数：

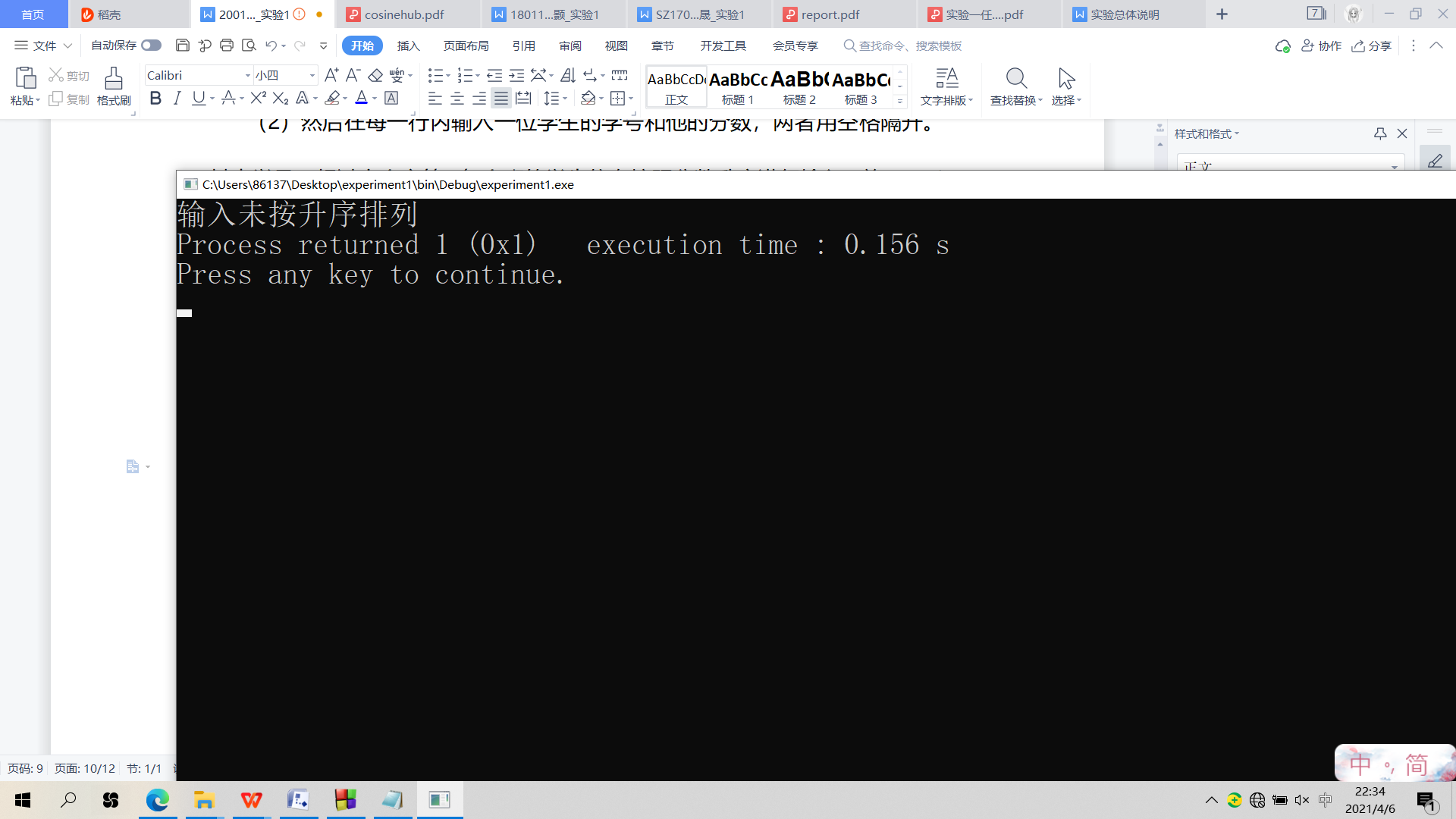
