//sushe.h

#define MAX\_NUM 10000000

#include <String>

#include<ctime>

#include <fstream>

using namespace std;

struct StuNode{

long stu\_num = 0;

char stu\_nam[40];

char stu\_cla[40];

char stu\_sus[40];

char stu\_chu[40];

StuNode \*nextstu;

};

class Sushe{

StuNode \*StuListHead;

public:

Sushe(); //构造函数

~Sushe(); //析构函数

void CreatSinfo(); //创建学生信息

void StuInsert(int num, char\* nam, char\* cla, char\* sus, char\* chu); //插入学生信息

void StuDelete(int num); //删除学生信息

StuNode \*StuFind(int num); //查找学生信息，传入参数学号

void StuModify(int num, char\* nam, char\* cla, char\* sus, char\* chu); //修改学生信息

void StuRead(); //从文件读入学生信息

void StuSave(); //保存学生信息到文件

void StuQuit();

void ShowInfo(); //遍历输出学生信息

void panduan(int stu\_num);

};

void pdwei(int stu\_num)

{

while (stu\_num<10000 || stu\_num>99999)

{

cout << "你输入的学号不正确，请输入一个五位数的学号" << endl;

cout << "学号:";

cin >> stu\_num;

}

}

void Sushe::panduan(int stu\_num)

{

pdwei(stu\_num);

StuNode \*p;

for (p = StuListHead->nextstu; p != NULL; p = p->nextstu)

{

while (p->stu\_num == stu\_num)

{

cout << "学号重复，请重新输入！" << endl;

cout << "学号：";

cin >> stu\_num;

}

}

}

void SuMenu()

{

time\_t t;

time(&t);

cout << "---------------------------------宿舍基本信息模块-------------------------------" << endl;

cout << "\t\t\t 本地时间:" << ctime(&t);

cout << "--------------------------------------------------------------------------------\n" << endl;

cout << "\t\t 1.读入学生宿舍信息 " << endl;

cout << "\t\t 2.录入多个宿舍信息 " << endl;

cout << "\t\t 3.添加单个宿舍信息 " << endl;

cout << "\t\t 4.删除已有宿舍信息 " << endl;

cout << "\t\t 5.查找已有宿舍信息 " << endl;

cout << "\t\t 6.修改已有宿舍信息 " << endl;

cout << "\t\t 9.输出所有宿舍信息 " << endl;

cout << "\t\t 10.保存现有宿舍信息 " << endl;

cout << "\t\t 0.返回信息主菜单 " << endl;

cout << "\n\t\t请选择：";

}

Sushe::Sushe() //构造函数

{

StuListHead = new StuNode;

StuListHead->nextstu = NULL;

StuListHead->stu\_num = 0;

}

Sushe::~Sushe() //析构函数

{

StuNode \*p;

while (StuListHead)

{

p = StuListHead;

StuListHead = StuListHead->nextstu;

delete p;

}

StuListHead = NULL;

}

void Sushe::CreatSinfo() //创建学生信息表

{

int n;

StuNode \*p, \*s;

p = StuListHead;

cout << "请输入学生宿舍人数：";

cin >> n;

for (int i = 1; i <= n; i++)

{

s = new StuNode;

cout << "学号："; cin >> s->stu\_num;

panduan(s->stu\_num);

cout << "姓名："; cin >> s->stu\_nam;

cout << "班级："; cin >> s->stu\_cla;

cout << "宿舍："; cin >> s->stu\_sus;

cout << "床号："; cin >> s->stu\_chu;

s->nextstu = p->nextstu;

p->nextstu = s;

p = p->nextstu;

}

if (p == NULL) //判断学生信息表是否创建成功

{

cout << "创建失败请重新创建！" << endl;

CreatSinfo();

}

}

void Sushe::ShowInfo() //遍历输出

{

StuNode \*p;

cout << "学号" << '\t' << "姓名" << '\t' << "班级" << '\t' << "宿舍" << '\t' << "床号" << endl;

for (p = StuListHead->nextstu; p != NULL; p = p->nextstu)

{

cout << p->stu\_num << '\t' << p->stu\_nam << '\t' << p->stu\_cla << '\t' << p->stu\_sus << '\t' << p->stu\_chu << endl;

}

}

void Sushe::StuInsert(int num, char\* nam, char\* cla, char\* sus, char\* chu) //插入学生信息（头插法）

