**东北大学计算机科学与工程学院**

**数据结构课程设计报告**

题目 排队购票问题

课题组长 于梓元

课题组成员 孟霞 张伟颖

专业名称 计算机科学与技术

班级 计1504

指导教师 张晓红

2017 年 1月

**课程设计任务书**

|  |
| --- |
| **问题描述：**  世界杯足球亚洲区预选赛正在激烈进行。决赛门票处于热卖。为使门票公平、安全的销售，售票处决定采用如下售票规则：  （1）购票者到购票处领取一个随机编号。购票者按随机编号从小到大排序。  （2）随机编号处于最小编号与最大编号之间的购票者，可直接到窗口排队购票。  （3）售票窗口空闲时随机发出0或1指令，指令为0时，最小编号者到窗口购票，指令为1时，最大编号者到窗口购票。 |
| **设计要求：**  设计算法实现按上述规则的排队售票程序。  （1）采用STL的双端队列类等数据结构。  （2）应用基本运算，设计算法求解 |
| **指导教师签字：**  **年　　月　　日** |

# 课题程序设计分工

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 姓名 | 程序设计函数原型、类 | 功能说明 |
| 20154561 | 于梓元 | show\_deque(intdeq &ideq)  menu\_back (intdeq &ideq, int &num,char command) | 显示当前队列元素  显示主菜单 |
| 20154379 | 张伟颖 | show\_command(intdeq &ideq )  delete\_dequenum(intdeq &ideq, int rand\_order1) | 生成随机指令  队列元素的删除 |
| 20154508 | 孟霞 | show\_num(intdeq &ideq )  deque\_num(intdeq &ideq, int num) | 生成随机编号  将新生成的编号插入到有序的队列里 |

# 课题报告分工

|  |  |  |
| --- | --- | --- |
| 章节 | 内容 | 完成人 |
| 1 课题概述 | 1.1 课题任务  1.2 课题原理  1.3 相关知识 | 张伟颖 |
| 2 需求分析 | 2.1 课题调研  2.2 用户需求分析 | 于梓元 |
| 3 方案设计 | 3.1 总体功能设计  3.2 数据结构设计  3.3 函数原型设计  3.4 主算法设计  3.5 用户界面设计 | 孟霞 |
| 4 方案实现 | 4.1 开发环境与工具  4.2 程序设计关键技术  4.3 个人设计实现（按组员分工）  4.3.1  4.3.2  4.3.3 | 于梓元  张伟颖  孟霞 |
| 5 测试与调试 | 5.1 个人测试（按组员分工）  5.1.1  5.1.2  5.1.3  5.2 组装与系统测试  5.3 系统运行 | 孟霞 |
| 6 课题总结 | 6.1 课题评价  6.2 团队协作  6.3 下一步工作  6.4 个人设计心得（按组员分工）  6.4.1  6.4.2  6.4.3 | 于梓元  孟霞  张伟颖 |

目录

[课题程序设计分工 - 2 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248370)

[课题报告分工 - 2 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248371)

[1 课题概述 - 4--](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248372)

[1.1 课题任务 - 4 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248373)

[1.2 课题原理 - 4 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248374)

[1.3 相关知识 - 5 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248375)

[2 需求分析 - 5 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248376)

[2.1 课题调研 - 5 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248377)

[2.2 用户需求分析 - 5 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248378)

[3 方案设计 - 5 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248379)

[3.1 总体功能设计 - 5 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248380)

[3.2 数据结构设计 - 5 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248381)

[3.3 函数原型设计 - 5 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248382)

[3.4 主算法设计 - 6 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248383)

[3.5 用户界面设计 -6 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248384)

[4 方案实现 - 7 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248386)

[4.1 开发环境与工具 - 7 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248387)

[4.2 程序设计关键技术 - 7 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248388)

[4.3 个人设计实现（按组员分工） - 8 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248389)

[4.3.1 于梓元设计实现 - 7 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248390)

[4.3.2 张伟颖设计实现 - 8 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248391)

4.3.3 孟霞设计实现----------------------------------------------------------------------------------------------------------------- 10-

[5 测试与调试--------------------------------------------------------------------------------------------------------------------- ------ 10 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248392)

[5.1 个人测试（按组员分工） - 11 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248393)

[5.1.1于梓元测试 - 11 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248394)

[5.1.2张伟颖测试 - 12 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248395)

5.1.3孟霞测试-------------------------------------------------------------------------------------------------------------------------12-

[5.2 组装与系统测试 - 14 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248396)

[5.3 系统运行 - 15 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248397)

[6 课题总结 - 18-](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248398)

[6.1 课题评价 - 18 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248399)

[6.2 团队协作 - 18 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248400)

[6.3 个人设计小结（按组员分工） - 18 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248401)

[6.3.1 于梓元的小结 - 18 -](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248402)

[6.3.2 张伟颖的小结 - 19-](file:///C:\Users\45714\Desktop\课程设计实验报告\排队购票问题.doc#_Toc441248403)

6.3.3 孟霞的小结---------------------------------------------------------------------------------------------------------------------19

7 附录......................................................--19-

# 1 课题概述

1.1课题任务

世界杯足球亚洲区预选赛正在激烈进行。决赛门票处于热卖。为使门票公平、安全的销售，售票处决定采用如下售票规则：

（1）购票者到购票处领取一个随机编号。购票者按随机编号从小到大排序。

（2）随机编号处于最小编号与最大编号之间的购票者，可直接到窗口排队购票。

（3）售票窗口空闲时随机发出0或1指令，指令为0时，最小编号者到窗口购票，指令为1时，最大编号者到窗口购票。

【设计要求】

设计算法实现按上述规则的排队售票程序。

（1）采用STL的双端队列类等数据结构。

（2）应用基本运算，设计算法求解。

1.2 课题原理

本课题采用了双端队列来存储生成的随机编号，并且将生成的随机编号插入队列，为方便后续操作，在插入的时候，保持了队列元素的有序性。首先利用随机函数产生随机编号，将其按生成顺序存储到某数组中，按递增顺序存储到双端队列中。而后利用随机函数产生随机指令0或1，指令为0时，显示当前队列最小编号者到窗口购票，将该编号在数组和双端队列中删除，指令为1时，显示当前队列最大编号者到窗口购票，将该编号在数组和双端队列中删除。显示正在排队的编号，即是把队列中的元素显示出来。

1.3 相关知识

1.随机数的生成。通过调用随机函数生成随机编号赋给购票者以及随机指令的发出。

2.STL双端队列的运用。利用STL的双端队列的初始化、插入、删除等基本数据结构实现随机编号的出入队列。

# 2 需求分析

2.1 课题调研

1）此例类似于银行排号问题。

2）能实现为购票者随机产生编号（假定为1-100）

3）将分配好的编号插入队列，编号在队列中从小到大排列，可以随时查看队列中的编号

4）随机发出0或1指令，以确保购票次序的相对公平

5）利用双端队列，实现队首队尾都可以进行插入和删除

2.2 用户需求分析

1）随机分配给购票者编号

2）能够从已分配的编号中选出最大编号和最小编号

3）将已经购完票的编号从双端队列里删去

4）系统随机给出1或0的指令

5）如果某个买票者未出现，可以刷新指令

# 3 方案设计

3.1 总体功能设计

本实验主要实现三个功能: 为购票者随机分配编号、售票窗口随机发出指令、随时查看队列中的编号，同时还隐藏着将最大编号或最小编号在队列中删除、将新生成的编号插入到编号有序的队列中。

为购票者随机分配编号，售票窗口随机生成指令，都需要用都随机函数

将最大编号或最小编号在队列中删除，涉及到双端队列的相关基本操作

将新生成的编号插入到编号有序的队列中，需要利用迭代器，将新生成的编号与队列中的编号一一比较。

3.2 数据结构设计

本程序使用了STL双端队列类里的deque，主要涉及的操作有：insert()，begin(),end()

pop\_front(),pop\_back(),push\_front(),push\_back()同时还调用了随机生成函数。

3.3 函数原型设计

show\_num(intdeq &ideq ) 领取随机编号

deque\_num(intdeq &ideq, int num) 将产生的随机编号插入到双端队列中

show\_deque(intdeq &ideq) 显示当前队列中的元素

delete\_dequenum(intdeq &ideq,int rand\_order1) 双端队列元素的移除，即购票之后相应编号出队列

show\_command(intdeq &ideq ) 生成并输出售票窗口的指令

menu\_back(intdeq &ideq,int &num,char command)主菜单

3.4 主算法设计

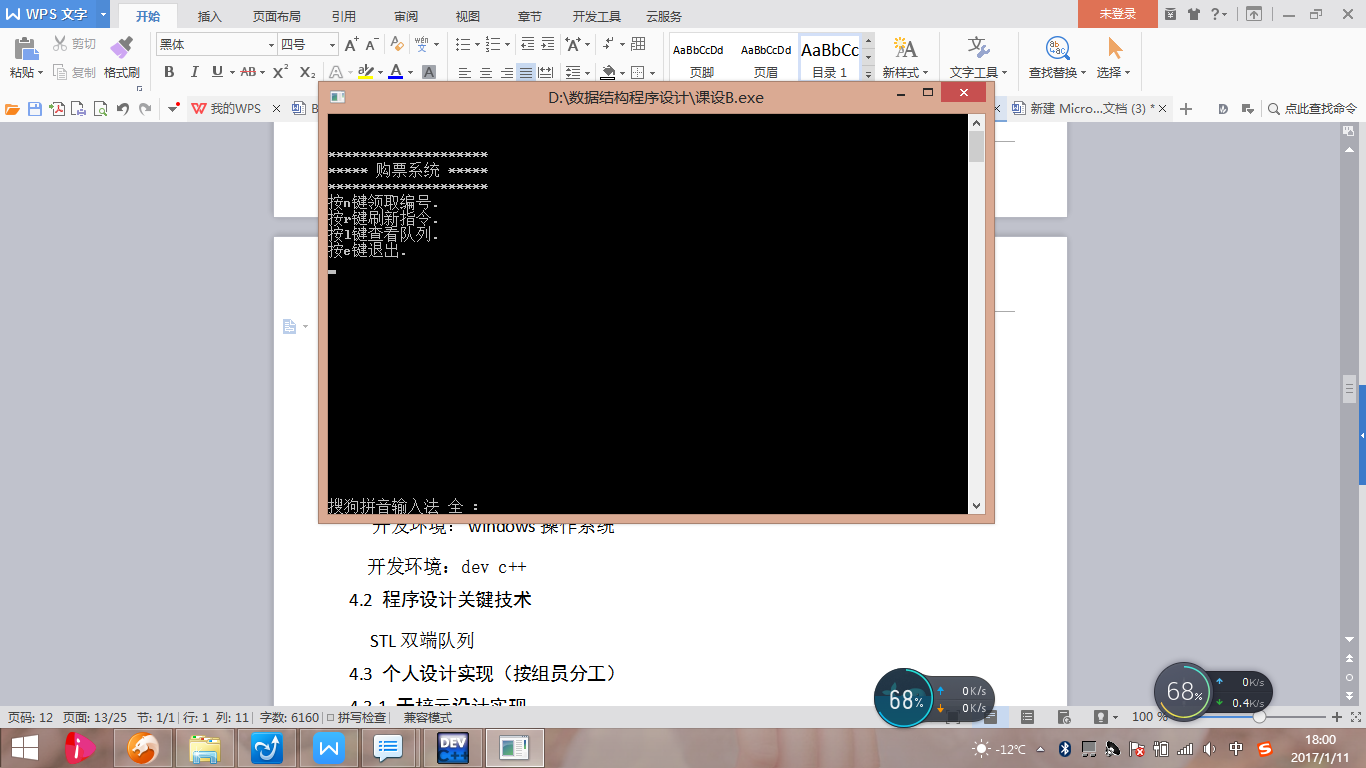
随机数的生成和随机指令的生成：使用srand( )提供种子数，再使用rand( )函数生成随机函数