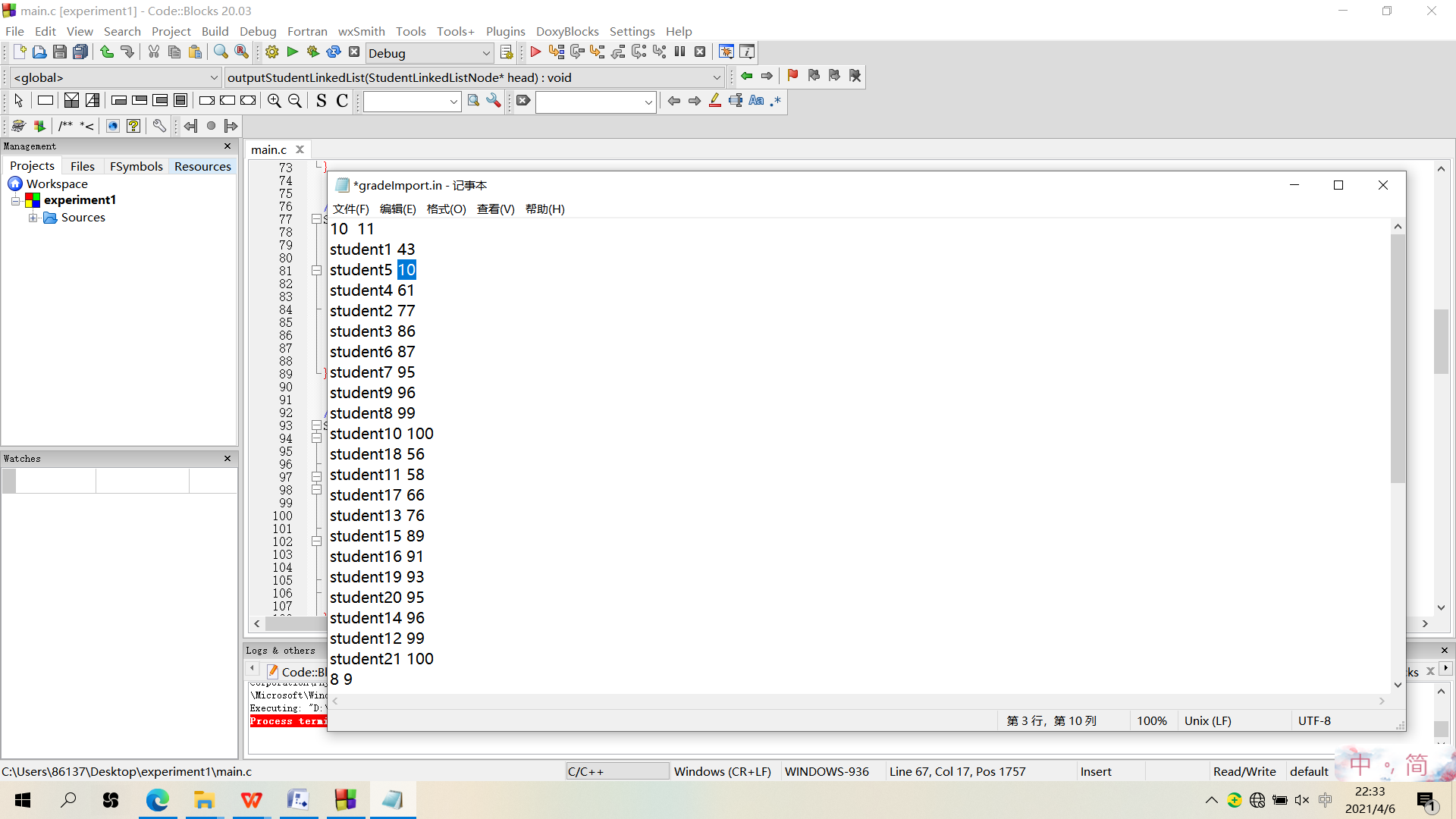


