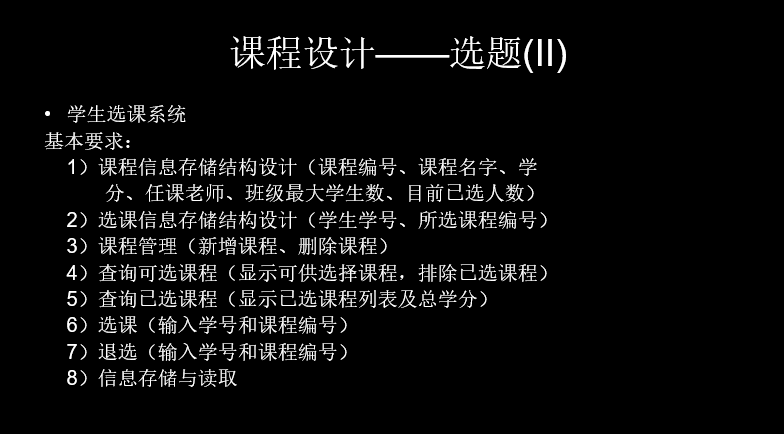
### 题目要求



### 题目分析及设计方案

对于一个选课系统来说，一定会有许多的功能，且这些功能都可以是用户选择性使用的，所以我们需要有一个菜单函数，通过输入不同的数字来选择不同的功能，按照题目要求，基本的功能分为

1. 添加课程
2. 查询课程信息
3. 删除课程
4. 查询可选课程
5. 查询已选课程
6. 选课
7. 退课
8. 退出系统

关于课程和学生的信息，可以定义两个结构体，分别存储两个结构的信息。

关于一些细节，也要考虑到位，如：

1. 选课后该课程已选人数+1
2. 删课后选择该课程的学生的信息需要改变
3. 学生选择的课程的总学分需要有上限
4. 选课不能重复，也不能选择人数已满的课
5. 添加课程不能重复
6. 退课时该课程的已选人数-1，该生的各项数据更新
7. 进入不同的功能最好时覆盖式输入
8. 功能结束后pause 一下

等等

根据以上分析，需首先设计一个进入系统的菜单页面，该页面应当显示各种功能，根据分析中的且七种功能建立至少七个函数，一个功能可对应一个函数。每个函数里应当包含各种细节，也要考虑到输入错误时要及时报错等等，设计要具有实用性、合理性。

### 代码

#include <iostream>

#include <fstream>

#include <cstring>

#include <cstdlib>

#define MAXLES 1000//最多课程数

#define MAXSTU 10000//学生学号不超过五位数

#define MAXCRE 30//最多可选学分

using namespace std;

bool st[MAXLES];//存储每门课是否存在，true为存在，false为不存在

bool st\_stu[MAXSTU];//存储某位学生是否已经选课

struct Lesson {

    int num;//编号

    char subject[10];//课程名

    int credits;//学分

    char teacher[10];//任课老师

    int max\_stu;//班级最大学生数

    int now\_stu;//目前已选人数

    //bool flag;//选课状态

}les[100];

struct student {

    int number;//学号--默认四位数

    char name[10];//学生姓名

    bool chosed\_lesson[MAXLES];//已选课程

    int sum;//已选课程数

    int credits;//已选总学分

}stu[100000];

int menu() //菜单

{

    char c;

    do {

        system("cls");

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

        cout << "----------------欢迎使用学生课程管理系统---------------" << endl;

        cout << "    \*          【1】添加课程信息                  \*    " << endl;

        cout << "    \*          【2】显示课程信息                  \*    " << endl;

        cout << "    \*          【3】删除课程信息                  \*    " << endl;

        cout << "    \*          【4】查询可选课程                  \*    " << endl;

        cout << "    \*          【5】查询已选课程                  \*    " << endl;

        cout << "    \*          【6】学生选课系统                  \*    " << endl;

        cout << "    \*          【7】学生退课系统                  \*    " << endl;

        cout << "    \*          【0】退出选课系统                  \*    " << endl;

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

        cout << "请选择您的操作 (0-7):" << endl;

        c = getchar();

    } while (c < '0' || c > '7');

    return (c - '0');

}

void write()//写入文件操作

{

    ofstream CreatFile;

    CreatFile.open("lesson.txt", ios::out | ios::binary);

    if (!CreatFile) {

        cout << "文件lesson无法打开！" << endl;

        abort();

