// SelectingCoursesSystem.cpp : 定义控制台应用程序的入口点。

//

/\*

第二周 第三题

题目：

3)

设计一个学生选课系统

假设每个学生最多只能选修5门课程，

每门课程最多30名学生选修。

根据用户要求，

可设置并能够得到学生所选课程名称；

给出学生姓名，可得到某门课成绩，

也可得到学生所有课程平均成绩；

给出课程名，可设置选修学生、

设置或修改选修学生的成绩，

也可计算该门课程的平均成绩。

编译环境：Visual Studio 2015

作者：杨政权

\*/

#include "stdafx.h"

#include <iostream>

#include<stdio.h>

#include<stdlib.h>

using namespace std;

int N1, N2;

struct couse \* head1;

struct student \* head2;

//课程信息结构体

struct couse

{

int num1;

char name1[100];

float score;

//课程已选人数

int kcyx;

//课程人数上限

int kcsx;

struct couse \* next;

};

//学生信息结构体

struct student

{

int num2;

char name2[100];

//已选课程编号

int nelenum[100];

//已选课程数量

int nelen;

struct student \* next;

};

//录入课程子函数

void keyboardc()

{

struct couse \*p1, \*p2;

N1 = 0;

p1 = p2 = (struct couse\*)malloc(sizeof(struct couse));

cout << '\t' << '\t';

cout << "\*\*\*\*\*\*录入课程以编号为0结束\*\*\*\*\*\*" << endl << endl;

cout << "课程编号\t课程名称\t学分" << endl;

cin >> p1->num1 >> p1->name1 >> p1->score;

p1->kcyx = 0;

p1->kcsx = 30;

head1 = NULL;

while (p1->num1 != 0)

{

N1 = N1 + 1;

if (N1 == 1)head1 = p1;

else p2->next = p1;

p2 = p1;

p1 = (struct couse \*)malloc(sizeof(struct couse));

cin >> p1->num1 >> p1->name1 >> p1->score;

p1->kcyx = 0;

p1->kcsx = 30;

}

p2->next = NULL;

}

//录入学生信息子函数

void keyboards()

{

int i;

struct student \*p1, \*p2;

N2 = 0;

p1 = p2 = (struct student \*)malloc(sizeof(struct student));

cout << '\t' << '\t';

cout << "\*\*\*\*\*\*录入学生信息以0结束\*\*\*\*\*\*" << endl << endl;

cout << "学生学号\t学生姓名" << endl;

cin >> p1->num2 >> p1->name2;

p1->nelen = 0;

for (i = 0; i<30; i++) p1->nelenum[i] = 0;

head2 = NULL;

while (p1->num2 != 0)

{

N2 = N2 + 1;

if (N2 == 1)head2 = p1;

else p2->next = p1;

p2 = p1;

p1 = (struct student \*)malloc(sizeof(struct student));

cin >> p1->num2 >> p1->name2;

p1->nelen = 0;

for (i = 0; i<30; i++) p1->nelenum[i] = 0;

}

p2->next = NULL;

}

//选课

void elect(struct student \* s)

{

struct couse \* p;

int num1, i;

//判断选满5门课

if (s->nelen == 5)

cout << "你已经选完五门课程" << endl;

else

{

cout << "请输入要选课的编号:" << endl;

cin >> num1;

p = head1;

while (p->num1 != num1) p = p->next;

//一门课已经有30人选

if (p->kcyx == 30)

cout << "该门课程已被选满!" << endl;

else

{

(p->kcyx)++;

for (i = 0; s->nelenum[i] != 0; i++);

s->nelenum[i] = num1;

(s->nelen)++;

cout << "选课成功!"<<endl;

}

}

}

//学生选课子函数(查询可选课程)

void cheakkx()

{

char e;

struct couse \* c;

struct student \* s;

int num2, i, t = 0;

cout << "请输入你的学号:"<<endl;

cin >> num2;

s = head2;

while (s->num2 != num2 && s->next != NULL) s = s->next;

c = head1;

cout << "你的可选课程编号:"<<endl;

while (c != NULL)