# 三、用户手册

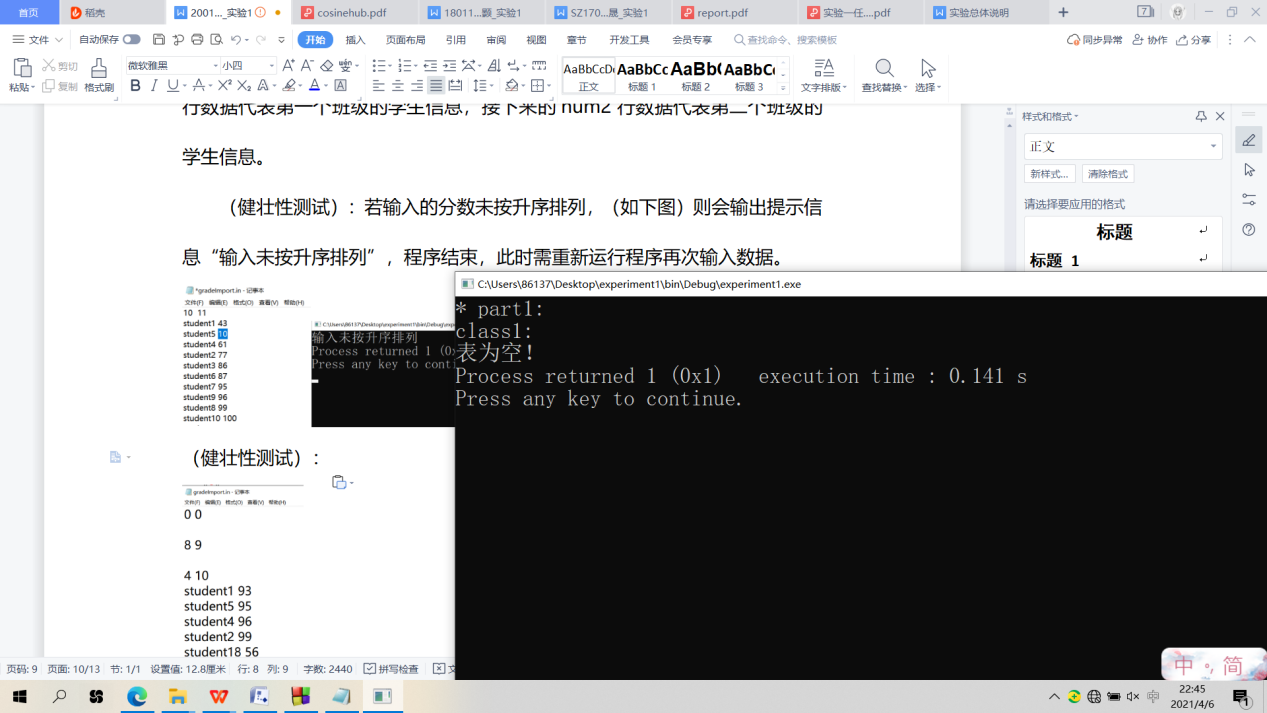