队列的操作：查找、插入、删除

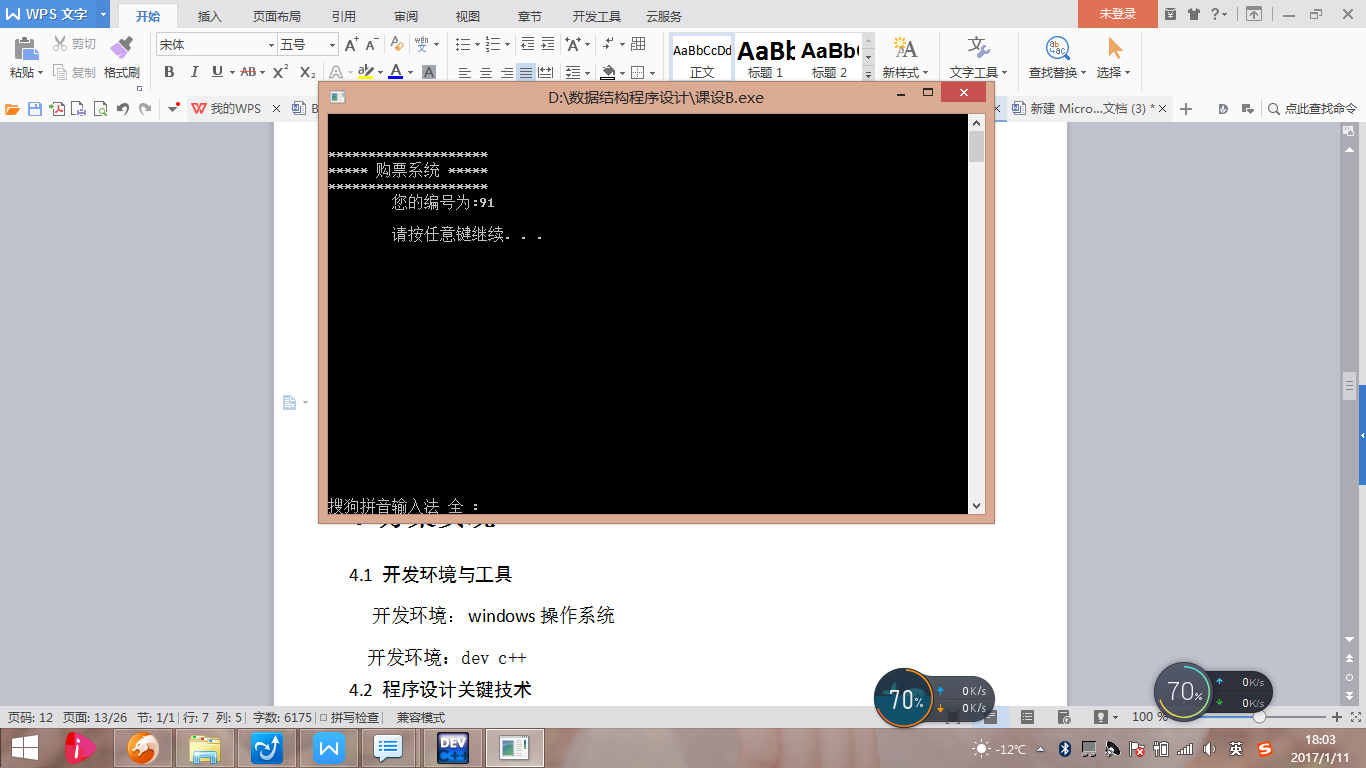
界面：显示当前队列、随机编号

3.5 用户界面设计

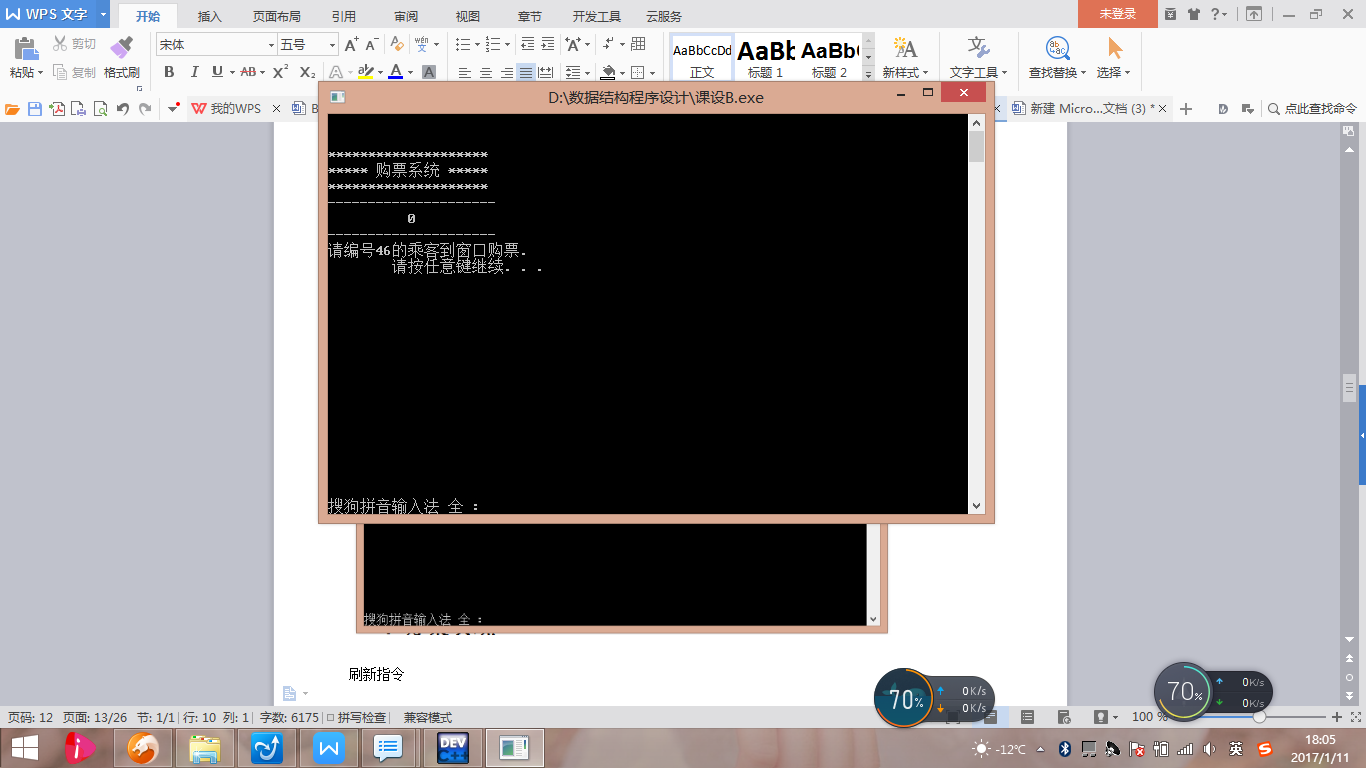
主界面



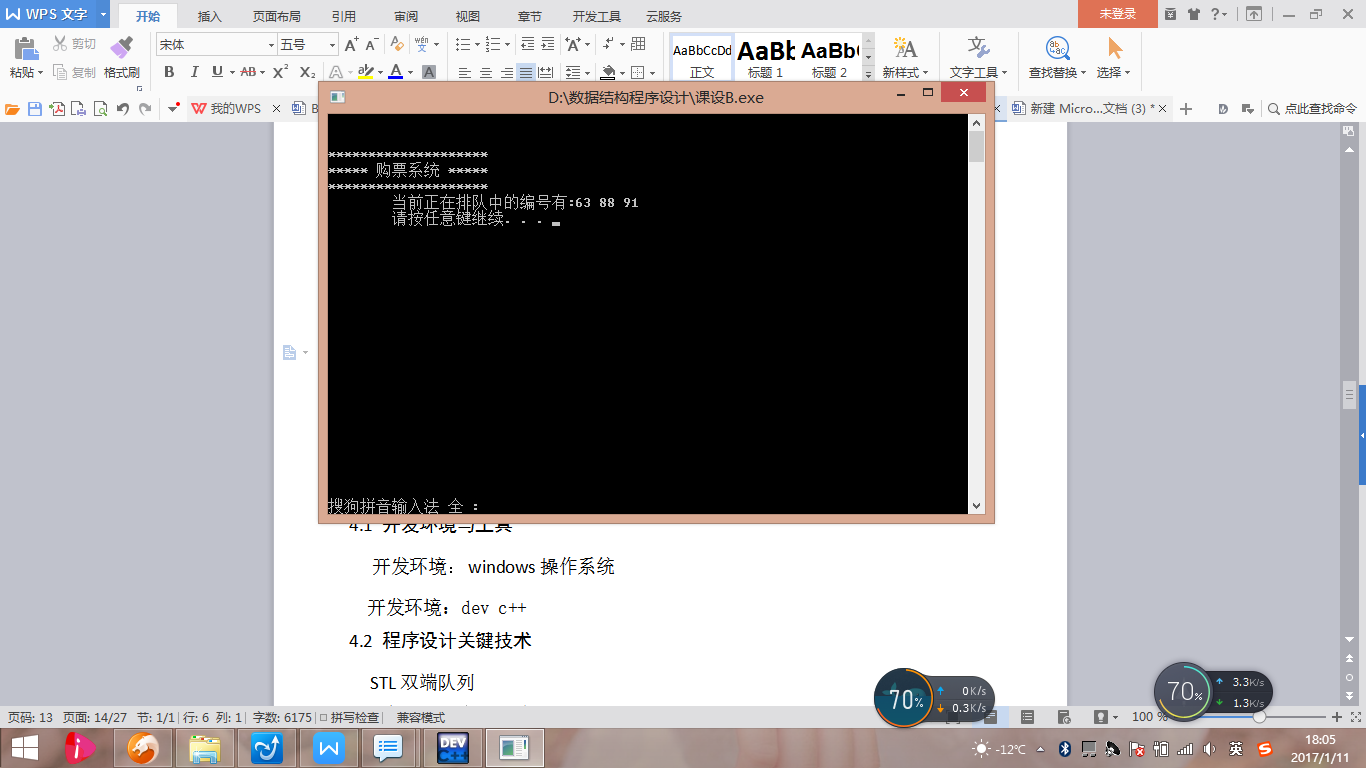
领取编号



刷新指令



查看队列



# 4 方案实现

4.1 开发环境与工具

开发环境： windows操作系统

开发工具： codeblocks

4.2 程序设计关键技术

STL双端队列

4.3 个人设计实现（按组员分工）

4.3.1 于梓元设计实现

主要负责显示当前队列和主菜单的设计

void show\_deque(intdeq &ideq)

{

system("cls");

cout << endl << endl;

cout <<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" <<endl;

cout <<"\*\*\*\*\* 购票系统 \*\*\*\*\*" <<endl;

cout <<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" <<endl;

if(ideq.empty())

cout <<"\t现在没有人排队。" << endl;

else

{

cout <<"\t当前正在排队中的编号有:";

copy(ideq.begin(),ideq.end(),screen);//输出队列ideq中的元素，从begin到end

//STL元素复制算法copy。该算法主要用于容器之间元素的拷贝

}

cout << endl << "\t";

system("pause");// system("pause")就是从程序里调用“pause”命令； 而“pause”这个系统命令的功能很简单，就是在命令行上输出一行类似于“Press any key to exit”的字，等待用户按一个键，然后返回。

}

//菜单信息

void menu\_back (intdeq &ideq, int &num,char command)

{

if( command == 'n' || command == 'N' )

{

num=show\_num(ideq);//领取随机编号

deque\_num(ideq, num );//对随机编号排序

}

else if (command == 'l' || command == 'L')

show\_deque(ideq);//显示当前队列

else if (command == 'r' || command == 'R')

show\_command (ideq) ;//输出售票窗口的指令

}

char menu()

{

char command;

system ("cls" );

cout<<endl<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" <<endl;

cout<<"\*\*\*\*\* 购票系统 \*\*\*\*\*" <<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" <<endl;

cout<<"按n键领取编号."<<endl;

cout<<"按r键刷新指令."<<endl;

cout<<"按l键查看队列."<<endl;

cout<<"按e键退出."<<endl;

cin>>command;

return command;

}

4.3.2 张伟颖设计实现

主要负责随机指令的生成和队列元素的删除

void show\_command(intdeq &ideq )

{

rand\_order = rand() % 2;//生成[0,2)之间的随机整数，也即随机生成0或1

system("cls");//清除上一个回车所显示的内容

cout<<endl<<endl;

cout <<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" <<endl;

cout <<"\*\*\*\*\* 购票系统 \*\*\*\*\*" <<endl;

cout <<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" <<endl;

cout<<"---------------------" <<endl;

cout<<" "<<rand\_order<<" " <<endl;

cout<<"---------------------" <<endl;

if (!ideq.empty())

{

if(rand\_order == 0)

{

cout<<"请编号"<<ideq.front()<<"的乘客到窗口购票."<<endl;//指令为0，最小编号购票

delete\_dequenum (ideq, rand\_order);//把最小编号在队列中删除

}

else if(rand\_order == 1)

{

cout<<"请编号"<<ideq.back()<<"的乘客到窗口购票."<<endl;//指令为1，最大编号购票

delete\_dequenum (ideq, rand\_order); //把最大编号在队列里删除

}

}

else

cout<<"当前窗口没有人排队."<<endl;

cout<<"\t";

system ("pause");//暂停，便于观察结果。按任意键继续

}

void delete\_dequenum(intdeq &ideq, int rand\_order1)

{

bool jud = 0;//jud表示之前的数是否是上界的前缀（即后面的数能否任意填）

int i=0;

if(rand\_order1 == 0)

{

for(i;i<sum;i++)

{

if(num[i] == ideq.front())//deq.front()返回第一个元素

jud = 1;

if(jud == 1)//通过循环把数组num[]中的队列首号码删除

num[i] = num[i+1];

}

if(jud == 1)

num[sum] = 0;

ideq.pop\_front();//deq.pop\_front()删除第一个元素

}

else if(rand\_order1 == 1)

{

for(i;i<sum;i++)

{

if(num[i] == ideq.back())//deq.back 返回最后一个元素

jud = 1;

if(jud = 1)//通过循环把数组num[]中的队列尾号码删除

num[i] = num[i+1];

}

if(jud == 1)

num[sum] = 0;

ideq.pop\_back();//删除最后一个元素

}

sum--;//总票数减一

}

4.3.3 孟霞设计实现

主要负责随机编号的生成，和将生成的编号插入到数组和双端队列中

//领取随机编号

int show\_num(intdeq &ideq )

{

system("cls" );//清屏

srand((unsigned )time(NULL));//指定种子数seed为当前系统流逝了的时间（单位为秒）：time\_t time(0)

//目的是使每次运行结果也即生成的随机数将不一样，因为每次启动程序的时刻都不同

int rand\_num = rand()% 100 + 1;//生成[1,100)的随机数

num[sum] = rand\_num;

int j = 0;

for(j;j<sum;j++)

{

if(rand\_num == num[sum])

rand\_num=rand()% 100 + 1;//如果生成的随机数在之前已经生成过，则重新生成一个随机数

}

sum++;

cout<<endl<<endl;

cout <<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" <<endl;

cout <<"\*\*\*\*\* 购票系统 \*\*\*\*\*" <<endl;

cout <<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" <<endl;

cout<<"\t您的编号为:"<<rand\_num<<endl;

cout<<endl<<"\t";

system("pause" );

return rand\_num;

}

//对随机编号排序

void deque\_num(intdeq &ideq, int num)

{

int i = 0; //cout "test" endl;

if(num < ideq.front())

ideq.push\_front(num);//num比队首元素还小，则插入到队列首位

else if(num > ideq.back())

ideq.push\_back(num);//num比队尾元素还大，则插入到队列尾

else

{

deque<int>::iterator deqIt;//deqIt是用deque<int>声明的容器，对于存储空间连续的数据结构，可以通过累加来遍历

//迭代器其实就是指针，读取集合或者数组中的一个值，读完以后又指向下一条数据。

deqIt = ideq.begin();//以上两语句进行指针初始化

while(num > ideq[i])

{

deqIt++;

i++;