    }

    CreatFile << "课程编号\t" << "课程名称\t" << "学分\t" << "任课老师\t" << "最多学生数\t" << "已选学生数\t" << endl;

    for (int i = 1; i <= MAXLES; i++)

    {

        if (st[i] == false)continue;

        CreatFile << les[i].num << "\t\t\t"

            << les[i].subject << "\t\t"

            << les[i].credits << "\t\t"

            << les[i].teacher << "\t\t"

            << les[i].max\_stu << "\t\t\t"

            << les[i].now\_stu << endl;

    }

    CreatFile.close();

}

void add\_les(int\* a)//添加课程信息

{

    system("cls");

    ++\* a;

    int n;

    cout << "=========>>    请输入课程信息    <<=========" << endl;

    cout << "-----------    输入0可以退出添加   ------------" << endl;

    cout << "课程编号 ： "; cin >> n;//课程编号不超过1000

    loop:

    while (n <= 0 || n >= 1000)

    {

        if (n == 0)return;//为用户提供可随时退出的功能

        cout << "输入格式有误!请重新输入(输入0可退出添加): ";

        cin >> n;

    }

    if (st[n] == true)

    {

        cout << "该编号已存在，请重新定义编号！";

        cin >> n;

        goto loop;//跳回loop重新运行

    }

    les[n].now\_stu = 0;//已选人数初始为0

    les[n].num = n;//输入课程编号

    st[n] = true;   //将该课程状态定义为存在

    cout << "课程名称：";  cin >> les[n].subject;

    cout << "学分：";      cin >> les[n].credits;

    cout << "任课老师：";   cin >> les[n].teacher;

    cout << "班级最大学生数："; cin >> les[n].max\_stu;

    cout << "=======> 是否要继续添加 (y/n)";

    char c; cin >> c;

    if (c == 'y')add\_les(a);

    else write();

}

void show()//显示课程信息

{

    system("cls");

    cout << endl << "=========>>    查找课程信息    <<=========" << endl;

    cout << "输入0可退出查询" << endl;

    int u;

    cout << "请输入课程编号："; cin >> u;

    if (u == 0)return;

    while (st[u] == false)

    {

        cout << "没有此课程请重新输入：";

        cin >> u;

    }

    cout << "课程编号：" << les[u].num << endl;

    cout << "课程名称：" << les[u].subject << endl;

    cout << "学分：" << les[u].credits << endl;

    cout << "最大学生数：" << les[u].max\_stu << endl;

    cout << "已选学生数：" << les[u].now\_stu << endl;

    cout << "是否继续查找---(y/n)：";

    char c;

    cin >> c;

    if (c == 'y')show();

    else return;

}

void init(int num)//每次删课后都要重置一下选择该课的学生的状态

{

    for (int i = 1000; i < 10000; i++)

    {

        //若某位学生选了被删的课，则其状态需要改变

        if (st\_stu[i] == true && stu[i].chosed\_lesson[num] == true)

        {

            stu[i].chosed\_lesson[num] = false;

            stu[i].sum--;

            stu[i].credits -= les[num].credits;

            if (stu[i].sum == 0)st\_stu[i] = false;//只选了被删了的课的学生

        }

    }

}

void remove()//删除课程信息

{

    system("cls");

    cout << endl << "======>>    删除课程信息，请慎重！    <<======" << endl;

    cout << "输入0可返回" << endl;

    cout << "请输入需要删除课程的编号：";

    int n;

    cin >> n;

    if (n == 0)return;

    if (st[n] == false)//输入有误，重新输入

    {

        if (n == 0)return;

        cout << "对不起，没有该课程，请重新输入！";

        system("pause");

        return;

    }

    cout << "该课程信息为: " << endl;

    cout << "课程编号: " << les[n].num << endl;

    cout << "课程名称: " << les[n].subject << endl;

    cout << "学分" << les[n].credits << endl;

    cout << "任课教师: " << les[n].teacher << endl;

    cout << "最大学生数: " << les[n].max\_stu << endl;

    cout << "已选学生数" << les[n].now\_stu << endl;

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

    cout << "请再次确认是否删除---(y/n): " << endl;

    char c;

    cin >> c;

    if (c == 'y')

    {

        init(n);//重置选择该课程的学生的状态

        st[n] = false;//将该课程的状态改变为不存在

        cout << "====================>   删除成功！   <====================" << endl;

    }

    system("pause");