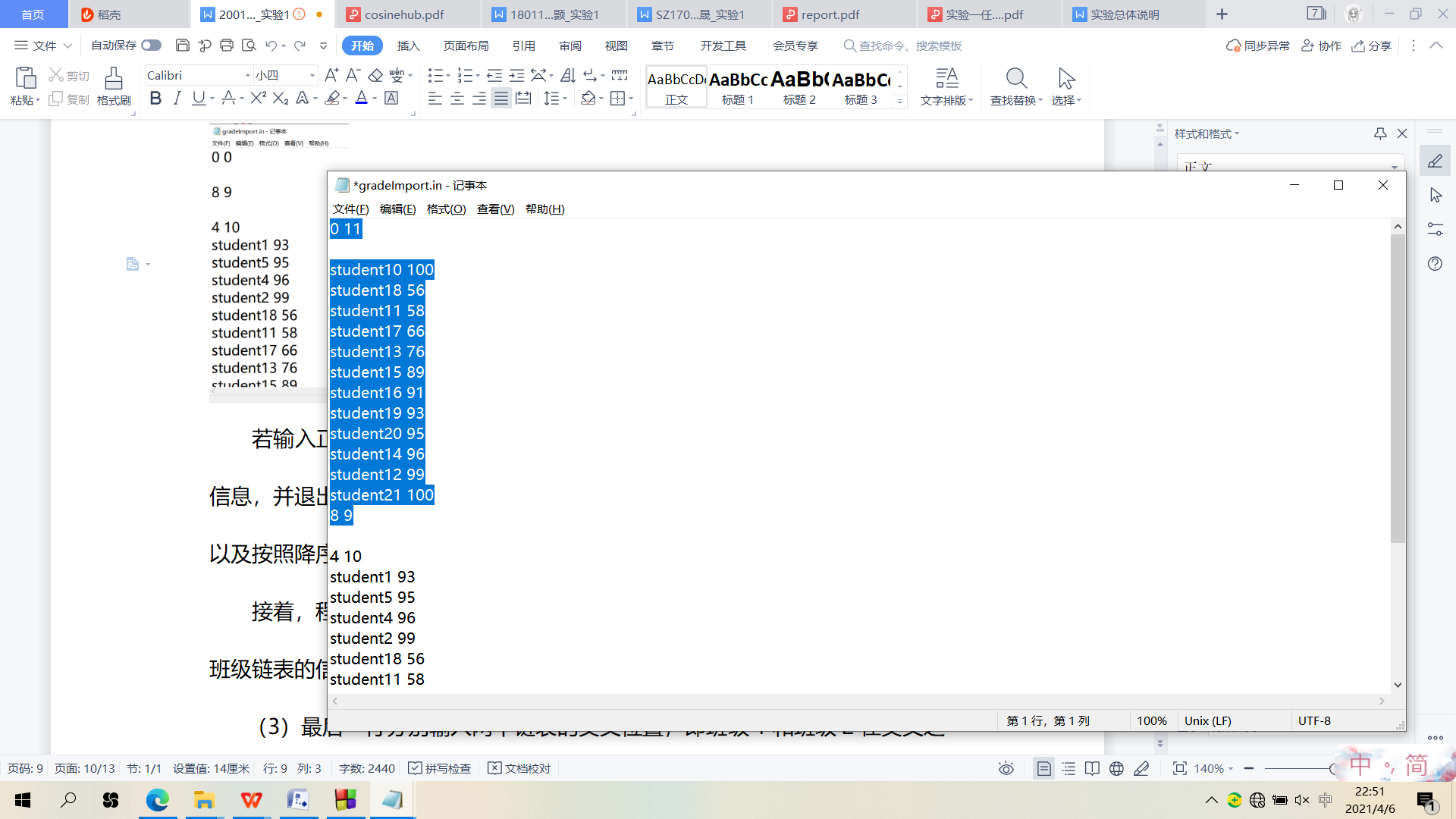