{

StuNode \*s, \*p;

s = new StuNode;

s->stu\_num = num;

strcpy(s->stu\_nam, nam);

strcpy(s->stu\_cla, cla);

strcpy(s->stu\_sus, sus);

strcpy(s->stu\_chu, chu);

p = StuListHead;

s->nextstu = p->nextstu;

p->nextstu = s;

}

void Sushe::StuDelete(int num)

{

StuNode \*p, \*ptemp;

p = StuListHead;

ptemp = p;

while (p->nextstu && p->stu\_num != num) //循环终止条件为p->nextstu不为空 而且没有找到相应学号的学生

{

ptemp = p;

p = p->nextstu;

}

if (p->stu\_num == num)

{

ptemp->nextstu = p->nextstu;

delete p;

}

else

{

cout << "未找到该学生信息！" << endl;

}

}

StuNode \*Sushe::StuFind(int num)

{

StuNode \*p;

p = StuListHead->nextstu;

while (p->nextstu && p->stu\_num != num) //循环终止条件为p->nextstu不为空 而且没有找到相应学号的学生

{

p = p->nextstu;

}

if (p->stu\_num == num)

{

return p;

}

else

{

cout << "未找到该学生信息！" << endl;

return NULL;

}

}

void Sushe::StuModify(int num, char\* nam, char\* cla, char\* sus, char\* chu) //修改信息

{

StuNode \*ItemStu = StuFind(num); //直接调用查找函数

if (ItemStu != NULL)

{

ItemStu->stu\_num = num;

strcpy(ItemStu->stu\_nam, nam);

strcpy(ItemStu->stu\_cla, cla);

strcpy(ItemStu->stu\_sus, sus);

strcpy(ItemStu->stu\_chu, chu);

cout << "修改成功！" << endl;

}

}

void Sushe::StuRead() //从文件读入数据

{

StuNode \*p;

p = StuListHead;

ifstream in("sushea.txt");

if (!in) { cout << "文件sushea.txt中没有学生信息，请先录入学生信息!" << endl; return; }

while (!in.eof())

{

int num; char nam[40]; char cla[40]; char sus[40]; char chu[40];

in >> num >> nam >> cla >> sus >> chu;

StuInsert(num, nam, cla, sus, chu);

}

}

void Sushe::StuSave() //保存学生信息

{

StuNode \*p;

p = StuListHead->nextstu;

ofstream out("sushea.txt");

if (!out) { cout << "不能打开文件sushea.txt！" << endl; return; }

while (p != NULL)

{

out << p->stu\_num << '\t' << p->stu\_nam << '\t' << p->stu\_cla << '\t' << p->stu\_sus << '\t' << p->stu\_chu << '\n';

p = p->nextstu;

}

}

void Sushe::StuQuit() //学生信息写入文件

{

char choice;

cout << "是否保存学生信息：？（Y/N）";

cin >> choice;

if (choice == 'y' || choice == 'Y')

{

StuSave();

cout << "学生信息已保存..." << endl;

}

}

void sushe()

{

system("mode con:cols=200 lines=1000");

int pnum;

char pnam[40]; char pcla[40]; char psus[40]; char pchu[40];

StuNode \*pfind;

Sushe stu;

while (1)

{

system("pause");

system("cls"); //清屏

SuMenu();

int x;

cin >> x;

if (x == 0){

return;

}

switch (x)

{

case 1:

stu.StuRead();

cout << "读入学生宿舍信息表：" << endl;

stu.ShowInfo();

break;

case 2:

stu.CreatSinfo();

cout << "请核对输入学生信息！" << endl;

stu.ShowInfo();

break;

case 3: //添加信息

cout << "请输入添加学生宿舍信息：" << endl;

cout << "学号："; cin >> pnum;

stu.panduan(pnum);

cout << "姓名："; cin >> pnam;

cout << "班级："; cin >> pcla;

cout << "宿舍："; cin >> psus;

cout << "床号："; cin >> pchu;

stu.StuInsert(pnum, pnam, pcla, psus, pchu);

cout << "更新学生宿舍信息表..." << endl;

stu.ShowInfo();

break;

case 4:

cout << "请输入要删除学生学号：";

cin >> pnum;

stu.StuDelete(pnum);

cout << "更新学生宿舍信息表..." << endl;

stu.ShowInfo();

break;

case 5:

cout << "请输入要查找学生学号：";

cin >> pnum;

pfind = stu.StuFind(pnum);

cout << "查找学生学号：" << pfind->stu\_num << " 姓名： " << pfind->stu\_nam << " 班级： " << pfind->stu\_cla << " 宿舍： " << pfind->stu\_sus << " 床号： " << pfind->stu\_chu << endl;

break;

case 6:

cout << "请输入要修改学生学号：";

cin >> pnum;

cout << "请重新输入学生宿舍信息!" << endl;

cout << "姓名："; cin >> pnam;

cout << "班级："; cin >> pcla;

cout << "宿舍："; cin >> psus;

cout << "床号："; cin >> pchu;

stu.StuModify(pnum, pnam, pcla, psus, pchu);

cout << "更新学生宿舍信息表..." << endl;

stu.ShowInfo();

break;

case 9:

stu.ShowInfo();

break;

case 10:

stu.StuQuit();

break;