}

void search\_les()//查询可选课程

{

    system("cls");

    cout << endl << "==========================>>    目前可选课程    <<==========================" << endl;

    cout << "课程编号\t" << "课程名称\t" << "学分\t" << "任课老师\t" << "最大学生数\t" << "已选学生数\t" << endl;

    for (int i = 0; i <= MAXLES; i++)

    {

        if (st[i] == true && les[i].max\_stu > les[i].now\_stu)//打印可选课程信息

        {//按照表格格式化

            printf("%-15d ", les[i].num);

            printf("%-15s ", les[i].subject);

            printf("%-7d ", les[i].credits);

            printf("%-15s ", les[i].teacher);

            printf("%-15d ", les[i].max\_stu);

            printf("%d \n", les[i].now\_stu);

        }

        else continue;

    }

    cout << endl << endl;

    system("pause");

}

void search\_stu()//查询学生已选课程

{

    system("cls");

    cout << endl << "======>>    查询学生已选课程    <<======" << endl;

    int number;

    cout << "请输入需要查询的学生的学号: "; cin >> number;

    if (st\_stu[number] == false)

    {

        cout << "未查询到该学生信息，请重新输入：";

        system("pause");

        return;

    }

    cout << endl << "该学生选课信息如下：" << endl;

    cout << "已选课数" << stu[number].sum << endl;

    cout << "已选课程总学分" << stu[number].credits << endl;

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

    int num = 1;

    for (int i = 0; i < MAXLES; i++)

    {

        if (stu[number].chosed\_lesson[i] == false || st[i] == false)continue;

        cout << "第" << num++ << "门课: " << endl;

        cout << "课程编号: " << i << endl;

        cout << "课程名称: " << les[i].subject << endl;

        cout << "学分: " << les[i].credits << endl;

        cout << "任课老师: " << les[i].teacher << endl;

        cout << "最大学生数: " << les[i].max\_stu << endl;

        cout << "已选学生数: " << les[i].now\_stu << endl;

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

        if (stu[number].sum < num)break;

    }

    system("pause");

}

void stu\_choose(int flag)//选课系统

{

    if (flag == 0)

    {

        system("cls");//第一次选择该功能时 覆盖输入

        cout << endl << "======>>    欢迎使用学生选课系统    <<======" << endl;

    }

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

    cout << "学号：";

    int n;//学号

    cin >> n;

    while (n < 1000 || n >9999)

    {

        cout << "输入格式有误，请重新输入 : ";

        cin >> n;

    }

    if (st\_stu[n] == true)

    {//已经有了该生的信息，直接打印姓名

        cout << "学生姓名: " << stu[n].name << endl;

    }

    else

    {

        //第一次选课需要输入姓名

        st\_stu[n] = true;

        stu[n].number = n;//学号

        cout << "学生姓名: "; cin >> stu[n].name;

    }

    int re;//课程编号

    cout << "请输入需要选择的课程的编号 : "; cin >> re;

    if (stu[n].chosed\_lesson[re] == true) {

        cout << "该生已经选择该课程。" << endl;

        system("pause");

        return;

    }

    if (st[re] == false) {

        cout << "没有关于该课程的信息。" << endl;

        system("pause");

        return;

    }

    if (les[re].now\_stu >= les[re].max\_stu)

    {

        cout << "选择该课程人数已达上限！" << endl;

        system("pause");

        return;

    }

    if (les[re].credits > MAXCRE - stu[n].credits)

    {

        cout << "已选学分已达上限!!!" << endl;

        system("pause");

        return;

    }

    cout << endl << "该课程信息为: " << endl;

    cout << "课程编号: " << les[re].num << endl;

    cout << "课程名称: " << les[re].subject << endl;

    cout << "学分: " << les[re].credits << endl;

    cout << "最大学生数: " << les[re].max\_stu << endl;

    cout << "已选学生数: " << les[re].now\_stu << endl;

    cout << "再次确定是否选择---(y/n): ";

    char c;

    cin >> c;

    if (c == 'y')

    {

        if (stu[n].sum == 10)

        {

            cout << "对不起，您选课已达上限!";

            char a;

            cin >> a;

            return;

        }

        les[re].now\_stu++;

        stu[n].chosed\_lesson[re] = true;//该生的改课状态定义为已选

        stu[n].sum++;

        stu[n].credits += les[re].credits;

        cout << "是否继续选课---(y/n): ";

        cin >> c;

        if (c == 'y')stu\_choose(1);

        else return;

    }

    system("pause");