}//到数据在队列中的插入位置

ideq.insert(deqIt, num);//插入

}

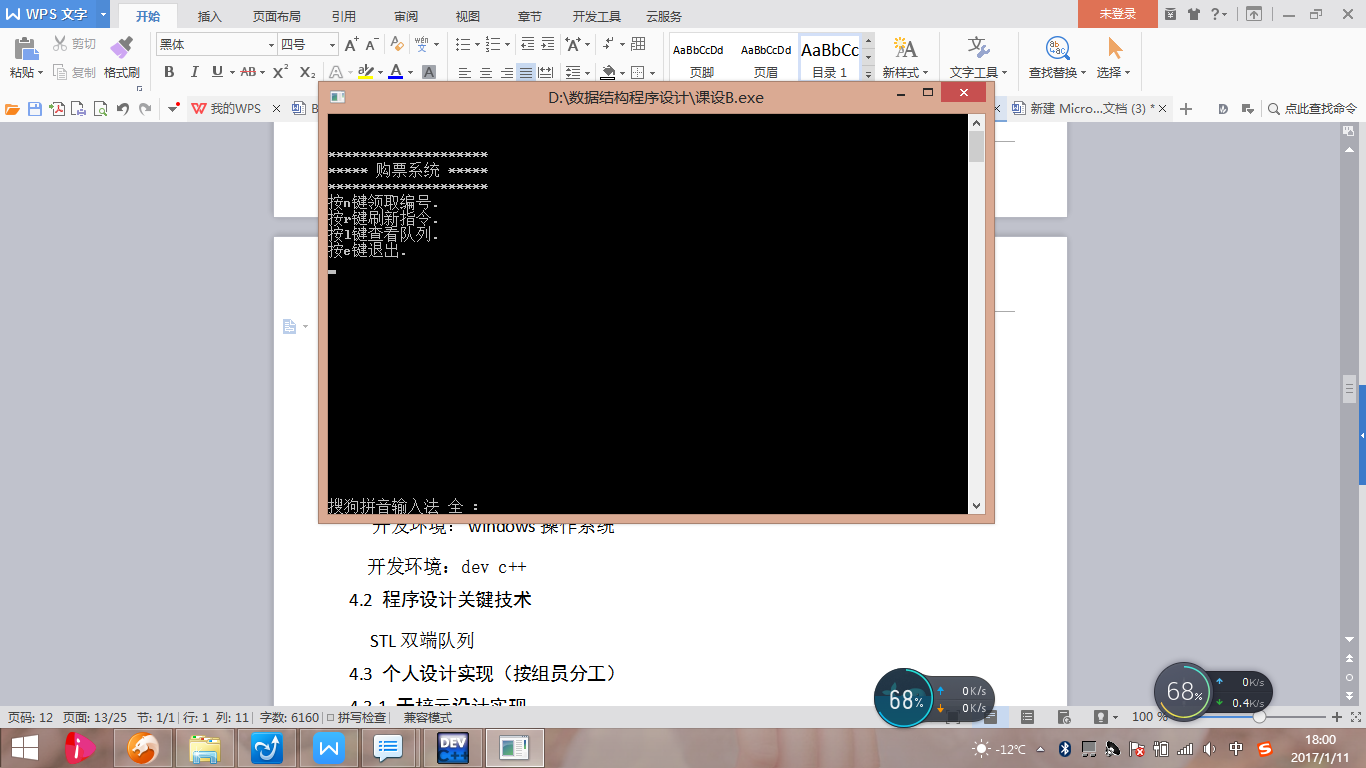
}

# 5 测试与调试

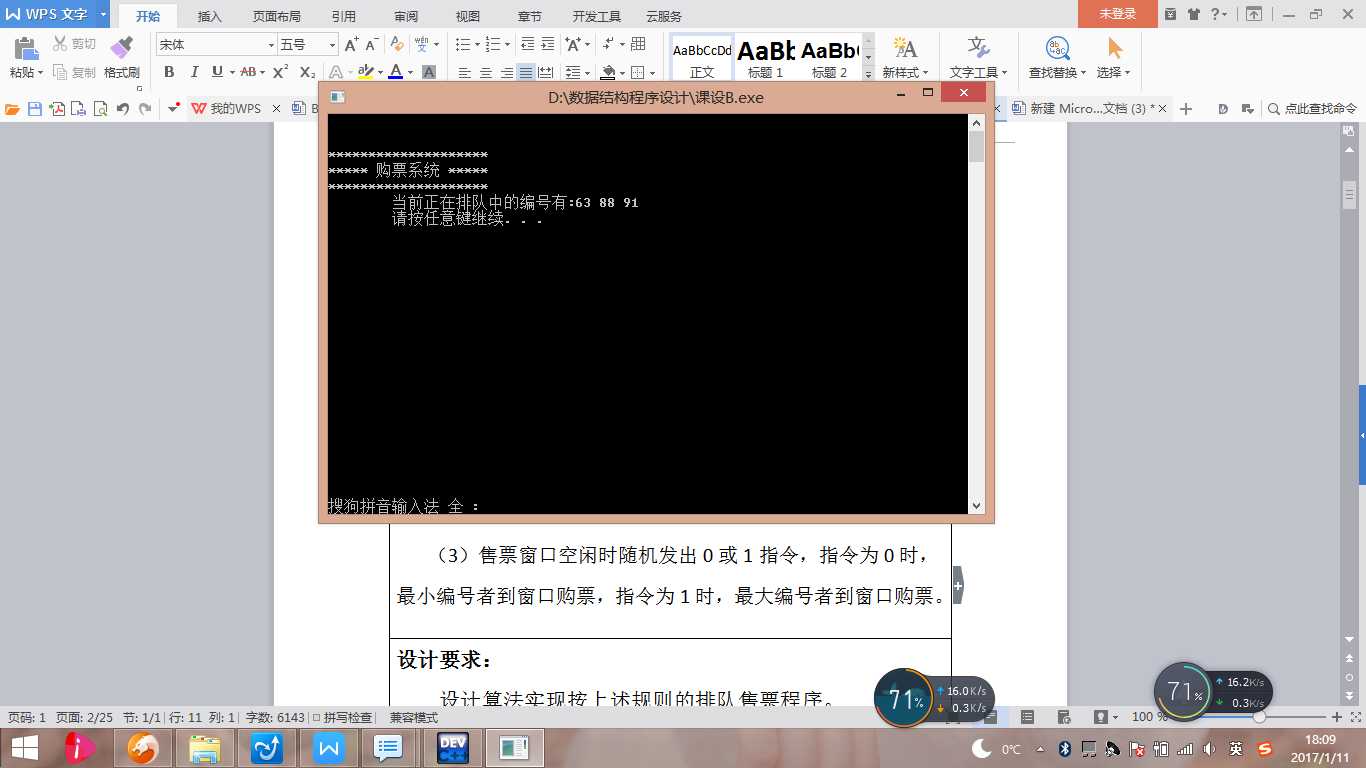
5.1 个人测试（按组员分工）

5.1.1 于梓元测试

主界面设计

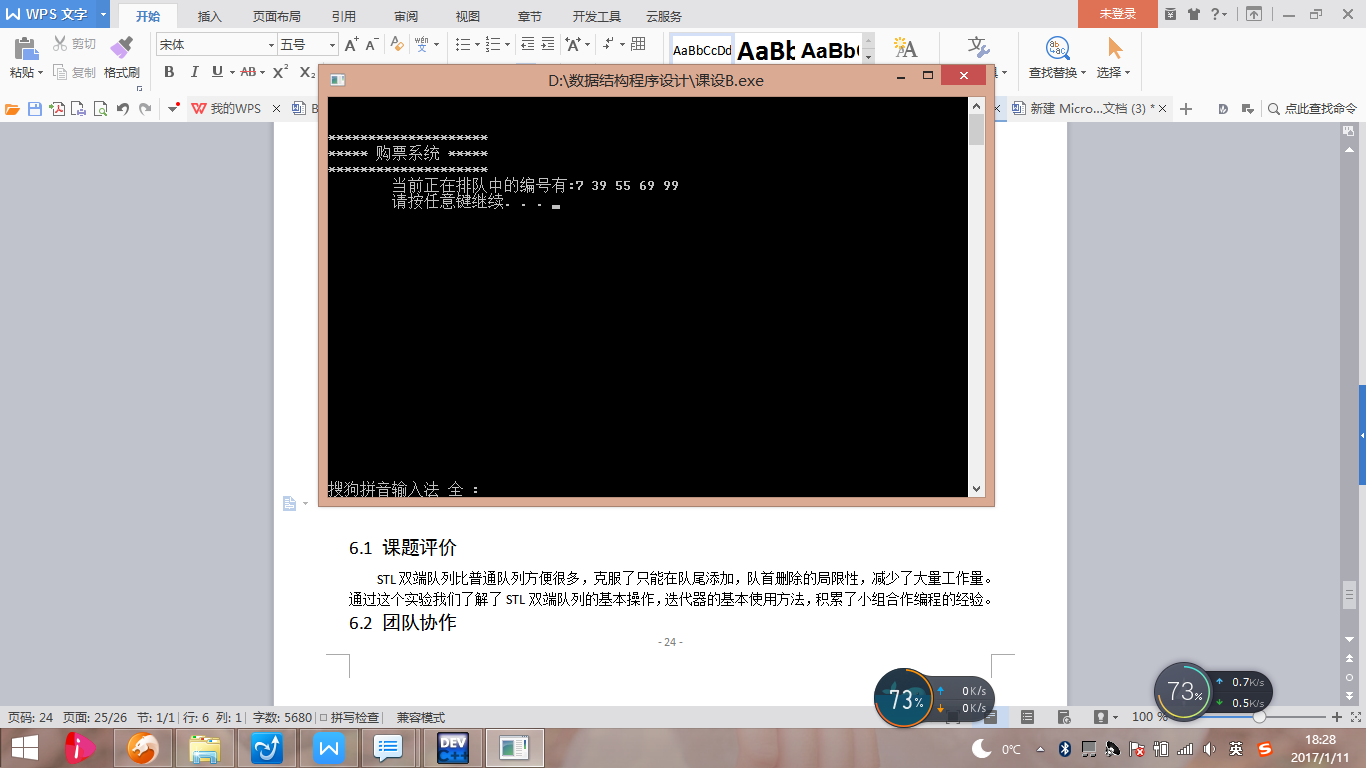


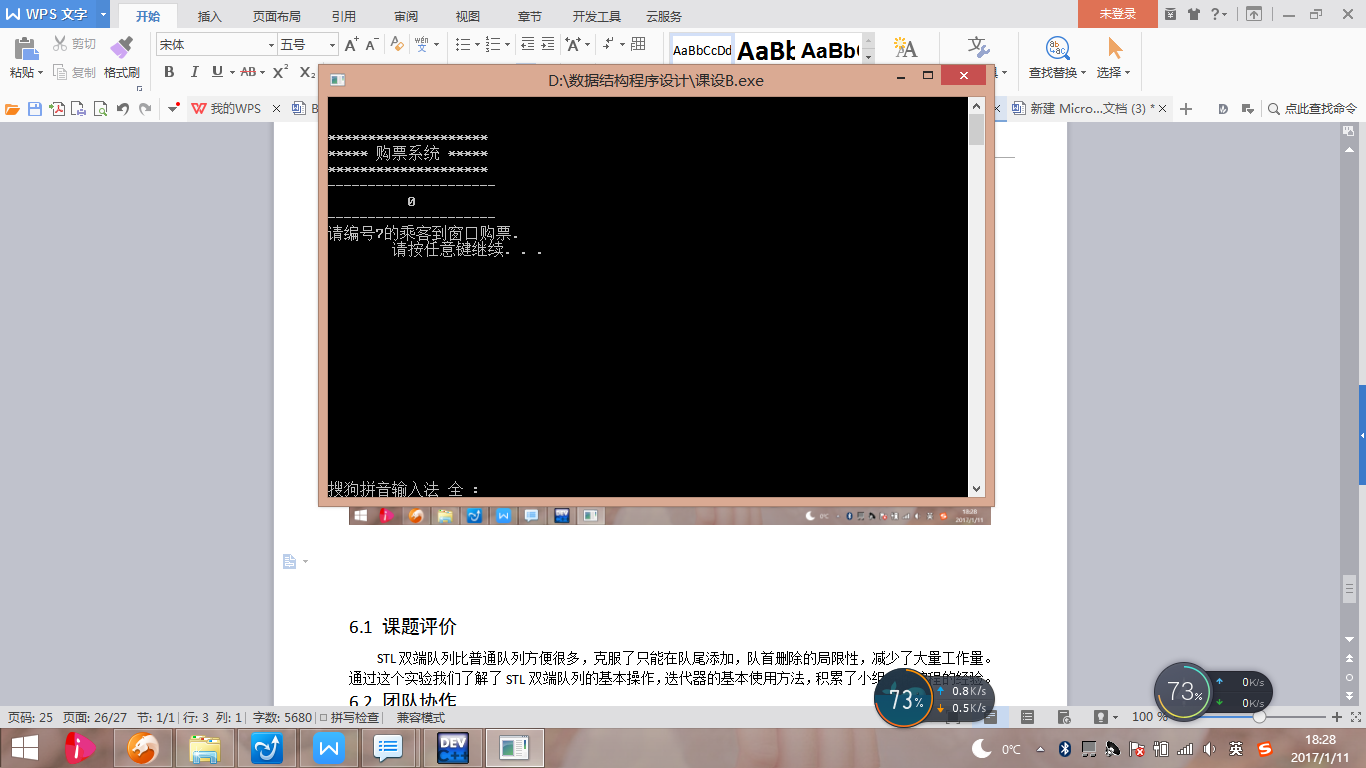
显示当前队列

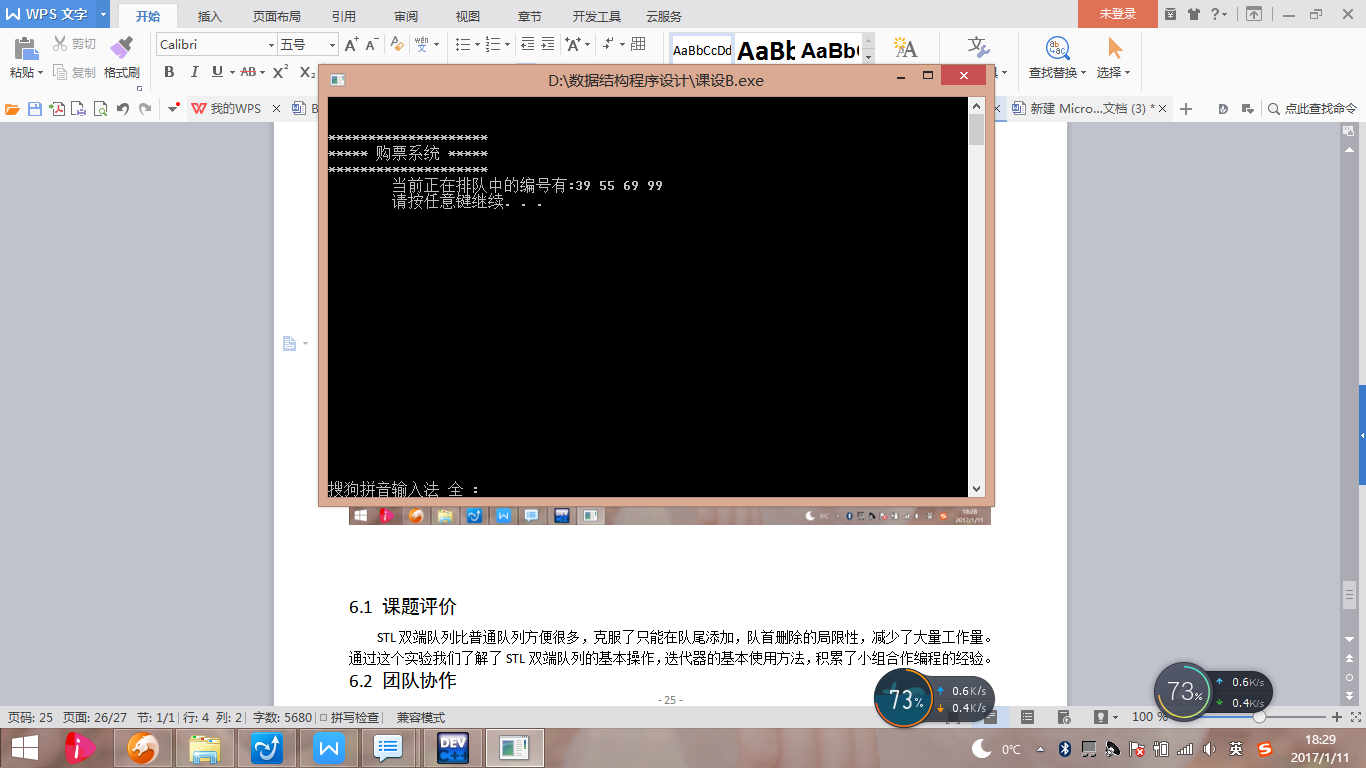


5.1.2张伟颖测试

生成随机指令，队列元素的删除

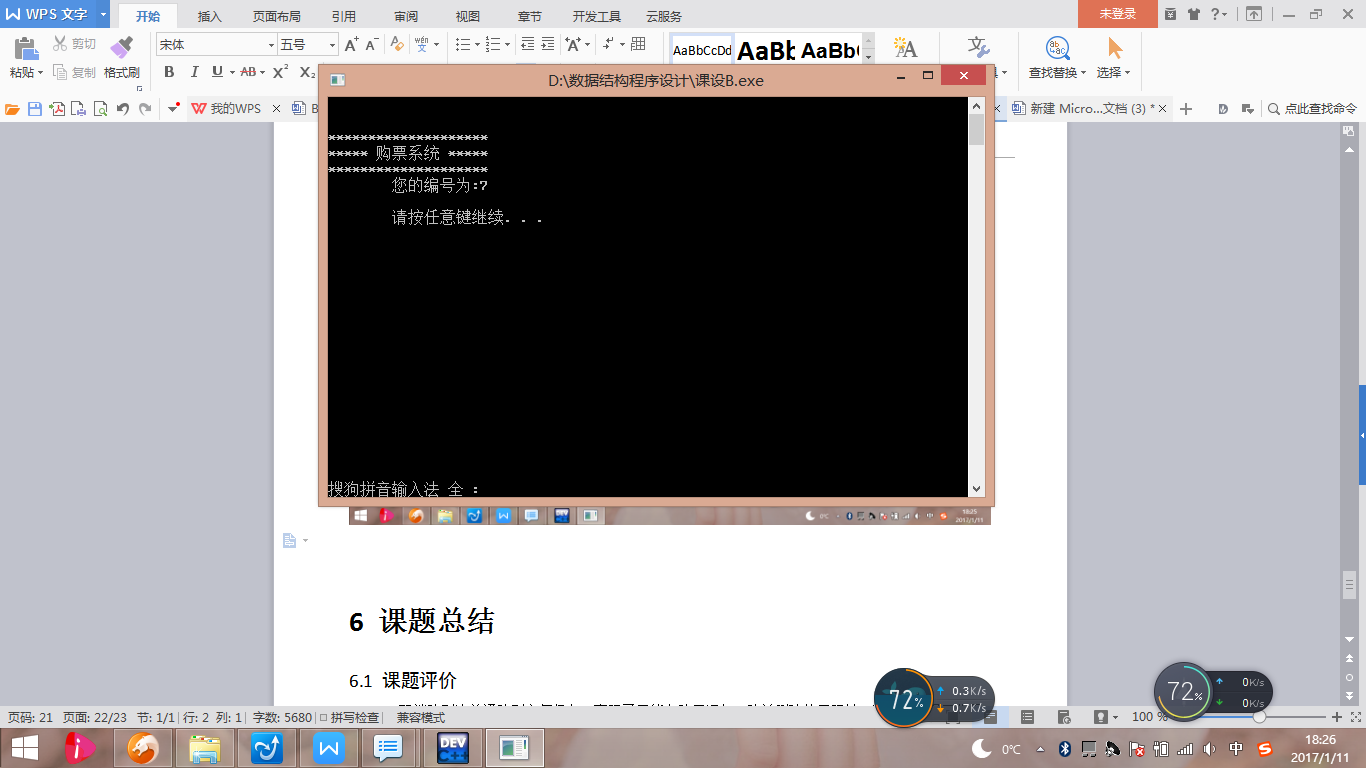


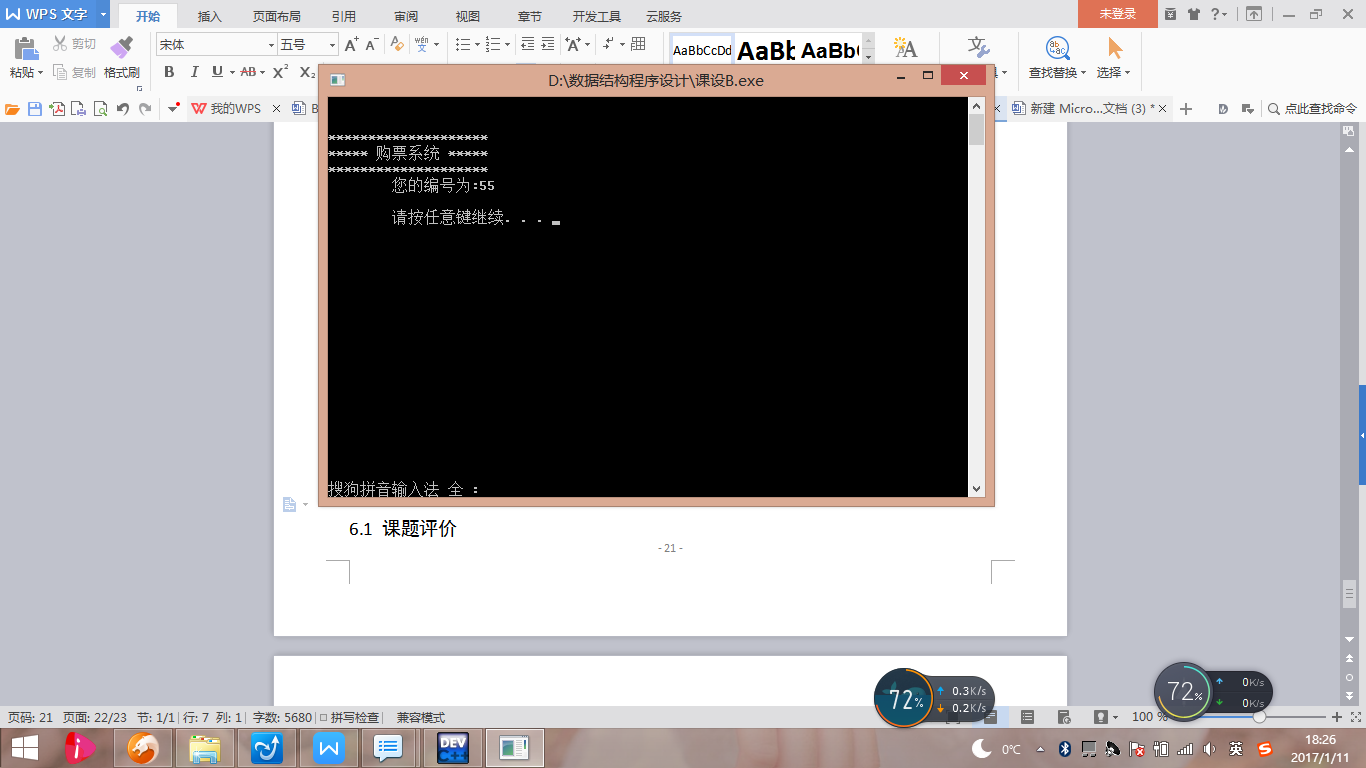