}

}

}

//Ruzhu.h

#define MAX\_NUM 10000000

#include <String>

#include<ctime>

#include <fstream>

using namespace std;

struct RuNode{

long ru\_num = 0;

char ru\_nam[40];

char ru\_cla[40];

char ru\_mon[40];

char ru\_check\_in[40];

char ru\_check\_out[40];

RuNode \*nextru;

};

class Ruzhu{

RuNode \*RuListHead;

public:

Ruzhu(); //构造函数

~Ruzhu(); //析构函数

void RuCreat(); //创建入住信息

void RuInsert(int num, char\* nam, char\* cla, char\* mon, char\* check\_in, char\* check\_out); //插入入住信息

RuNode \*RuFind(int num); //查找入住信息，传入参数学号

void RuRead(); //从文件读入入住信息

void RuSave(); //保存学生入住信息到文件

void RuQuit();

void RuShow(); //遍历输出学生入住信息

void panduan(int ru\_num);

};

void pdweiru(int ru\_num)

{

while (ru\_num<10000 || ru\_num>99999)

{

cout << "你输入的学号不正确，请输入一个五位数的学号" << endl;

cout << "学号:";

cin >> ru\_num;

}

}

void Ruzhu::panduan(int ru\_num)

{

pdweiru(ru\_num);

RuNode \*p;

for (p = RuListHead->nextru; p != NULL; p = p->nextru)

{

while (p->ru\_num == ru\_num)

{

cout << "学号重复，请重新输入！" << endl;

cout << "学号：";

cin >> ru\_num;

}

}

}

void RuMenu()

{

time\_t t;

time(&t);

cout << "---------------------------------学生入住信息模块-------------------------------" << endl;

cout << "\t\t\t 本地时间:" << ctime(&t);

cout << "--------------------------------------------------------------------------------\n" << endl;

cout << "\t\t 1.读入学生入住信息 " << endl;

cout << "\t\t 2.录入入住信息表格 " << endl;

cout << "\t\t 3.添加新的入住信息 " << endl;

cout << "\t\t 4.查找已有学生信息 " << endl;

cout << "\t\t 5.输出所有学生信息 " << endl;

cout << "\t\t 6.保存现有学生信息 " << endl;

cout << "\t\t 0.返回主菜单 " << endl;

cout << "\n\t\n\t\t请选择：";

}

Ruzhu::Ruzhu() //构造函数

{

RuListHead = new RuNode;

RuListHead->nextru = NULL;

RuListHead->ru\_num = 0;

}

Ruzhu::~Ruzhu() //析构函数

{

RuNode \*p;

while (RuListHead)

{

p = RuListHead;

RuListHead = RuListHead->nextru;

delete p;

}

RuListHead = NULL;

}

void Ruzhu::RuRead() //从文件读入数据

{

RuNode \*p;

p = RuListHead;

ifstream in("ruzhua.txt");

if (!in) { cout << "文件ruzhua.txt中没有入住信息，请先录入入住信息!" << endl; return; }

while (!in.eof())

{

int num; char nam[40]; char cla[40]; char mon[40]; char check\_in[40]; char check\_out[40];

in >> num >> nam >> cla >> mon >> check\_in >> check\_out;

RuInsert(num, nam, cla, mon, check\_in, check\_out);

}

}

void Ruzhu::RuCreat() //创建学生信息表

{

int n;

RuNode \*p, \*s;

p = RuListHead;

cout << "请输入学生宿舍入住人数：";

cin >> n;

for (int i = 1; i <= n; i++)

{

s = new RuNode;

cout << "学号："; cin >> s->ru\_num;

panduan(s->ru\_num);

cout << "姓名："; cin >> s->ru\_nam;

cout << "班级："; cin >> s->ru\_cla;

cout << "缴费金额："; cin >> s->ru\_mon;

cout << "入住时间："; cin >> s->ru\_check\_in;

cout << "退房时间："; cin >> s->ru\_check\_out;

s->nextru = p->nextru;

p->nextru = s;

p = p->nextru;