{

for (t = 0, i = 0; s->nelenum[i] != 0; i++)

{

if (c->num1 == s->nelenum[i]) t = 1;

}

if (t == 0 && (c->kcyx < c->kcsx))

cout << c->num1 << endl;

c = c->next;

}

cout << "选课(y/n)?:" << endl;

getchar();

e = getchar();

if (e == 'y')

elect(s);

}

//退课

void back(struct student \* p)

{

int num1, i, j;

cout << "请输入你要退掉的课程编号:"<<endl;

cin >> num1;

while (head1->num1 != num1) head1 = head1->next;

for (i = 0; p->nelenum[i] != num1; i++);

for (j = i; p->nelenum[j] != 0; j++) p->nelenum[j] = p->nelenum[j + 1];

p->nelenum[--j] = 0;

(head1->kcyx)--;

cout << "退课成功!" << endl;

}

//学生选课子函数（查询已选课程)

void cheakyx()

{

char e;

struct student \* p;

int num2, i;

float f = 0.0;

cout << "请输入学号:"<<endl;

cin >> num2;

p = head2;

while (p->num2 != num2 && p != NULL) p = p->next;

cout << "已选课程编号:"<<endl;

for (i = 0; p->nelenum[i] != 0; i++)

cout << p->nelenum[i]<<endl;

cout << "是否进行退课(y/n)?"<<endl;

getchar();

e = getchar();

if (e == 'y')

{

back(p);

(p->nelen)--;

}

}

//学生选课主函数

void elective()

{

int i;

cout << "\t\t\t学生选课"<<endl;

cout << "1.查询可选课程" << endl;

cout << "2.查询已选课程" << endl;

cout << "3.返回主菜单" << endl;

cout << "请输入(1~3):" << endl;

cin >> i;

switch (i)

{

case(1):cheakkx(); break;

case(2):cheakyx(); break;

case(3):break;

}

}

//输出课程信息

void listc()

{

struct couse \* p;

p = head1;

cout<<"课程编号 课程名称 学分 课程已选人数 课程人数上限"<<endl;

while (p != NULL)

{

printf("%-8d%10s%6.1f%8d%12d\n", p->num1, p->name1, p->score, p->kcyx, p->kcsx);

p = p->next;

}

}

//输出学生信息

void lists()

{

int i;

float f = 0.0;

struct couse \* p0;

struct student \* p1;

p1 = head2;

cout << "学生学号 学生姓名 已选课程数量 总学分"<<endl;

while (p1 != NULL)

{

float f = 0.0;

for (i = 0; p1->nelenum[i] != 0; i++)

{

p0 = head1;

while (p0->num1 != p1->nelenum[i]) p0 = p0->next;

f = f + p0->score;

}

printf("%-4d %10s %10d %15.1f\n", p1->num2, p1->name2, p1->nelen, f);

p1 = p1->next;

}

}

//信息主函数

void store()

{

int i;

cout << "\t\t\*\*\*\*\*\*浏览课程及学生信息\*\*\*\*\*\*"<<endl<<endl;

cout << "1.查看课程信息" << endl;

cout << "2.查看学生信息" << endl;

cout << "3.返回主菜单" << endl;

cout << "请输入(1~3):" << endl;

cin >> i;

switch (i)

{

case(1):listc(); break;

case(2):lists(); break;

case(3):break;

}

}

int main()

{

int i;

start:

//欢迎界面

cout<<"\n\t\t\*\*\*\*\*\*欢迎使用学生选课系统!\*\*\*\*\*\*" << endl << endl;

cout<<"菜单:" << endl;

cout<<"1.录入课程信息" << endl;

cout<<"2.录入学生信息" << endl;

cout<<"3.学生选课及退课" << endl;

cout<<"4.浏览课程及学生信息" << endl;

cout<<"5.退出系统" << endl;

cout<<"\n请输入菜单选项(1~5):" << endl;

cin >> i;

//命令选择

switch (i)

{

case(1):

{

system("cls");

keyboardc();

goto start;

break;

}

case(2):

{

system("cls");

keyboards();

goto start;

break;

}

case(3):