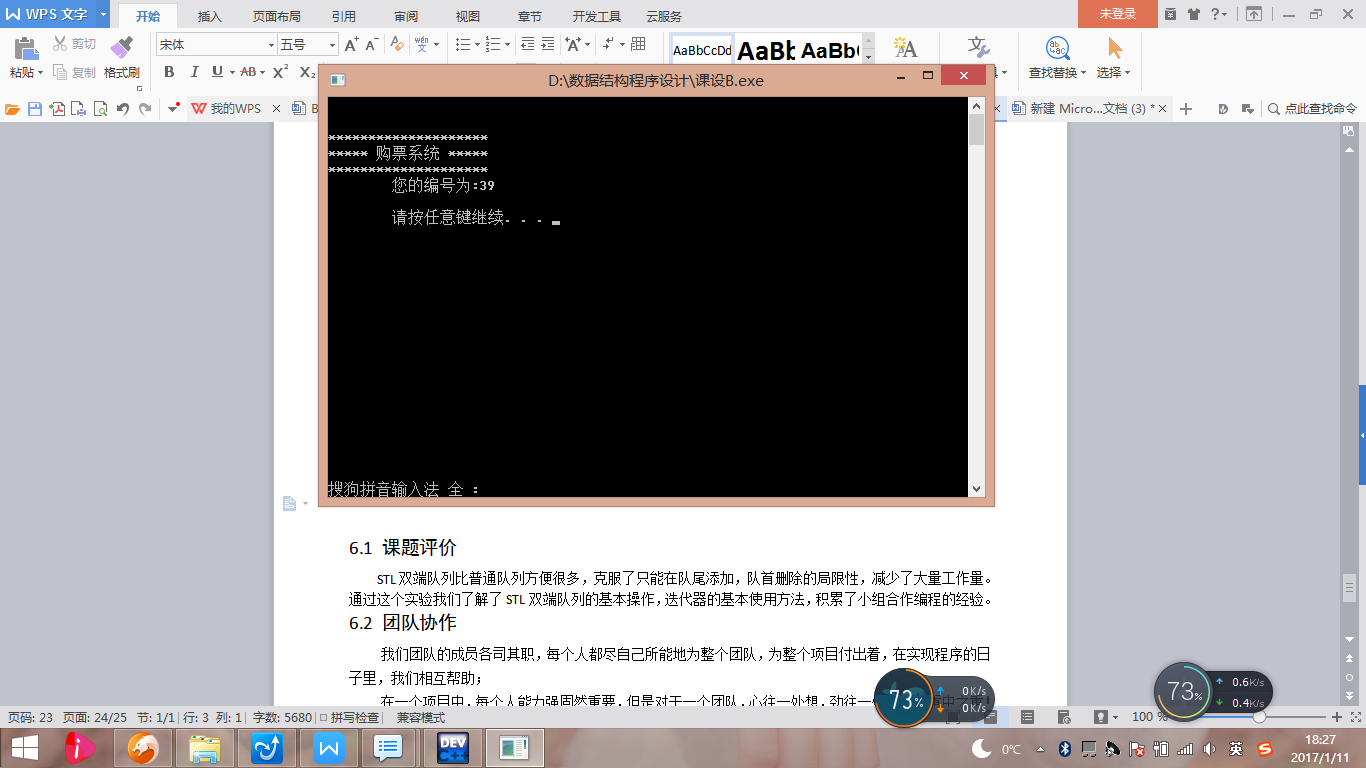


5.1.3 孟霞测试

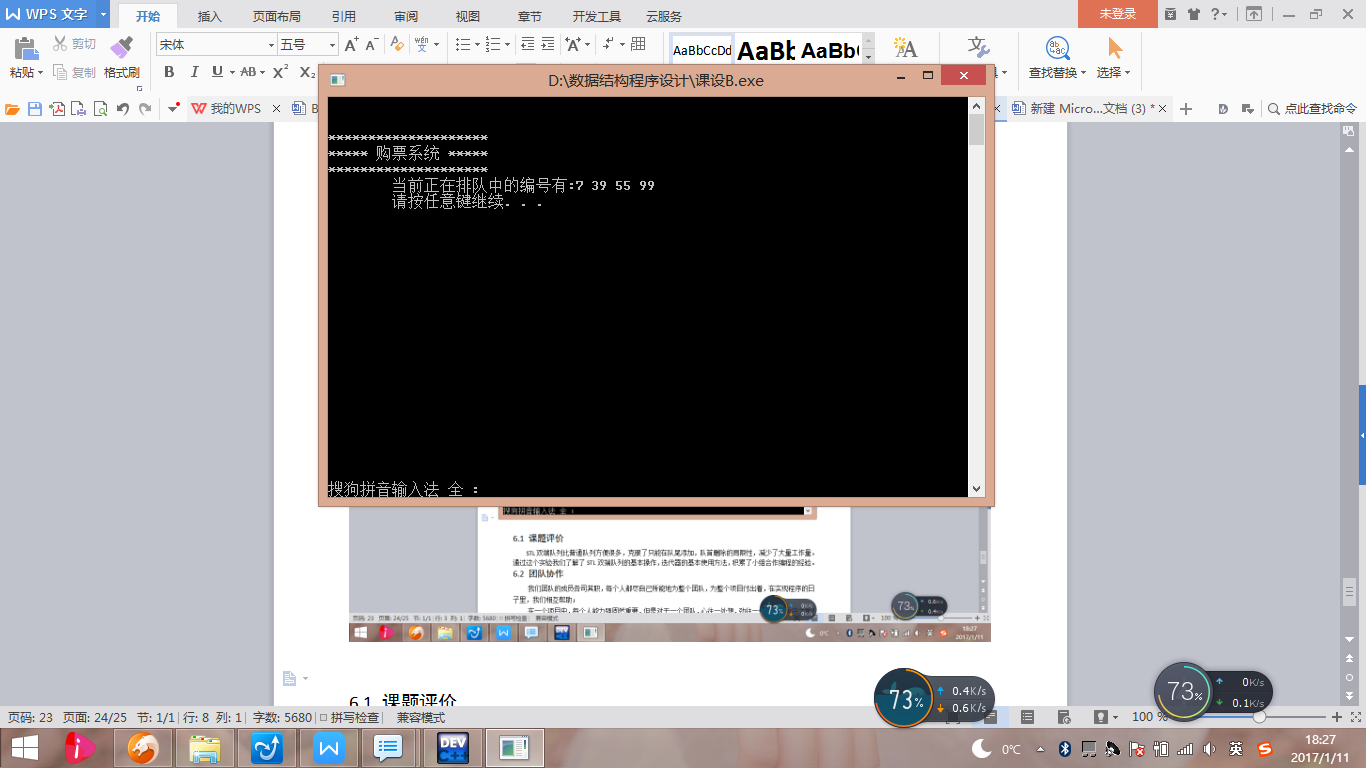
领取随机编号





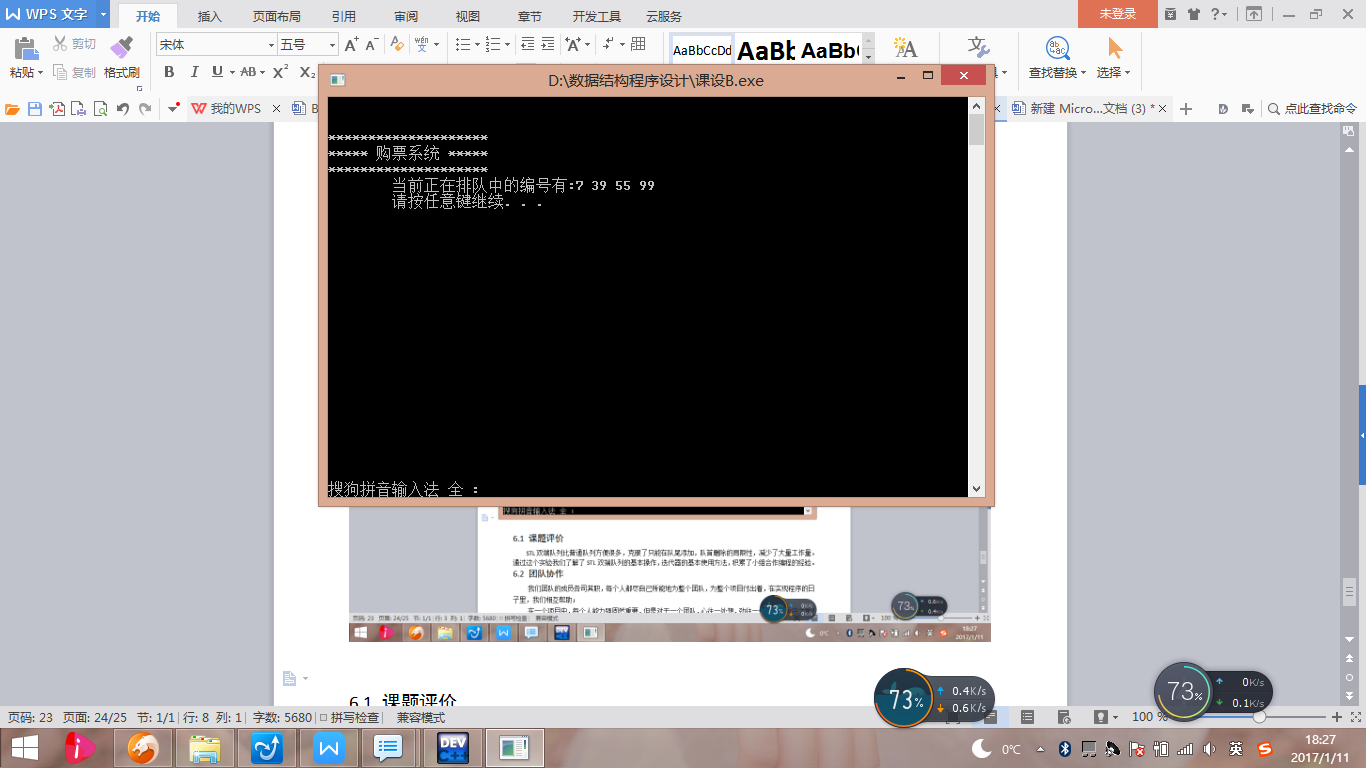


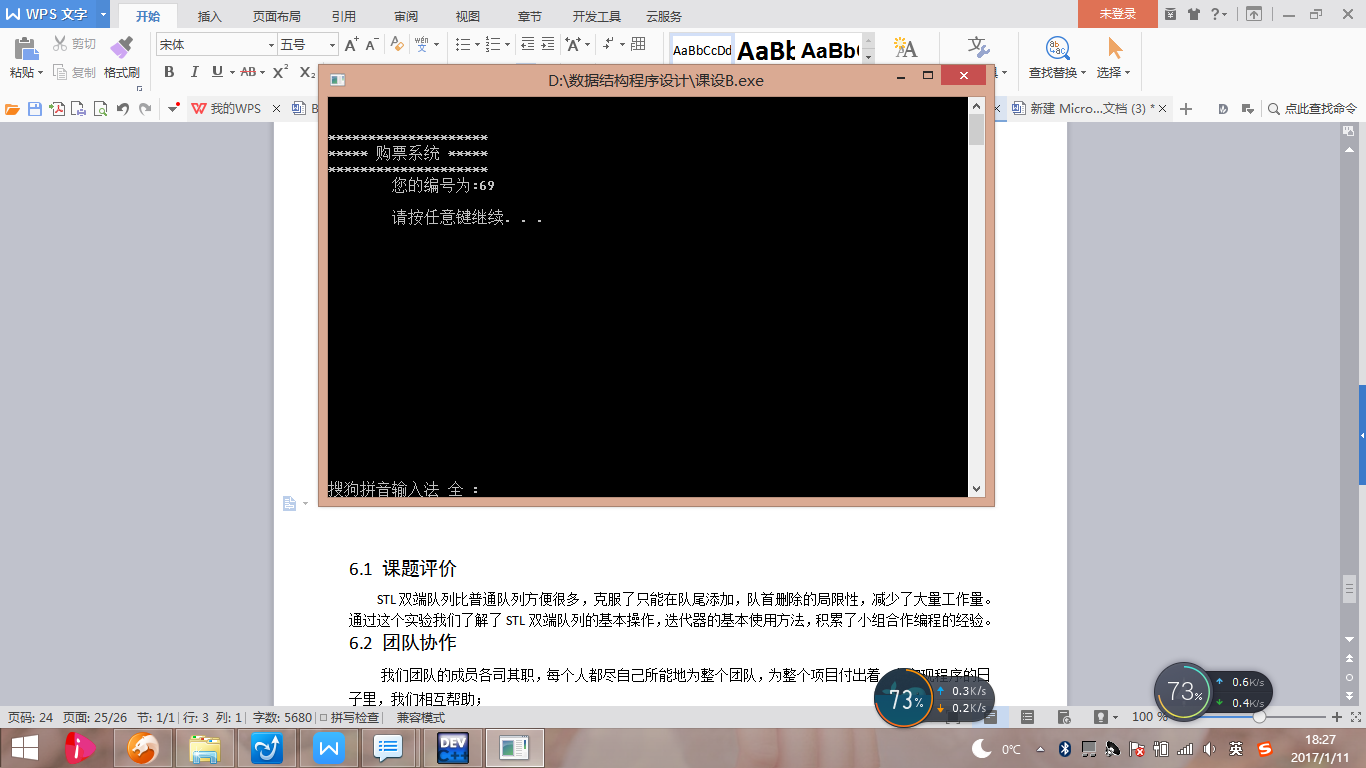


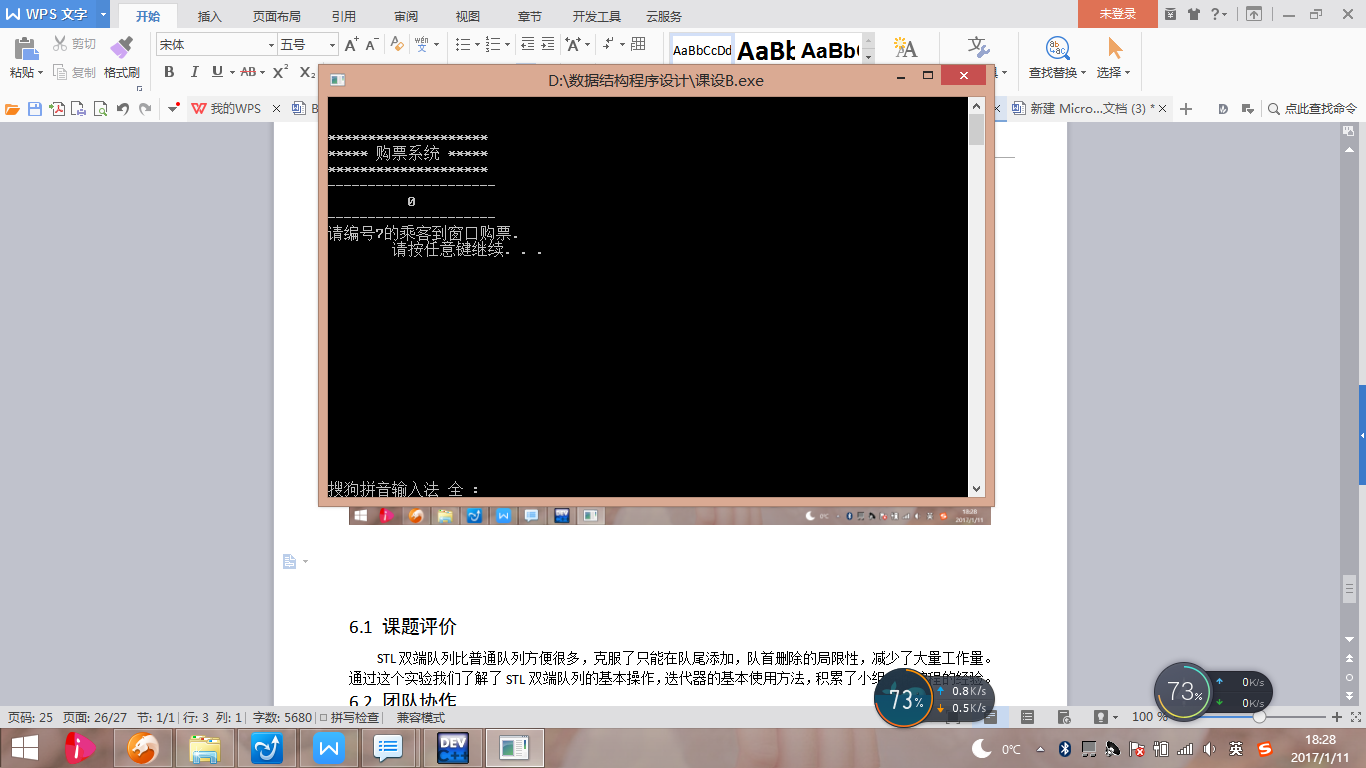


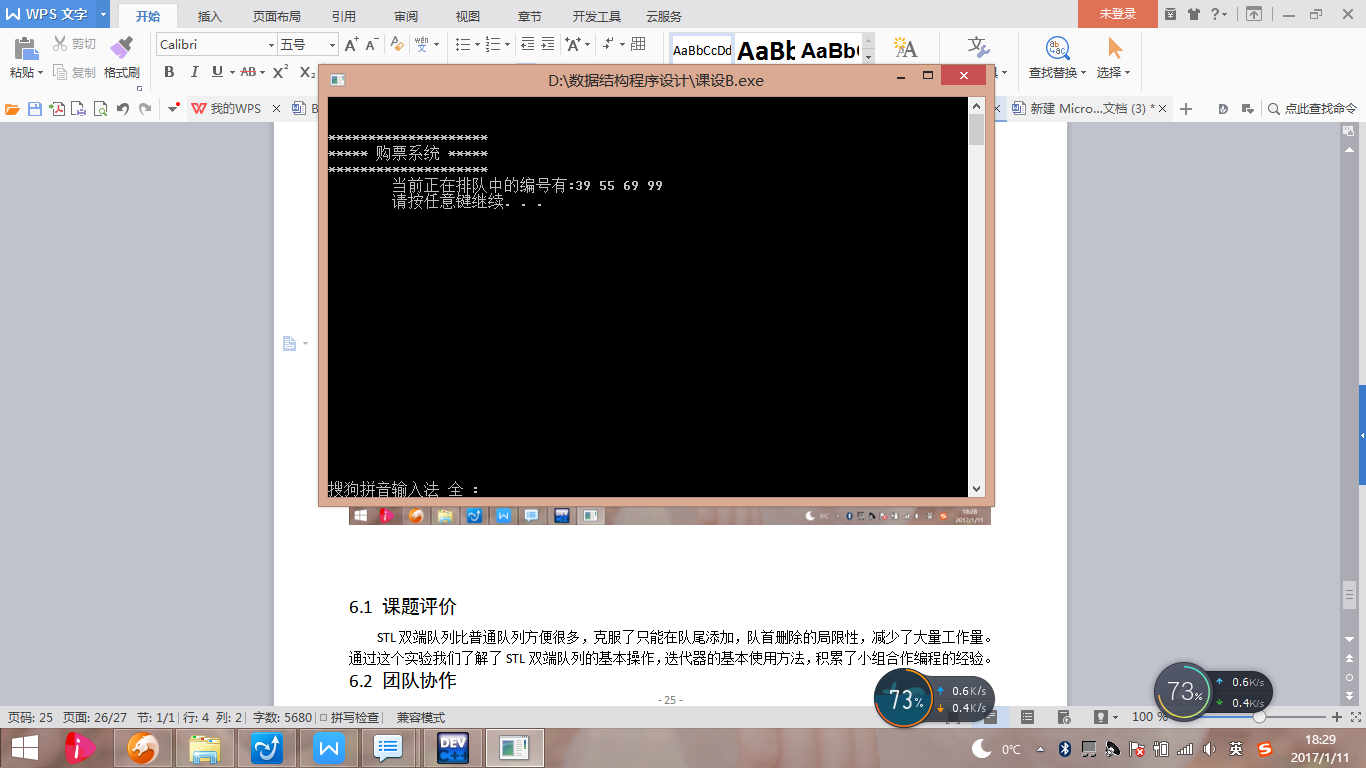
5.2 组装与系统测试

为了测试新生成的编号是否成功添加到队列，指令生成后队列元素是否成功删除，我们分别显示操作前后的队列，进行比较





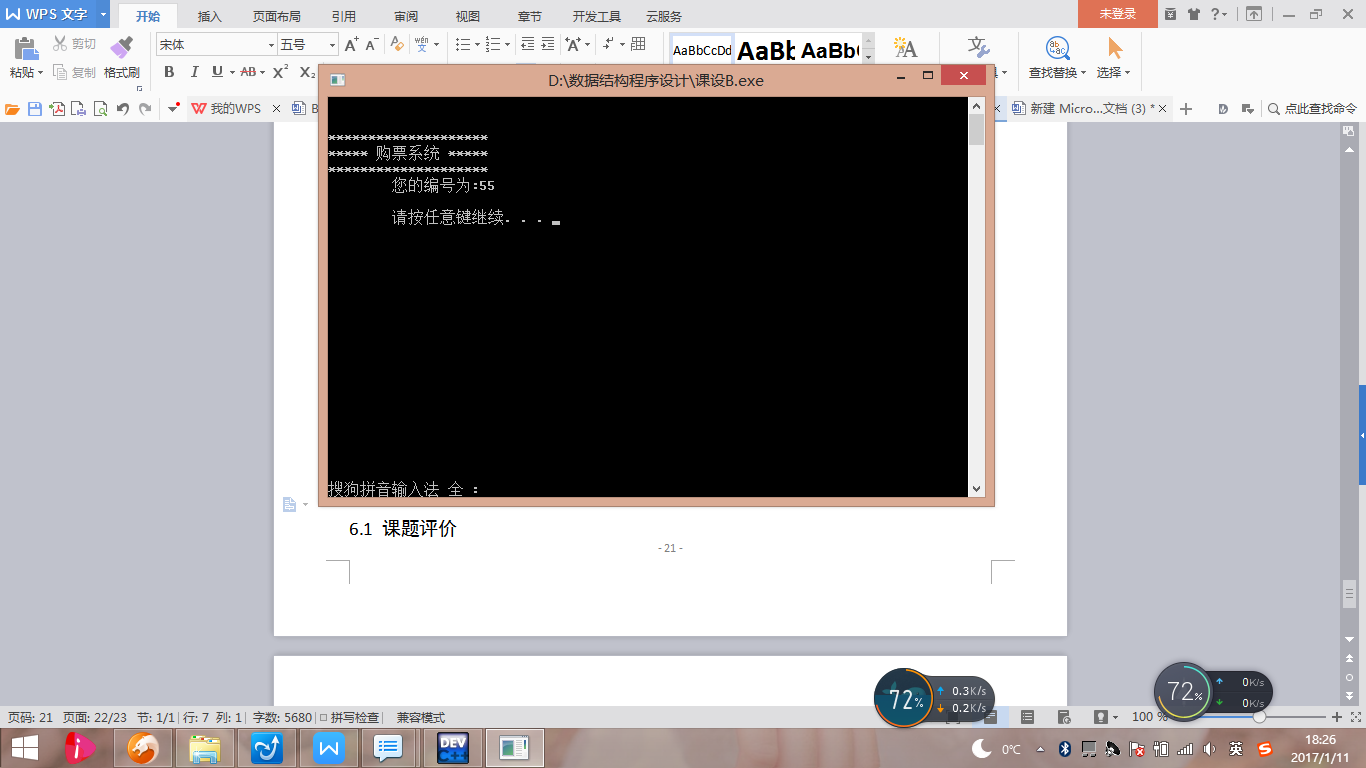