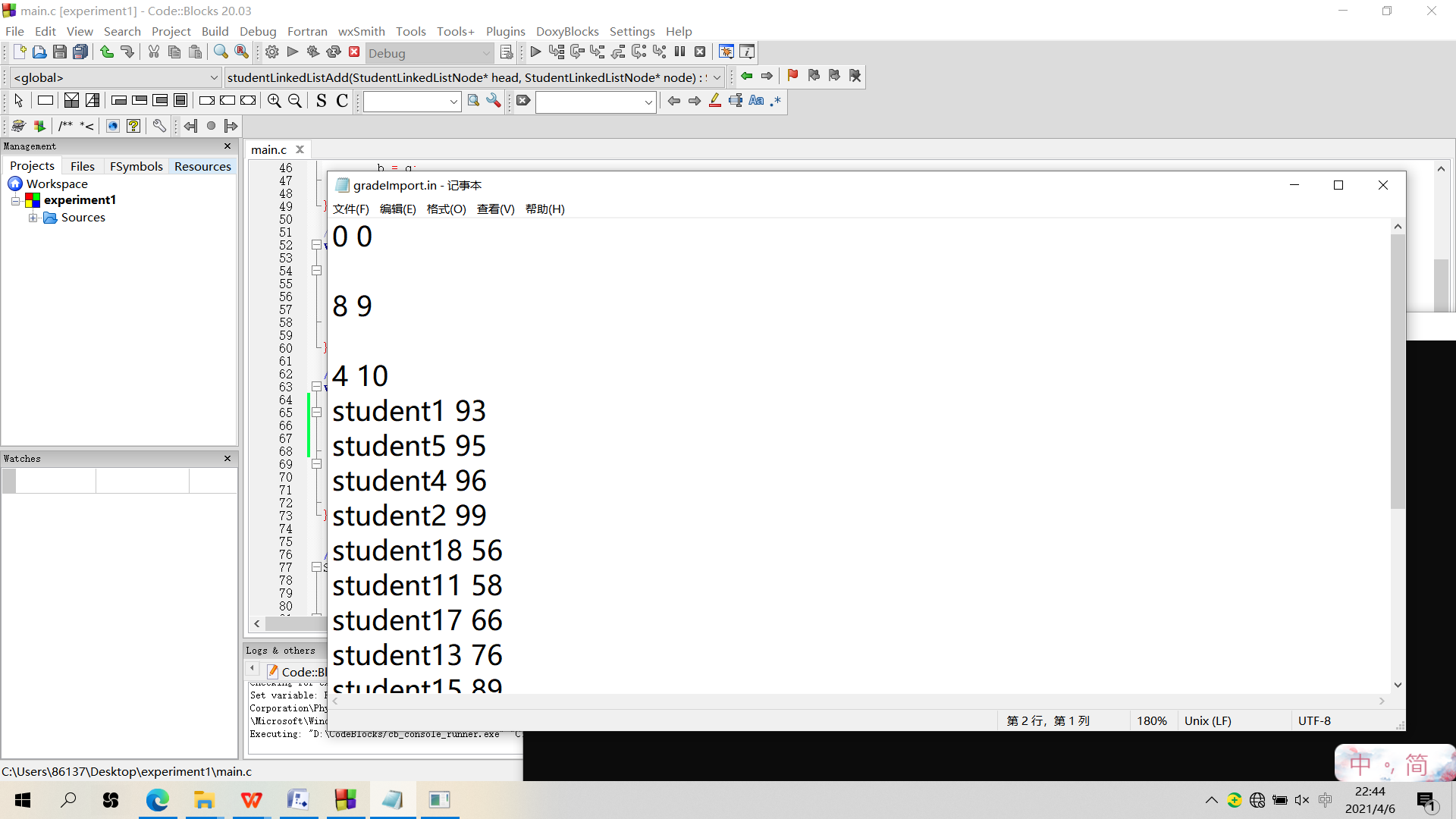
输入数据的方式以及实现各种功能的操作方式：