{

system("cls");

elective();

goto start;

break;

}

case(4):

{

system("cls");

store();

goto start;

break;

}

case(5):

{

system("cls");

cout<<"感谢使用学生选课系统!"<<endl<<endl;

cout << '\t';

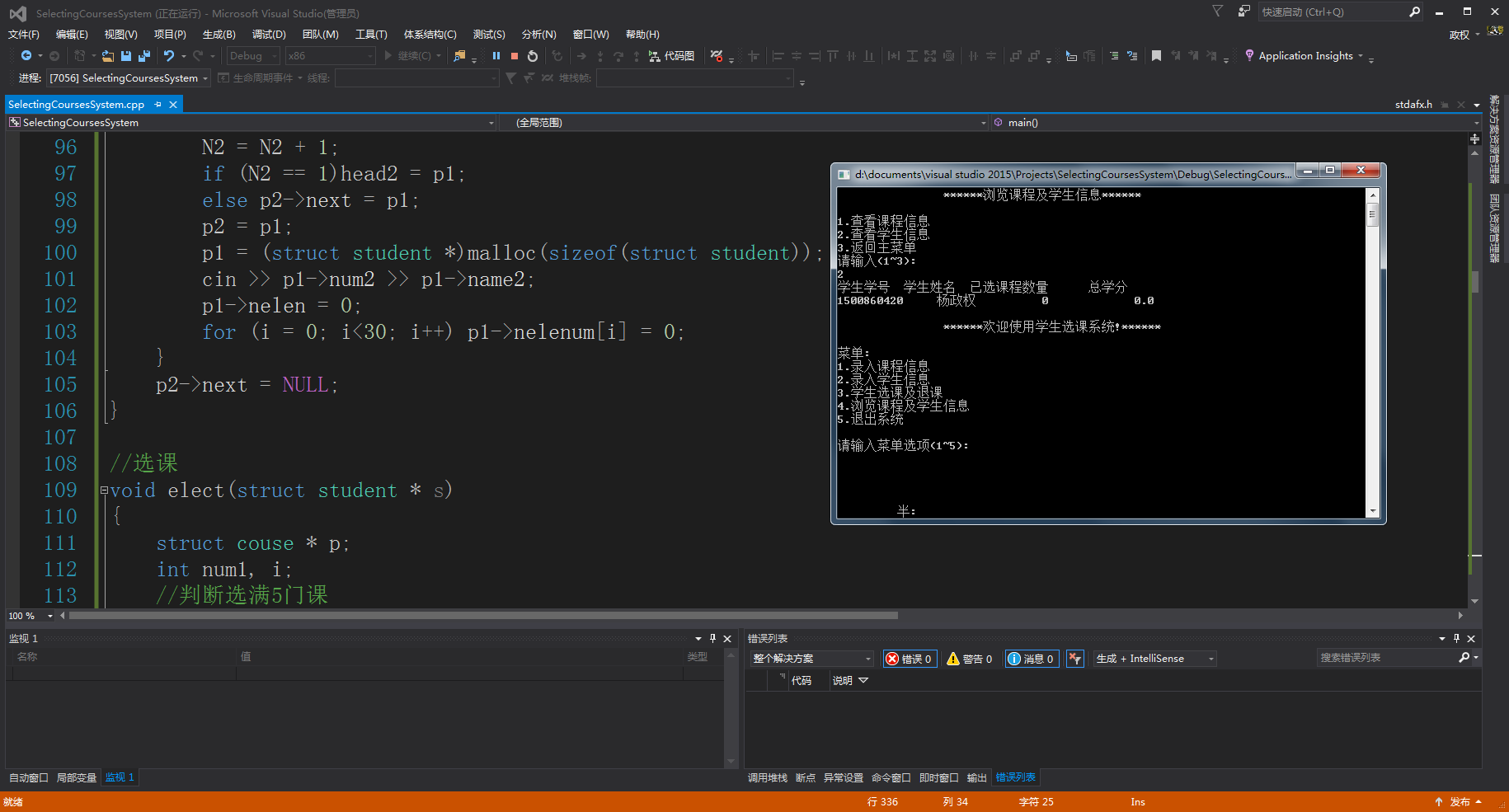
cout << "再见！" << endl;