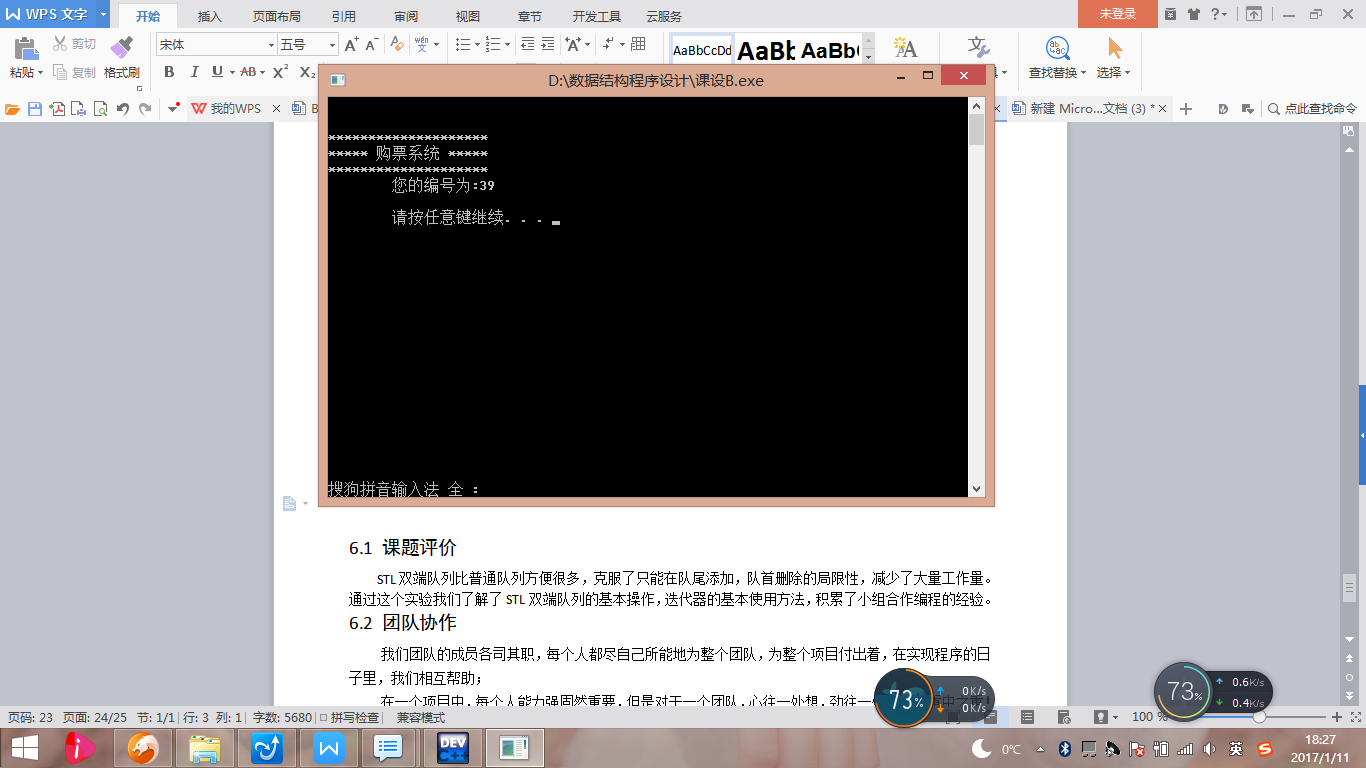
5.3 系统运行

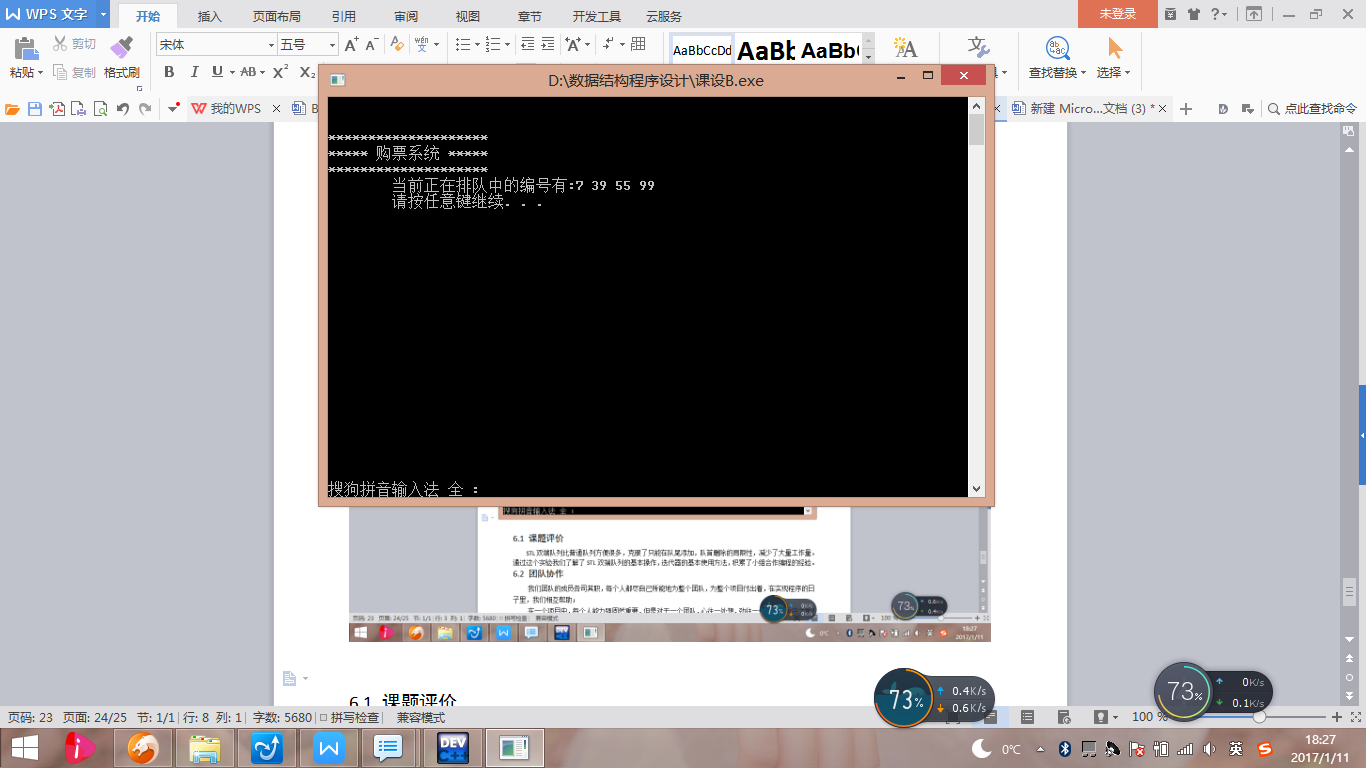


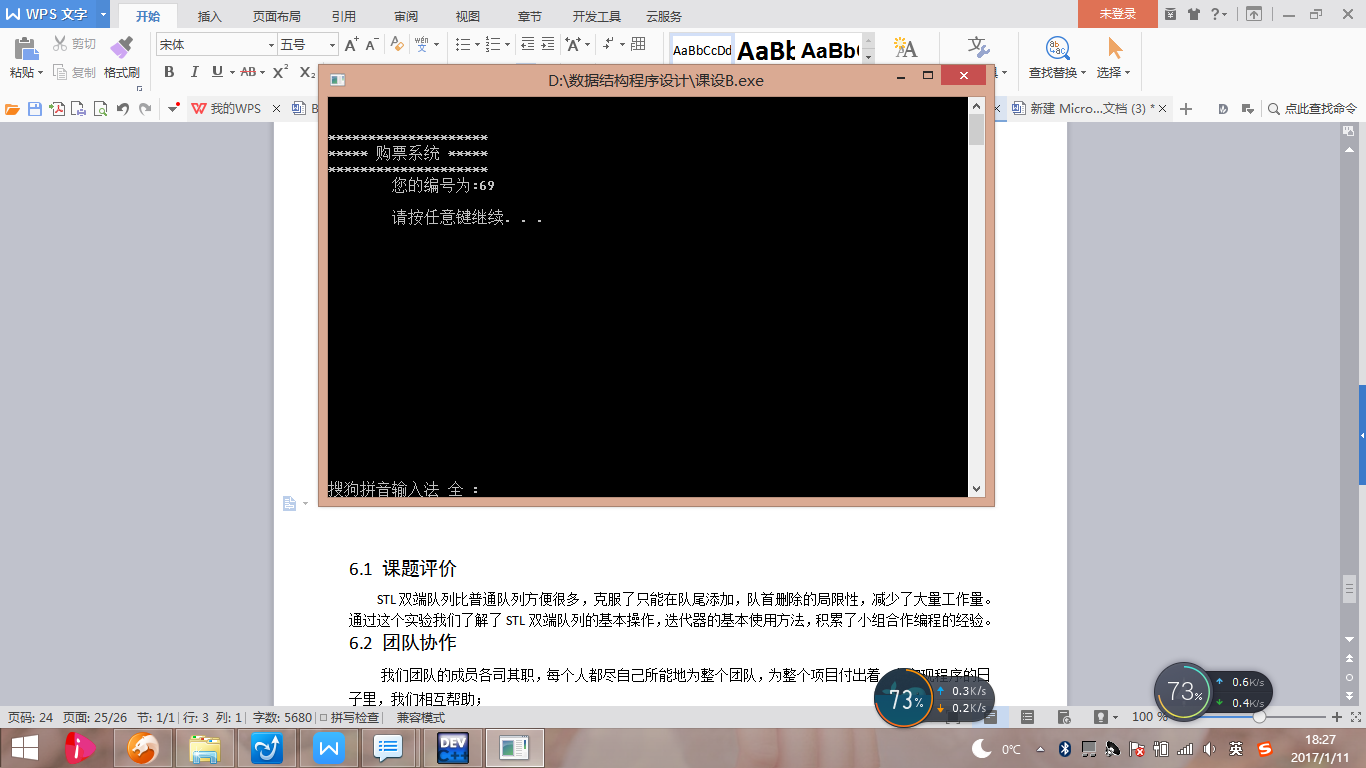
# 

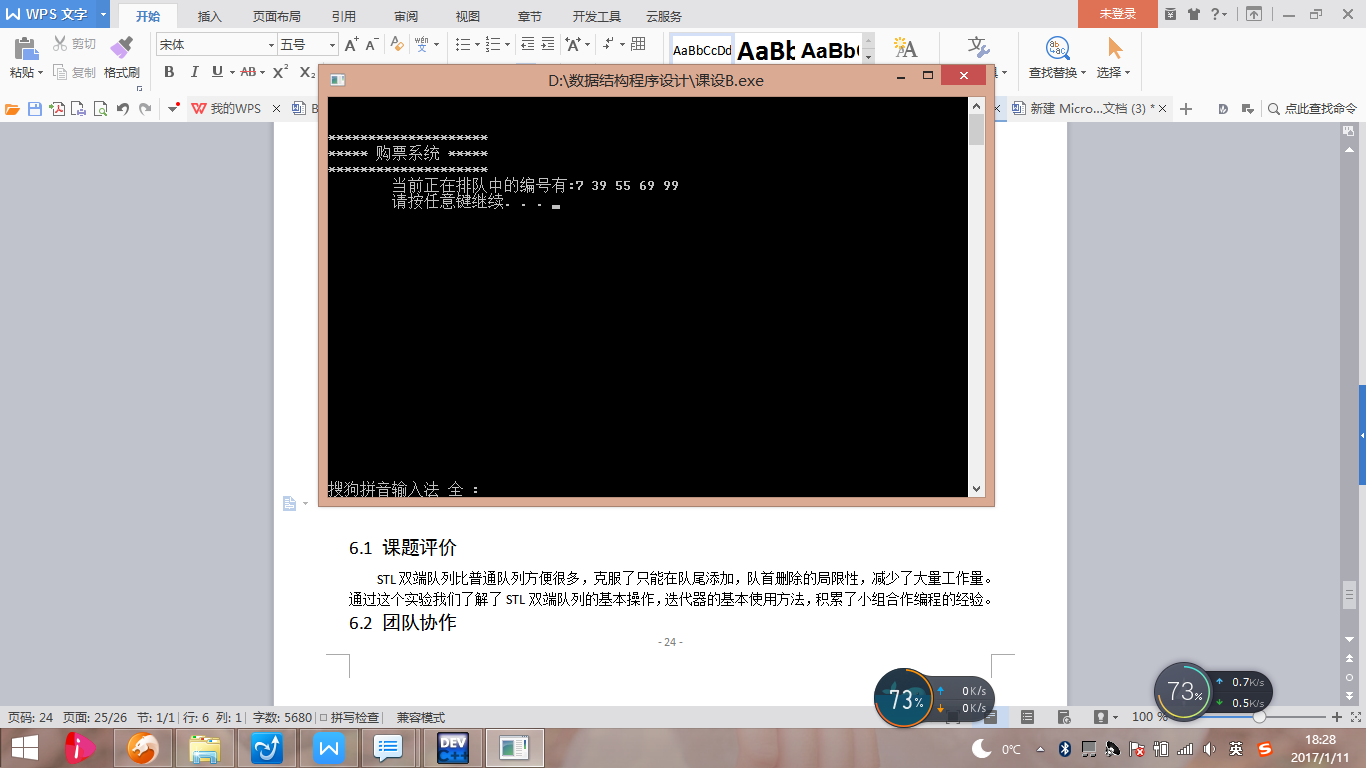


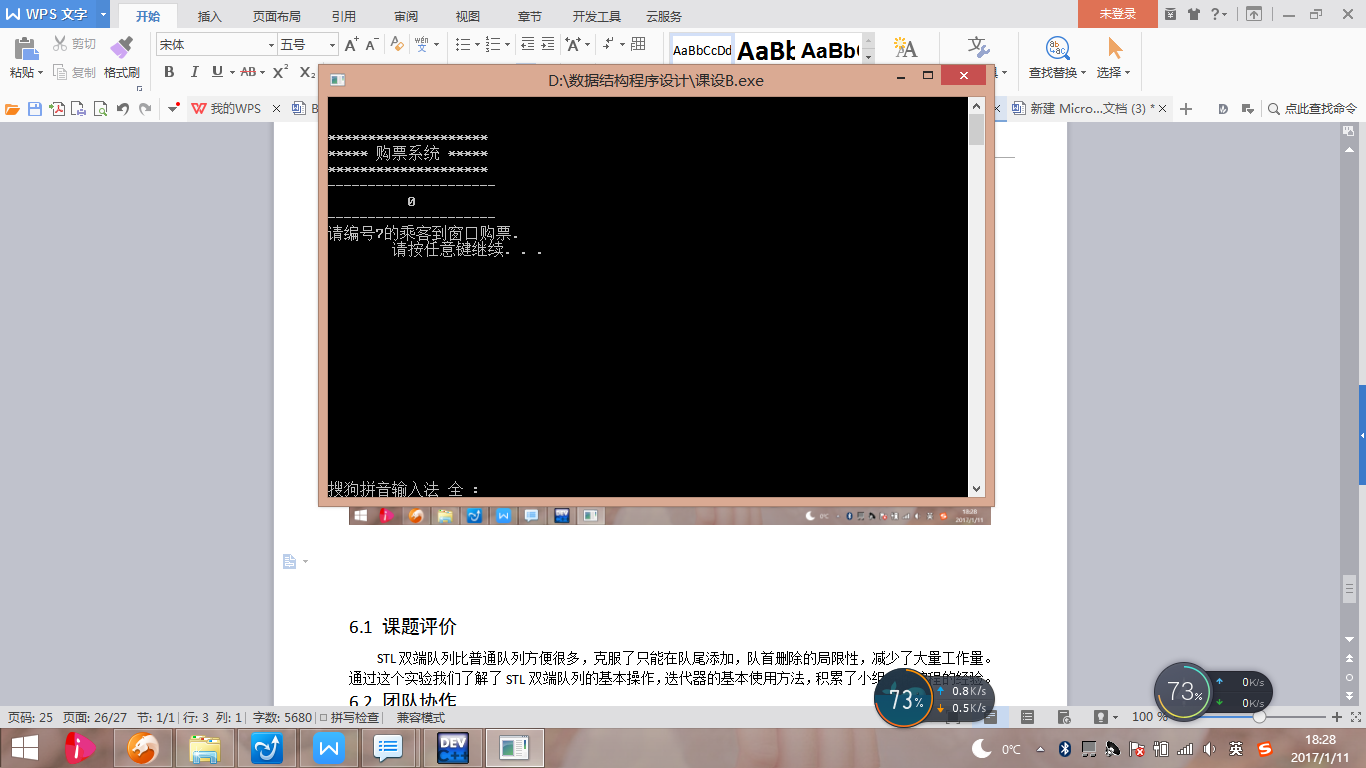


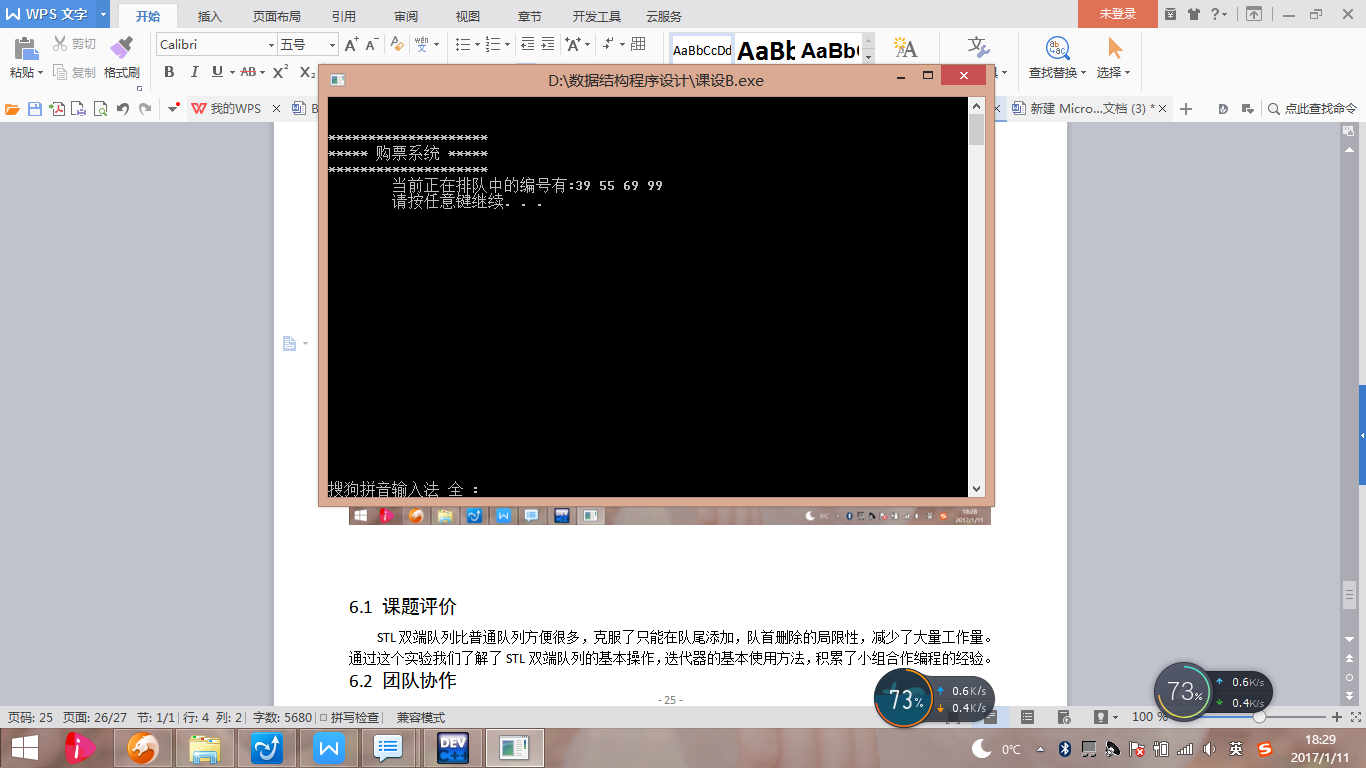


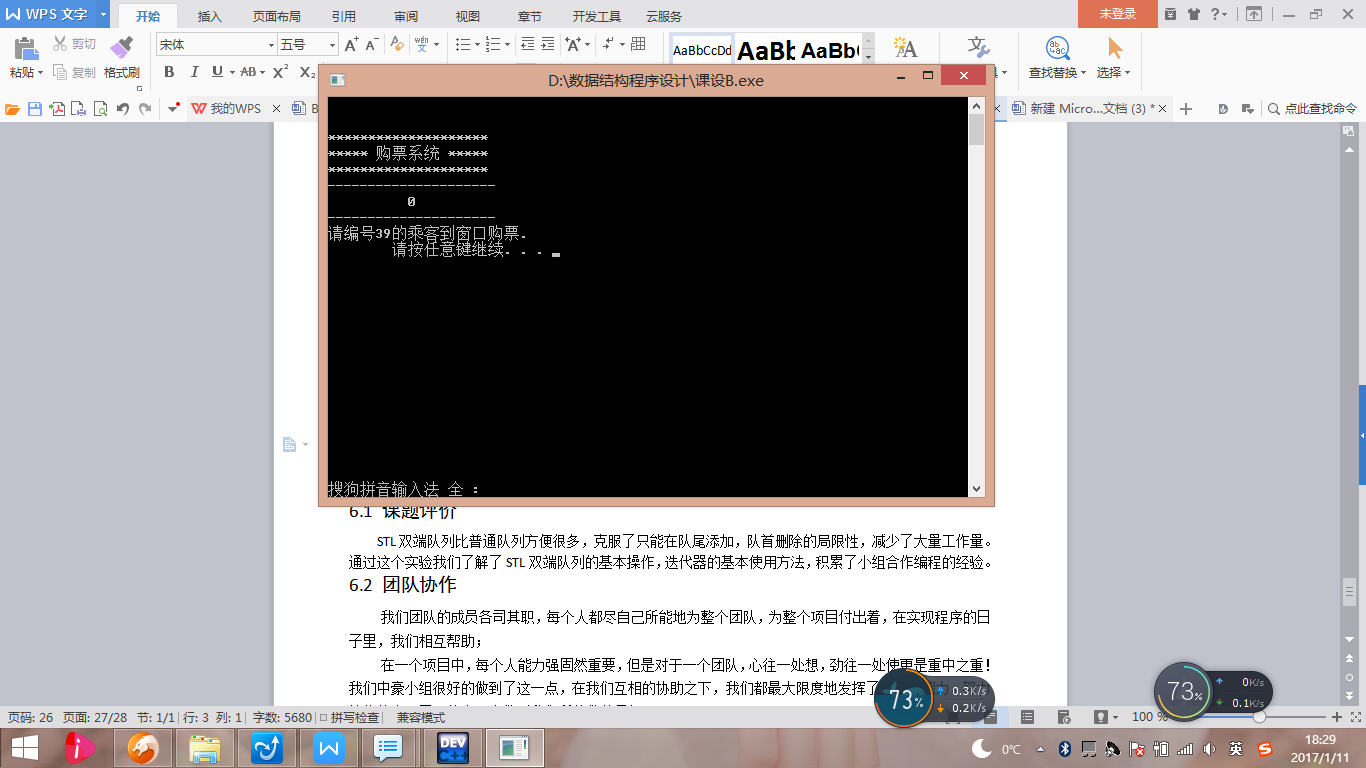


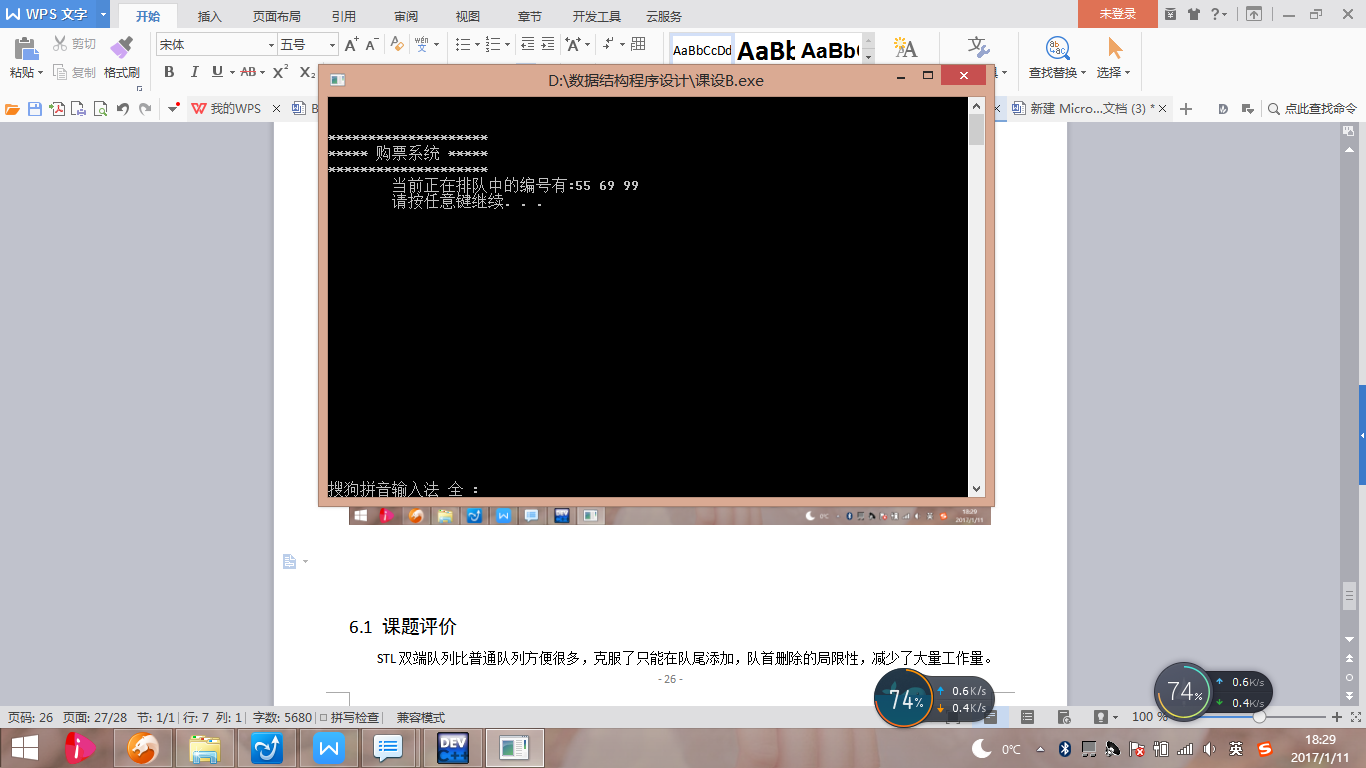


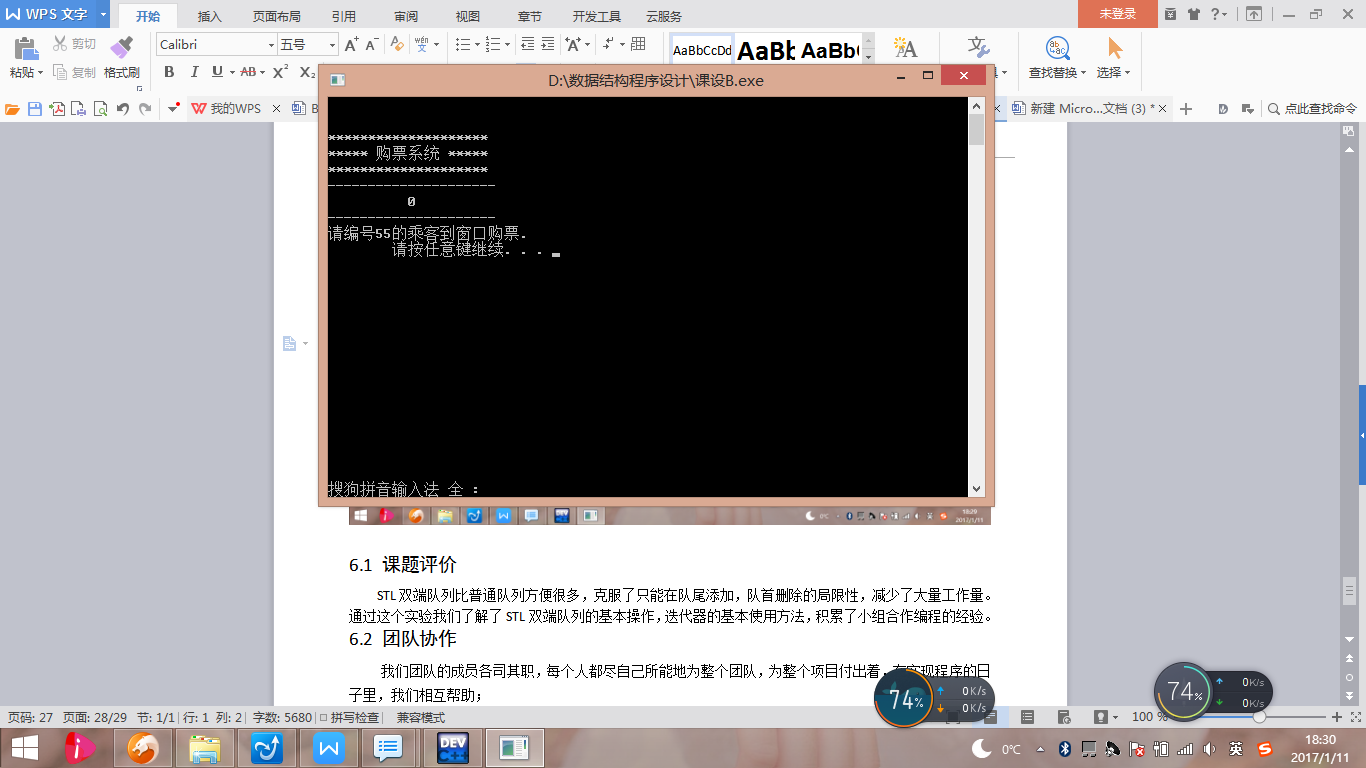


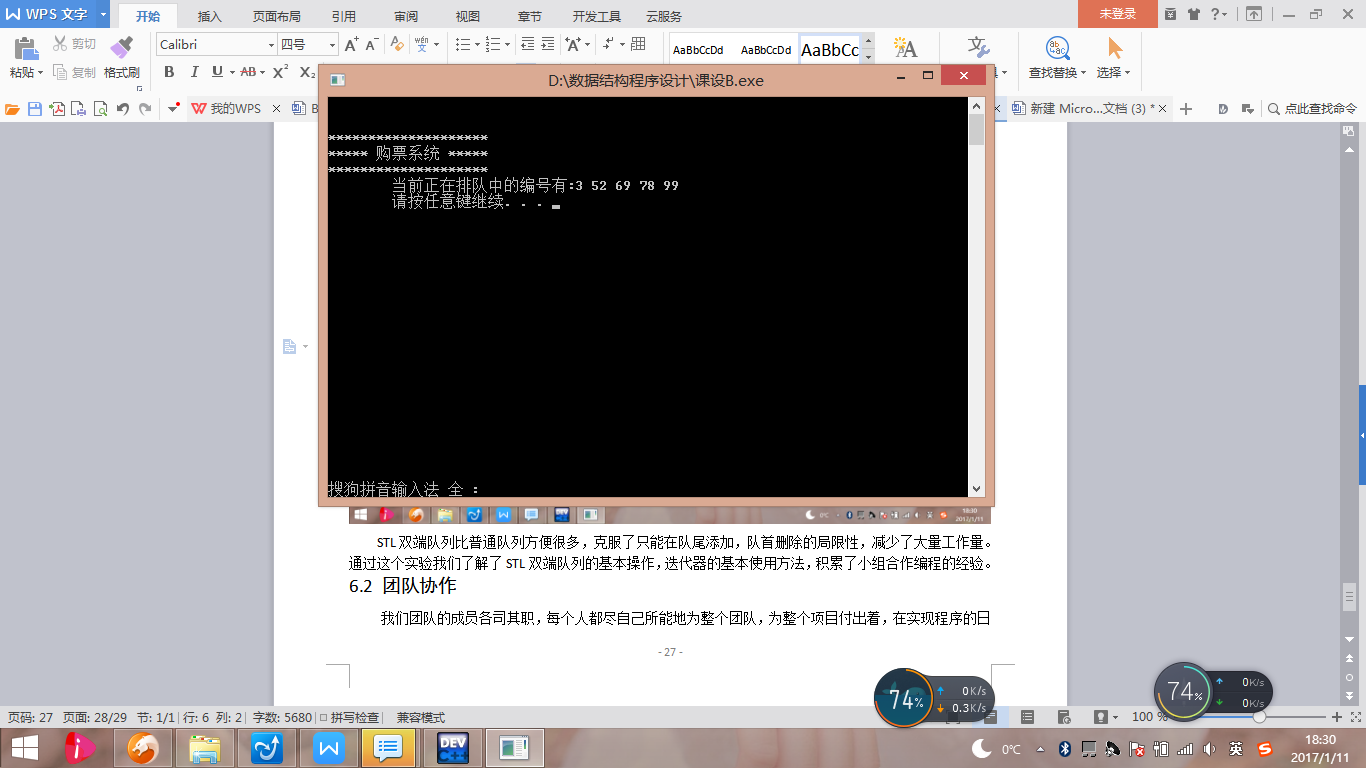