}

void drop()//退课

{

    system("cls");

    cout << endl << "======>>    欢迎使用学生退课系统    <<======" << endl;

    cout << "学号：";

    int number;

    cin >> number;//学号

    while (number < 1000 || number >9999)//保证学号是四位数

    {

        cout << "输入格式有误，请重新输入 : ";

        cin >> number;

    }

    while (st\_stu[number] == false)

    {

        cout << "无法查询到该生的信息,请重新输入： ";

        cin >> number;

    }

    cout << "学生姓名: " << stu[number].name << endl;

    int re;//课程编号

    cout << "请输入需要退选的的课程的编号 : "; cin >> re;

    while (stu[number].chosed\_lesson[re] == false)

    {

        cout << "该生没有选择这门课，请重新选择";

        cin >> re;

    }

    cout << endl << "该课程信息为:" << endl;

    cout << "课程编号: " << les[re].num << endl;

    cout << "课程名称: " << les[re].subject << endl;

    cout << "学分: " << les[re].credits << endl;

    cout << "任课教师: " << les[re].teacher << endl;

    cout << "最大学生数: " << les[re].max\_stu << endl;

    cout << "已选学生数: " << les[re].now\_stu << endl;

    cout << "再次确定是否退课---(y/n): ";

    char c;

    cin >> c;

    if (c == 'y')

    {

        stu[number].chosed\_lesson[re] = false;

        cout << "======>>   退课成功!   <<======" << endl;

        les[re].now\_stu--;

        stu[number].sum--;

        stu[number].credits -= les[re].credits;

    }

    system("pause");

}

int main()

{

    int count = 0;

    while (1)

        switch (menu())

        {

        case 1:

            add\_les(&count);

            break;

        case 2:

            show();

            break;

        case 3:

            remove();

            break;

        case 4:

            search\_les();

            break;

        case 5:

            search\_stu();

            break;

        case 6:

            stu\_choose(0);

            break;

        case 7:

            drop();

            break;

        case 0:

            cout << "================感谢您使用学生选课系统==============" << endl;

            write();//更新课程数据库

            exit(0);

        }

    return 0;

}

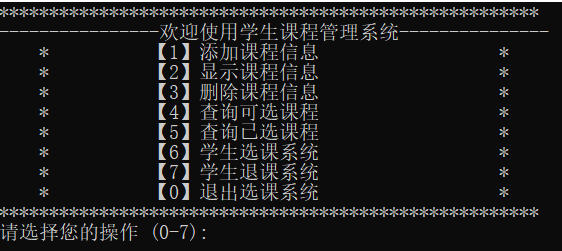
### 问题及解决方案

首先我们面临的问题有

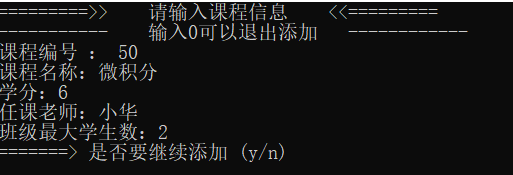
1. 课程编号使用字符串表示还是用整型数表示，在这里，我选择用整型数表示，但限制了了编号必须小于1000
2. 学生的学号也是用整型数表示，但是限制了只能有四位数且首位不能为0。
3. 用户可能手误点了某个功能想退出，所以我定义了只要输入0就可以退出该功能
4. 为了确保学生选课或者退课的决定准确，我定义了当学生选择到某课时，会弹出该课程的各项信息，并要求用户再次确定功能！