}

if (p == NULL) //判断学生信息表是否创建成功

{

cout << "创建失败请重新创建！" << endl;

RuCreat();

}

}

void Ruzhu::RuInsert(int num, char\* nam, char\* cla, char\* mon, char\* check\_in, char\* check\_out) //插入学生入住信息（头插法）

{

RuNode \*s, \*p;

s = new RuNode;

s->ru\_num = num;

strcpy(s->ru\_nam, nam);

strcpy(s->ru\_cla, cla);

strcpy(s->ru\_mon, mon);

strcpy(s->ru\_check\_in, check\_in);

strcpy(s->ru\_check\_out, check\_out);

p = RuListHead;

s->nextru = p->nextru;

p->nextru = s;

}

void Ruzhu::RuShow() //遍历输出

{

RuNode \*p;

cout << "学号" << '\t' << "姓名" << '\t' << "班级" << '\t' << "缴费金额" << '\t' << "入住时间" << '\t' << "退房时间" << endl;

for (p = RuListHead->nextru; p != NULL; p = p->nextru)

{

cout << p->ru\_num << '\t' << p->ru\_nam << '\t' << p->ru\_cla << '\t' << p->ru\_mon << '\t' << p->ru\_check\_in << '\t' << p->ru\_check\_out << endl;

}

}

RuNode \*Ruzhu::RuFind(int num)

{

RuNode \*p;

p = RuListHead->nextru;

while (p->nextru && p->ru\_num != num) //循环终止条件为p->nextru不为空 而且没有找到相应学号的学生

{

p = p->nextru;

}

if (p->ru\_num == num)

{

return p;

}

else

{

cout << "未找到该学生信息！" << endl;

return NULL;

}

}

void Ruzhu::RuSave() //保存学生信息

{

RuNode \*p;

p = RuListHead->nextru;

ofstream out("ruzhua.txt");

if (!out) { cout << "不能打开文件ruzhua.txt！" << endl; return; }

while (p != NULL)

{

out << p->ru\_num << '\t' << p->ru\_nam << '\t' << p->ru\_cla << '\t' << p->ru\_mon << '\t' << p->ru\_check\_in << '\t' << p->ru\_check\_out << '\n';

p = p->nextru;

}

}

void Ruzhu::RuQuit() //学生信息写入文件

{

char choice;

cout << "是否保存学生入住信息：？（Y/N）";

cin >> choice;

if (choice == 'y' || choice == 'Y')

{

RuSave();

cout << "学生入住信息已保存..." << endl;

}

}

void ruzhu()

{

int pnum;

char pnam[40]; char pcla[40]; char pmon[40]; char pcheck\_in[40]; char pcheck\_out[40];

RuNode \*pfind;

Ruzhu ru;

while (1)

{

system("pause");

system("cls"); //清屏

RuMenu();

int x;

cin >> x;

if (x == 0){

return;

}

switch (x)

{

case 1:

ru.RuRead();

cout << "读入学入住信息表：" << endl;

ru.RuShow();

break;

case 2:

ru.RuCreat();

cout << "请核对输入学生信息！" << endl;

ru.RuShow();

break;

case 3: //添加信息

cout << "请输入添加学生入住信息：" << endl;

cout << "学号："; cin >> pnum;

ru.panduan(pnum);

cout << "姓名："; cin >> pnam;

cout << "班级："; cin >> pcla;

cout << "缴费金额："; cin >> pmon;

cout << "入住时间："; cin >> pcheck\_in;

cout << "退房时间："; cin >> pcheck\_out;

ru.RuInsert(pnum, pnam, pcla, pmon, pcheck\_in, pcheck\_out);

cout << "更新学生入住信息表..." << endl;

ru.RuShow();

break;

case 4:

cout << "请输入要查找学生学号：";

cin >> pnum;

pfind = ru.RuFind(pnum);

cout << "查找学生学号：" << pfind->ru\_num << " 姓名： " << pfind->ru\_nam << " 班级： " << pfind->ru\_cla << " 缴费金额： " << pfind->ru\_mon << " 入住时间： " << pfind->ru\_check\_in << "退房时间：" << pfind->ru\_check\_out << endl;

break;

case 5:

ru.RuShow();

break;

case 6:

ru.RuQuit();

break;