1. 用户首先应该输入分别代表两个班级人数的数据num1和num2，用空格隔开
2. 然后在每一行内输入一位学生的学号和他的分数，两者用空格隔开。其中学号不超过十个字符，每个班的学生信息按照分数升序进行输入。前num1行数据代表第一个班级的学生信息，接下来的num2行数据代表第二个班级的学生信息。

（健壮性测试）：若输入的分数未按升序排列，（如下图）则会输出提示信息“输入未按升序排列”，程序结束，此时需重新运行程序再次输入数据。



（健壮性测试）：若输入的班级数据为空，（如下图）则会输出提示信息“表为空！”，程序结束，此时需重新运行程序再次输入数据。



若输入正确，（健壮性：若申请内存失败，均会跳出“overflow”的提示信息，并退出程序），则此时两个班级链表已经建好，程序会输出“\* part1:”以及按照降序排列的两个班级链表的信息。

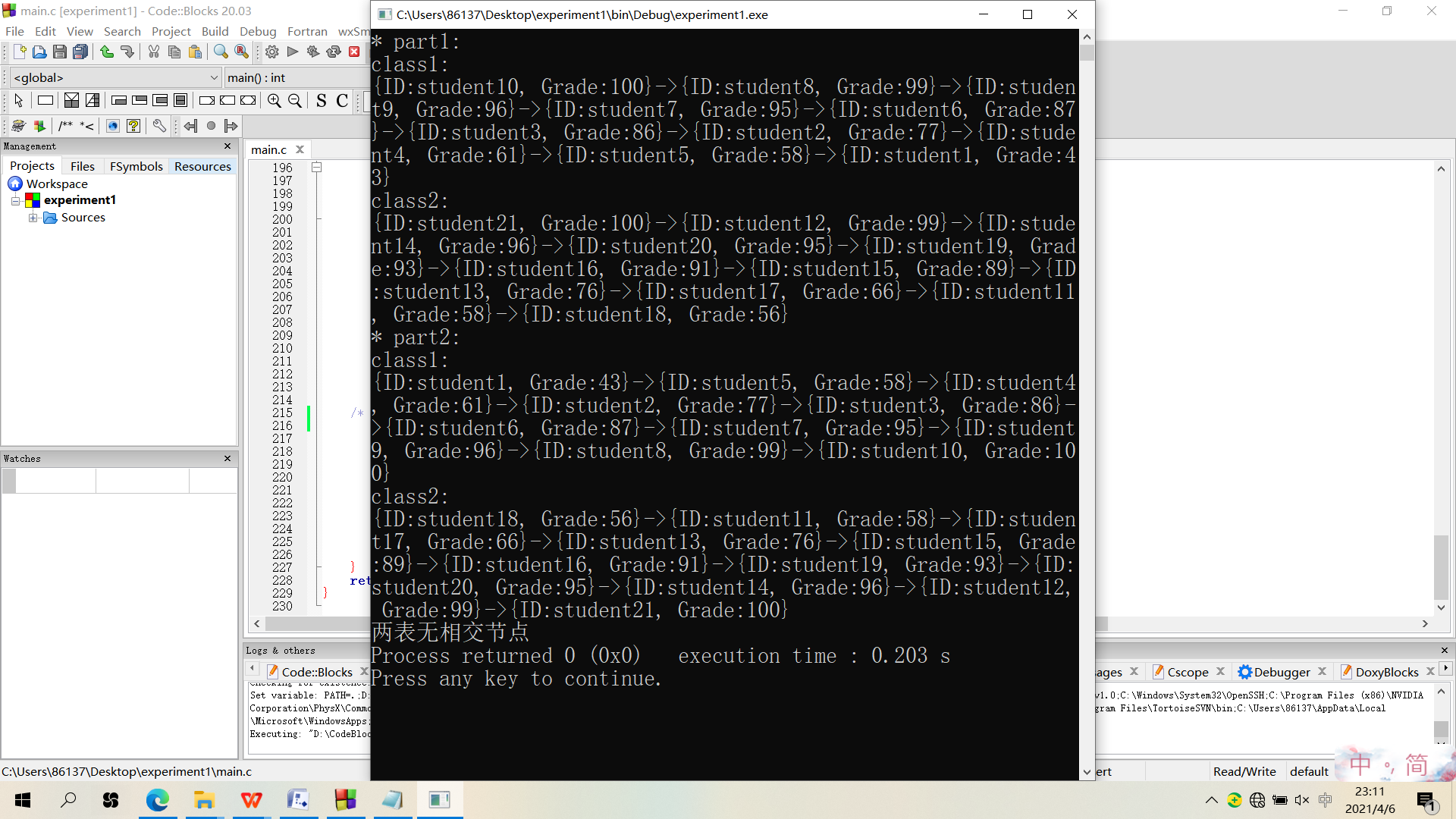
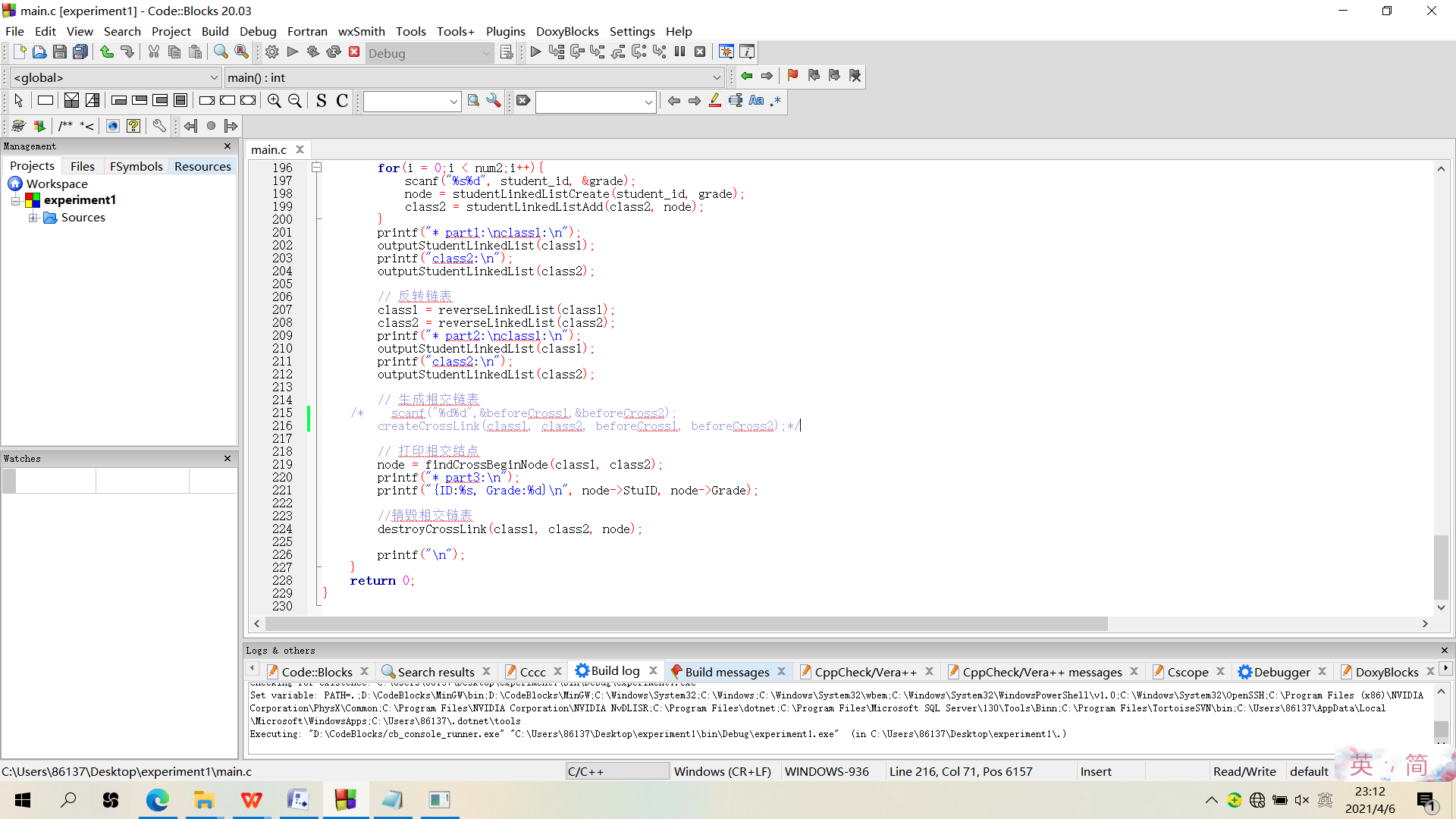
接着，程序会将两个链表反转（该函数内部也包含表是否为空的判断，但鉴于程序总是先执行输入部分的函数，当表为空时已经跳出程序，故该函数的健壮性在该程序中并无体现），然后输出“\* part2:”以及反转过后的两个班级链表的信息。