### 实验结果展示

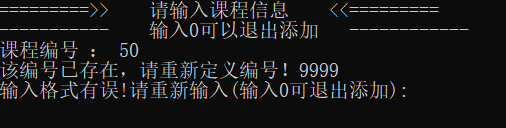
1. **菜单页面**



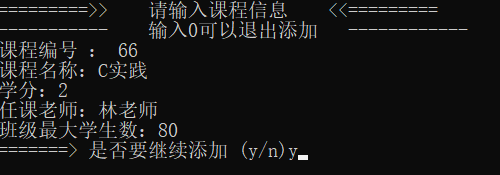
1. **添加课程**

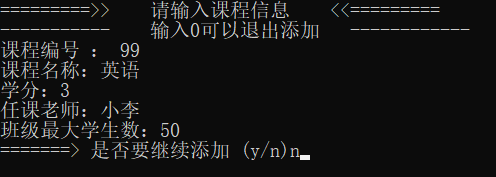


输入错误时会报错

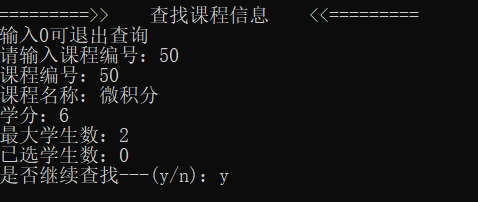


继续添加两个课程

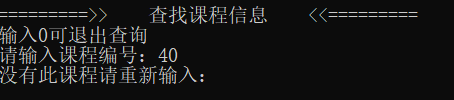




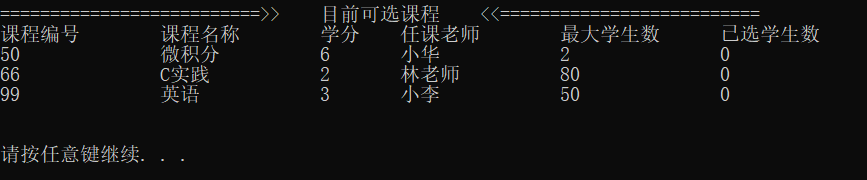
1. 查询课程信息



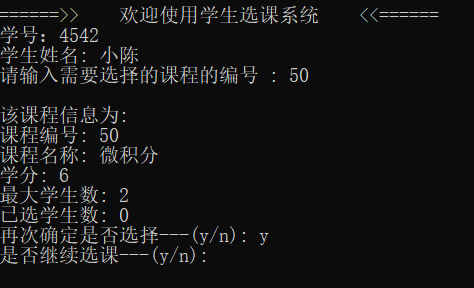
无法查询时会报错



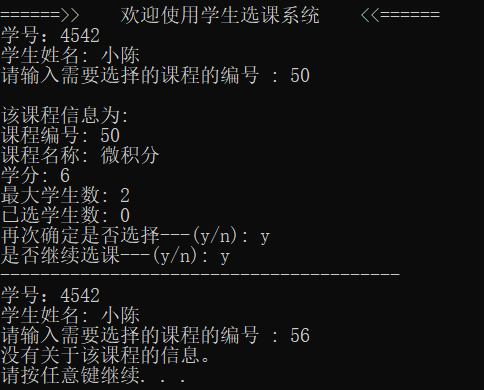
1. 查询可选课程（列表输出，一目了然）



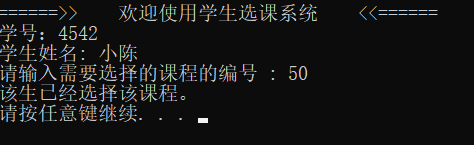
1. 选课



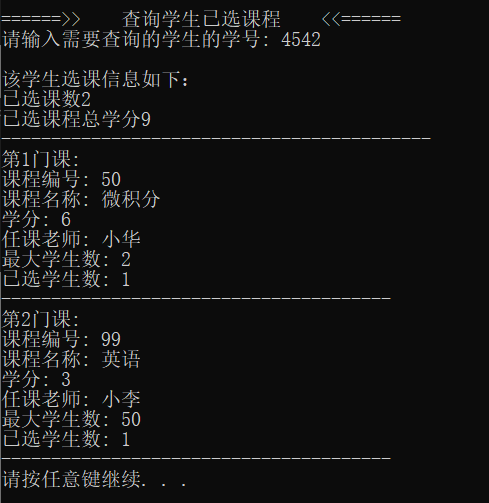
选课错误时



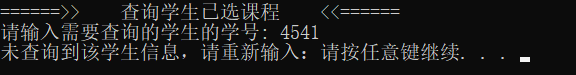
重复选课时



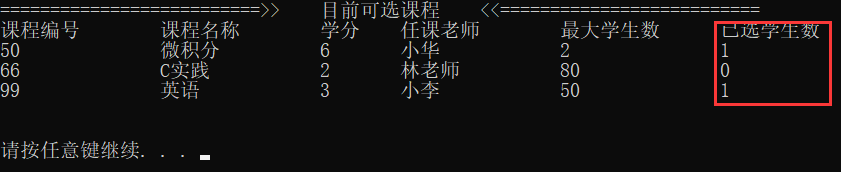
1. 查询已选课程



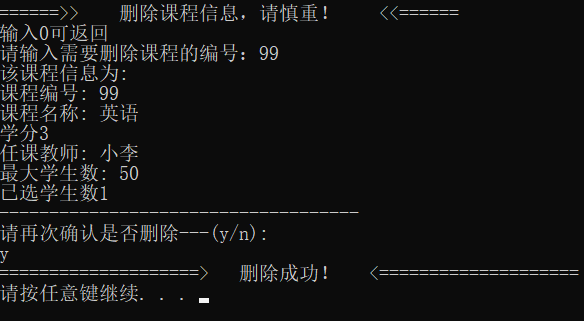