}

}

}

//Fuzhu.h

#define MAX\_NUM 10000000

#include <String>

#include<ctime>

#include<windows.h>

#include<cstdlib>

#include<iostream>

#include<cstdlib>

#define random(a,b) (rand()%(b-a+1)+a)

using namespace std;

void suijishu(int a, int b)

{

srand((unsigned)time(NULL));

for (int i = 0; i<10; i++)

cout << random(a, b) << " ";

}

/\*void pdwei(int ru\_num)

{

while (ru\_num<10000 || ru\_num>99999)

{

cout << "你输入的学号不正确，请输入一个五位数的学号" << endl;

cout << "学号:";

cin >> ru\_num;

}

}\*/

int Systemdoor()

{

string username = "hello", password = "nihao";

string name, temp;

int number = 3;

while (1)

{

time\_t t;

time(&t);

cout << "---------------------------------欢迎使用学生宿舍管理信息系统-------------------------------" << endl;

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

cout << "\t\t 本地时间:" << ctime(&t);

cout << " 用 户 名：";

cin >> name;

cout << " 密 码：";

cin >> temp;

cout << endl;

if (name != username || temp != password)

{

number--;

if (number >0)

{

cout << " 用户名/密码错误!你还有" << number << "次机会" << endl;

}

else

cout << "用户名/密码错误!" << endl, exit(0);

}

else

{

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*密码正确\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl << endl;

for (int i = 0; i <3; i++)

{

Sleep(1000); /\* windows 使用Sleep，参数为毫秒 \*/

}

system("cls"); //清屏

return 1;

}

}

}

//Weisheng.h

#define MAX\_NUM 10000000

#include <String>

#include<ctime>

#include <fstream>

using namespace std;

struct WeiNode{

char wei\_score[40];

int wei\_n = 0;

char wei\_sus[40];

WeiNode \*nextwei;

};

class Weisheng{

WeiNode \*WeiListHead;

public:

Weisheng(); //构造函数

~Weisheng(); //析构函数

void WeiCreat(); //创建信息

void WeiInsert(int n, char\* sus, char\* score); //插入入住信息

WeiNode \*WeiFind(char\* sus); //查找入住信息，传入参数宿舍号

void WeiSort(char ch); //根据 总分排序

void WeiCopy(WeiNode \*ptemp, WeiNode \*p);

void WeiRead(); //从文件读入信息

void WeiSave(); //保存学信息到文件

void WeiQuit();

void WeiShow(); //遍历输出信息

void panduan(int ru\_num);

void Weiin();

};

void WeiMenu()

{

time\_t t;

time(&t);

cout << "---------------------------------学生宿舍卫生评比模块-------------------------------" << endl;

cout << "\t\t\t 本地时间:" << ctime(&t);

cout << "--------------------------------------------------------------------------------\n" << endl;

cout << "\t\t 1.读入宿舍卫生信息 " << endl;

cout << "\t\t 2.录入宿舍卫生表格 " << endl;

cout << "\t\t 3.查找宿舍卫生信息 " << endl;

cout << "\t\t 4.输出所有宿舍信息 " << endl;

cout << "\t\t 5.保存现有宿舍信息 " << endl;

cout << "\t\t 0.返回主菜单 " << endl;

cout << "\n\t\t请选择：";

}

Weisheng::Weisheng() //构造函数

{

WeiListHead = new WeiNode;

WeiListHead->nextwei = NULL;

}

Weisheng::~Weisheng() //析构函数

{

WeiNode \*p;

while (WeiListHead)

{

p = WeiListHead;

WeiListHead = WeiListHead->nextwei;

delete p;

}

WeiListHead = NULL;

}

void Weisheng::WeiInsert(int n, char\* sus, char\* score) //插入入住信息（头插法）

{

WeiNode \*s, \*p;

s = new WeiNode;

s->wei\_n = n;

strcpy(s->wei\_sus, sus);

strcpy(s->wei\_score, score);

p = WeiListHead;

s->nextwei = p->nextwei;

p->nextwei = s;

}

void Weisheng::WeiRead() //从文件读入数据

{

WeiNode \*p;

p = WeiListHead;

ifstream in("weishenga.txt");

if (!in) { cout << "文件weishenga.txt中没有宿舍信息，请先录入宿舍信息!" << endl; return; }

while (!in.eof())

{

int n; char sus[40]; char score[40];

in >> n >> sus >> score;

WeiInsert(n, sus, score);

}

}

void Weisheng::WeiCreat() //创建宿舍卫生信息表

{

int n;

WeiNode \*p, \*s;

p = WeiListHead;

cout << "请输入学生宿舍卫生条数：";

cin >> n;

for (int i = 1; i <= n; i++)