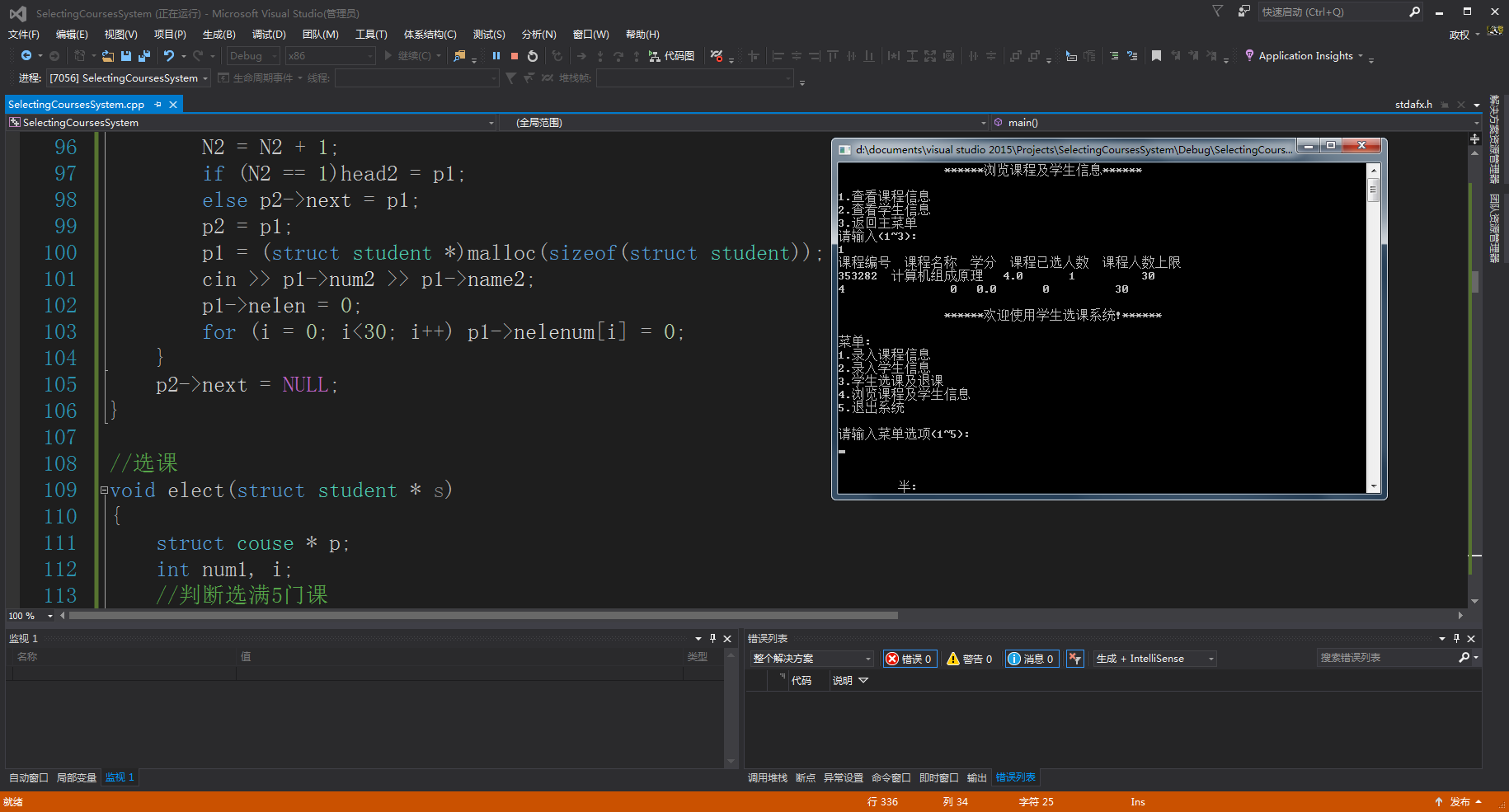
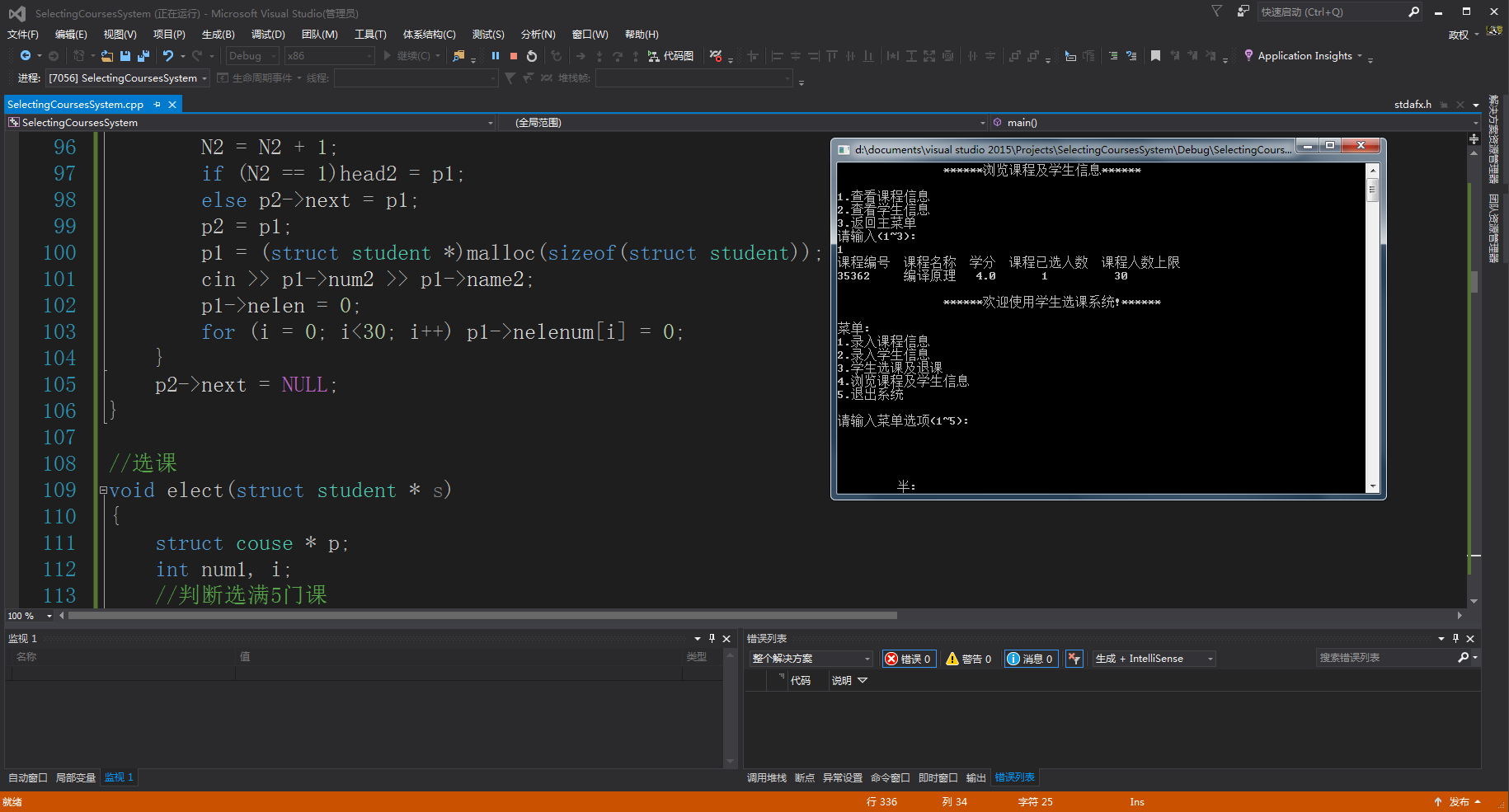
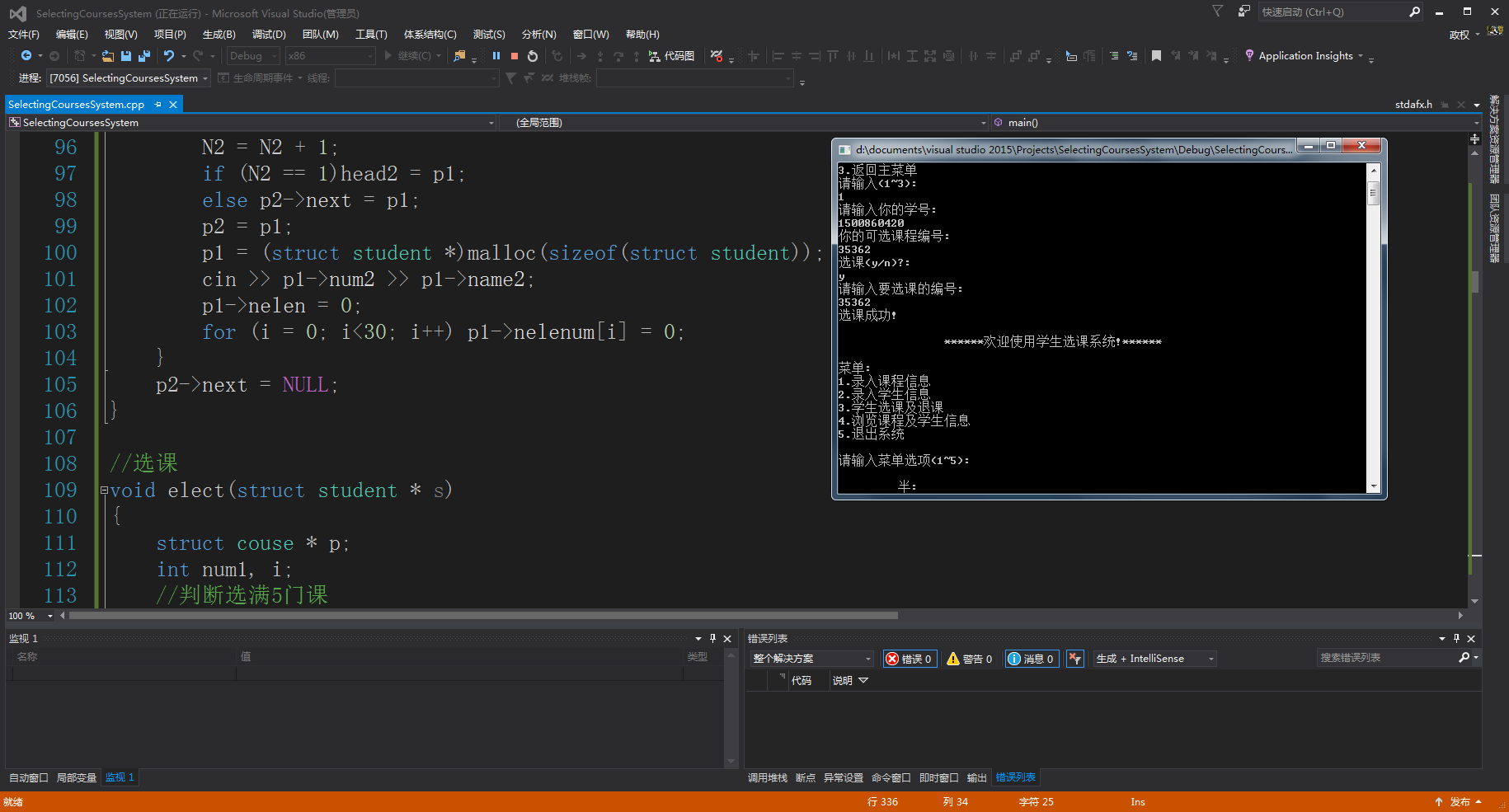