查询错误时



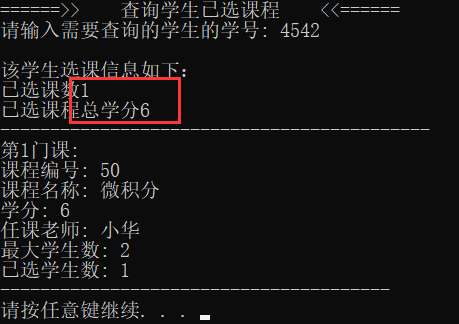
此时的可选课程页面也发生了变化



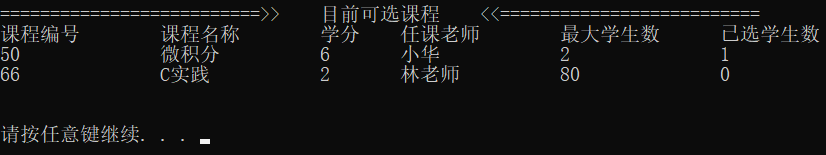
1. 删除课程



此时再查询该生的信息（实时更新）

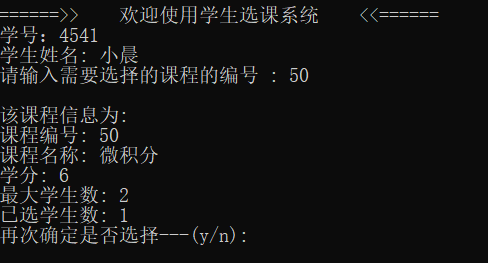


以及可选课程

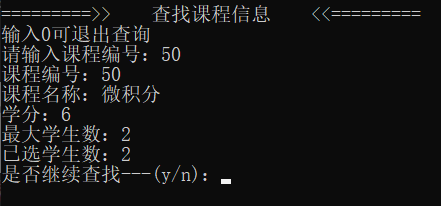


1. 最大学生数 == 已选学生数时

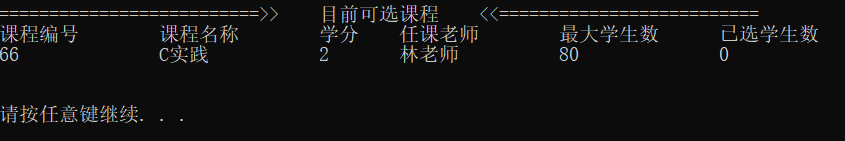
为方便，再添加一名学生



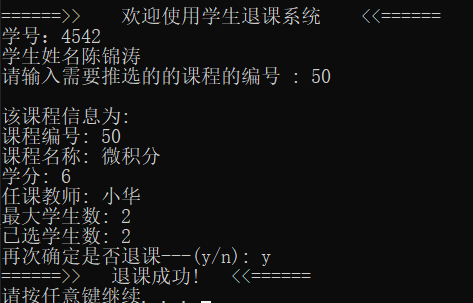
显示课程信息



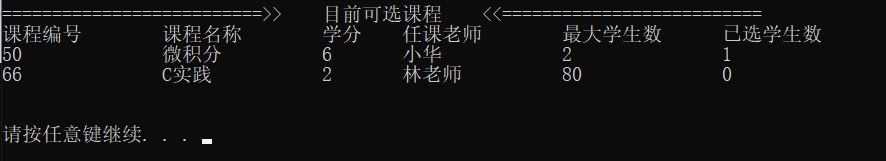
查询可选



1. 退课



再次查找可选课程信息（已完成实时更新）



### 本课程学习总结

在本课程中，关于C语言的一些疑难知识点掌握了不少，尤其是关于链表一节的内容，还有位操作增加了我们逻辑运算知识，指针的使用更是让我清晰址与成员值间的关系，以及如何运用指针使运行速度加快和多种类型成员名的指针如何运用，以前总是觉得文件是个难点，现在也行初步的完成读写的操作了，课程设计的大作业也让我懂得了如何将一个复杂的系统拆成不同的几个小功能，分别去对各功能去进行编写和优化，在实现系统功能的同时也顾及了系统的实用性及可行性。

总之，收获颇丰！