1. 最后一行分别输入两个链表的交叉位置，即班级1和班级2在交叉之前分别有几个结点，两者用空格隔开。

然后，程序会根据输入数据生成相交链表。

接着利用函数找到相交的节点

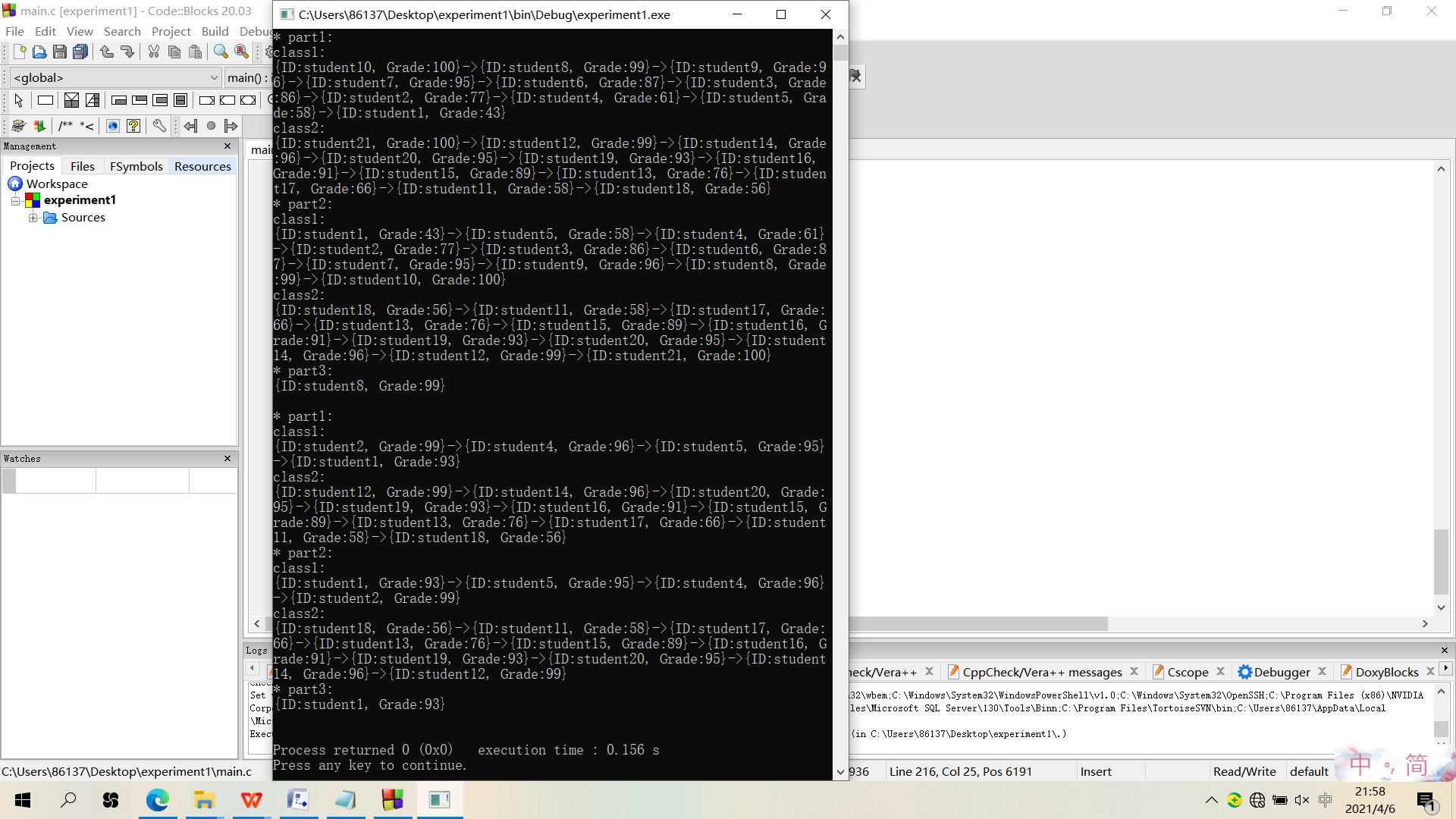
（健壮性测试）：该函数中增加了对于两表是否为空，是否有交叉节点的判断（如下图），若无交叉节点，则输出提示信息，程序结束，此时需重新运行程序再次输入数据。



若程序正常运行，则输出“\* part3:”以及相交的节点信息。

1. 若用户继续输入分别代表两个班级人数的数据num1和num2，用空格隔开，那么程序将会从上述第二点开始重复运行；否则程序结束。

# 结果



# 五、总结

该实验涉及到的数据结构主要是线性结构，题中是用单链表这种存储结构实现的。涉及的算法主要是创建链表、插入节点、遍历访问链表、反转链表、查找相交节点、销毁链表。

在此次试验中，我遇到了一些问题，最后都顺利解决了。

比如，不能直接在函数中改变头指针指向的地址，因为虽然参数列表里写的是指针，但实际上传进函数的是头指针指向的地址，在函数内改变这个地址是无效的，如果要想改变头指针指向的地址，只能通过函数返回值的地方返回一个新的地址给头指针。如果头指针后有一个头结点，将头指针传进函数，则是可以直接改变头结点指针域指向的内存地址的。

再比如，我深刻地认识到了链表有无头节点的操作的不同，当没有头结点时，需要增加一些边界情况的判断。

还有，一个循环的终止条件是当前指针为空指针，还是当前节点的指针域为空指针，真的是一件需要好好考虑的事情，这一点在实际实验时给我造成了不小的困扰。

通过这次实验，我对于链表有了更深刻的认识，熟悉了链表相关的基本操作，也学到了新的算法；同时，我也回顾了c语言的相关知识点，比如malloc函数、函数的按值调用和按引用调用、字符数组的拷贝等；也锻炼了调试的能力。