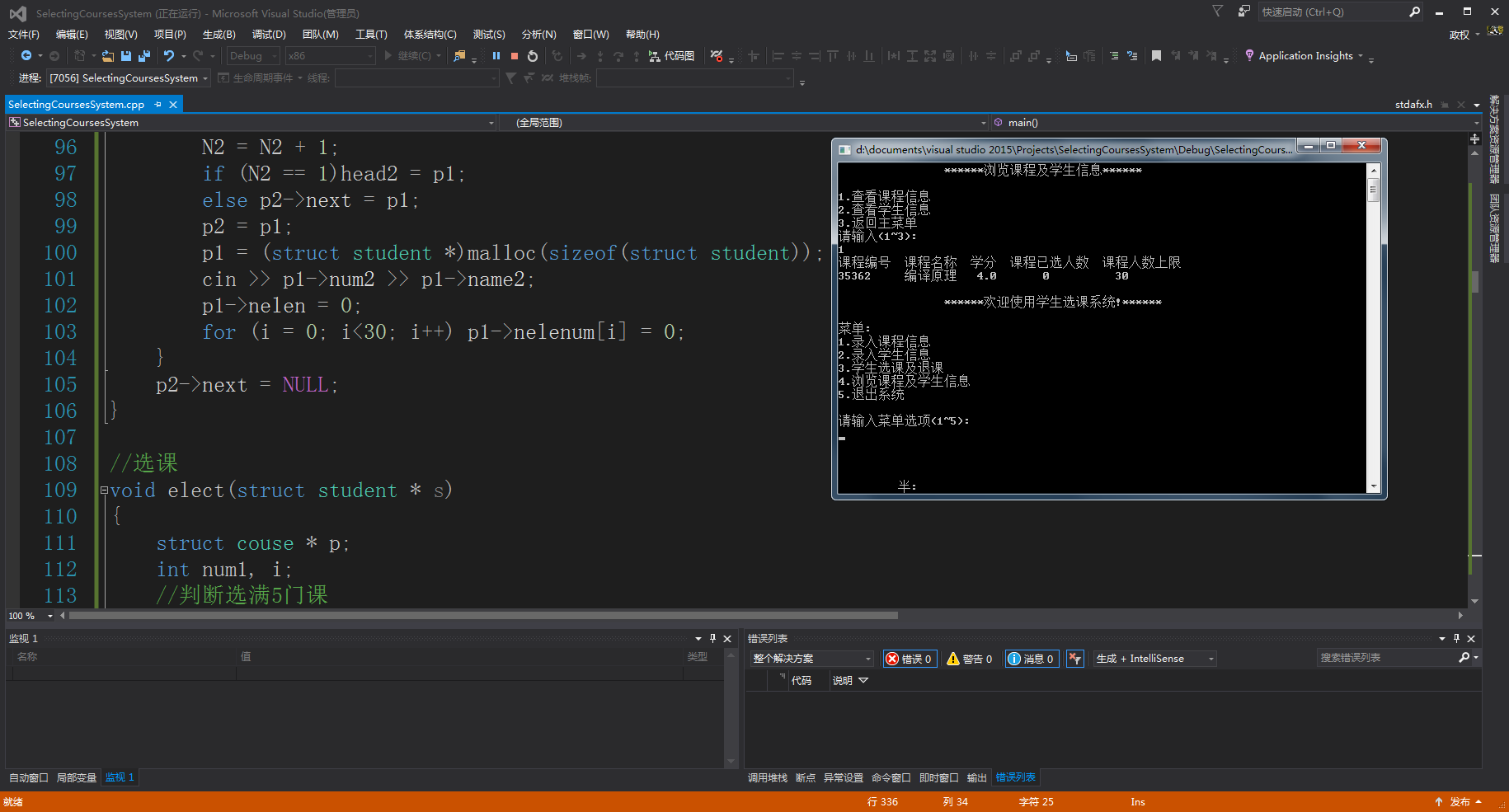
}

}

}

//程序运行截图：



//实验总结：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* 这道题本来是留到最后来做一个大系统的

\*\*\* 但是由于时间关系,之前在做方式1的课程设计

\*\*\* 所以，只做了一部分功能

\*\*\* 不过也在题目的基础上加了一些其他功能

\*\*\* 也算弥补了一些吧

\*\*\* 时间不够，没办法了。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/