# 6 课题总结

6.1 课题评价

STL双端队列比普通队列方便很多，克服了只能在队尾添加，队首删除的局限性，减少了大量工作量。通过这个实验我们了解了STL双端队列的基本操作以及迭代器的基本使用方法，积累了小组合作编程的经验。

6.2 团队协作

我们团队的成员各司其职，每个人都尽自己所能地为整个团队，为整个项目付出着，在实现程序的日子里，我们相互帮助；

在一个项目中，每个人能力强固然重要，但是对于一个团队，心往一处想，劲往一处使更是重中之重！我们中豪小组很好的做到了这一点，在我们互相的协助之下，我们都最大限度地发挥了自己的能力。努力地将整个项目，整个程序做到我们所能做的最好！

6.3 个人设计小结（按组员分工）

6.3.1 于梓元的小结

转眼，《数据结构》的课程设计接近尾声了，在这次实战中感觉到自己的在编程中的各个不足，只看书不行，只编也不行，而是都要结合起来，才能提高你的编程能力。总结起来，主要是牢固掌握自己的基础知识，没有C语言的知识，数据结构怎么可能学好，C语言在大一学的虽然有些遗忘，但慢慢查查资料，问问老师同学，然后慢慢有思路。所以我告诫自己在学好英语同时一定要牢固掌握专业基础知识。还有要有耐心，细心，严谨的态度。在编程过程中由于少了分号的小错误而导致整个程序运行不起来，不够认真细微，给自己带来了许多麻烦。编程是一件十分严谨的事情，容不得马虎。所以在今后自己一定要培养严谨，细心的态度，这不仅是对程序设计还是做任何事情都应如此。再次充分认识到《数据结构》的重要性，它可以给我们大纲和思想。

6.3.2 张伟颖的小结

在这次实验中，通过查阅资料初次接触到了随机函数以及STL双端队列，在编程实现过程中，与以往知识相比，发现其能大大简化程序，节省内存空间的同时提高了运行效率，充分认识到了算法设计中的效率与低存储量需求的重要性，为以后的算法提供了思想基础。本次实验仅仅用到了STL双端队列的一小部分知识即能把问题解决，其中的许多知识点还有待进一步的学习与掌握。通过此次实验，也充分认识到掌握不同的方法的重要性，在平时的时候应对类似方法加以比较分析，找寻其优劣所在，以便在需要时可以快速选择高效率低存储的方法提高程序运行效率，提高自我以及团队合作编程能力。

6.3.3 孟霞的小结

经过近三周的课程设计，不得不说，我对《数据结构》这门课各章知识的理解更加深入、扎实，在这三周里，我对以往枯燥复杂的算法进行了耐心的阅读和理解，又通过自己编程来练习。可以说，亲自动手编程不仅是一种很有效的练习途径，更是检验自己学习成果的有效途径。但是，只看课本，不足以让我们完成这个程序设计，我们还需要通过网络、课外书等搜找一些相关信息和知识。刚开始时，由于很长时间不曾编程，很多与c语言有关的基本操作都忘了，这个经历更提醒了我平时要多加练习，对学过的知识也要时时复习。在编程中，我们需要一种踏实严谨的态度，不能求速成，马马虎虎，否则在编译时会很难通过，给实验进程带来很大阻碍。