{

s = new WeiNode;

cout << "宿舍号："; cin >> s->wei\_sus;

//panduan(s->wei\_num);

cout << "卫生评分："; cin >> s->wei\_score;

cout << "排名："; cin >> s->wei\_n;

s->nextwei = p->nextwei;

p->nextwei = s;

p = p->nextwei;

}

if (p == NULL) //判断信息表是否创建成功

{

cout << "创建失败请重新创建！" << endl;

WeiCreat();

}

}

void Weisheng::WeiShow() //遍历输出

{

WeiNode \*p;

cout << "排名" << '\t' << "宿舍" << '\t' << "评分" << endl;

for (p = WeiListHead->nextwei; p != NULL; p = p->nextwei)

{

cout << p->wei\_n << '\t' << p->wei\_sus << '\t' << p->wei\_score << endl;

}

}

void Weisheng::Weiin()

{

WeiNode \*p;

int i = 1;

for (p = WeiListHead->nextwei; p != NULL; p = p->nextwei)

{

p->wei\_n = i;

i++;

}

}

WeiNode \*Weisheng::WeiFind(char\* sus)

{

WeiNode \*p;

p = WeiListHead->nextwei;

while (p->nextwei && strcmp(p->wei\_sus, sus) != 0) //循环终止条件为p->nextru不为空 而且没有找到相应宿舍

{

p = p->nextwei;

}

if (strcmp(p->wei\_sus, sus) == 0)

{

return p;

}

else

{

cout << "未找到该宿舍信息！" << endl;

return NULL;

}

}

void Weisheng::WeiCopy(WeiNode \*ptemp, WeiNode \*p) //拷贝信息(将p的信息拷贝到ptemp中)

{

if (p == NULL)

{

cout << "拷贝目标为空！" << endl;

}

else

{

ptemp->wei\_n = p->wei\_n;

strcpy(ptemp->wei\_sus, p->wei\_sus);

strcpy(ptemp->wei\_score, p->wei\_score);

//ptemp->nextwei = p->nextwei; //只是信息拷贝,next不能拷贝否则信息丢失

}

}

void Weisheng::WeiSort(char ch) //根据 总分排序

{

if (ch == '>')

{

for (WeiNode \*p = WeiListHead->nextwei; p != NULL; p = p->nextwei)

{

for (WeiNode \*q = WeiListHead->nextwei; q != NULL; q = q->nextwei)

{

if (strcmp(p->wei\_score, q->wei\_score)>0)

{

WeiNode \*ptemp = new WeiNode;

WeiCopy(ptemp, p);

WeiCopy(p, q);

WeiCopy(q, ptemp);

}

}

}

}

else if (ch == '<')

{

for (WeiNode \*p = WeiListHead->nextwei; p != NULL; p = p->nextwei)

{

for (WeiNode \*q = WeiListHead->nextwei; q != NULL; q = q->nextwei)

{

if (strcmp(p->wei\_score, q->wei\_score)<0)

{

WeiNode \*ptemp = new WeiNode;

WeiCopy(ptemp, p);

WeiCopy(p, q);

WeiCopy(q, ptemp);

}

}

}

}

else

{

cout << "排序条件出错！" << endl;

}

}

void Weisheng::WeiSave() //保存信息

{

WeiNode \*p;

p = WeiListHead->nextwei;

ofstream out("weishenga.txt");

if (!out) { cout << "不能打开文件weishenga.txt！" << endl; return; }

while (p != NULL)

{

out << p->wei\_n << '\t' << p->wei\_sus << '\t' << p->wei\_score << '\n';

p = p->nextwei;

}

}

void Weisheng::WeiQuit() //信息写入文件

{

char choice;

cout << "是否保存宿舍卫生信息：？（Y/N）";

cin >> choice;

if (choice == 'y' || choice == 'Y')

{

WeiSave();

cout << "宿舍卫生信息已保存..." << endl;

}

}

void weisheng()

{

char psus[40];

WeiNode \*pfind;

Weisheng wei;

while (1)

{

system("pause");

system("cls"); //清屏

WeiMenu();

int x;

cin >> x;

if (x == 0){

return;

}

switch (x)

{

case 1:

wei.WeiRead();

cout << "读入宿舍卫生评分信息表：" << endl;

wei.WeiShow();

break;

case 2:

wei.WeiCreat();

cout << "请核对输入评分信息！" << endl;

wei.WeiShow();

break;

case 3:

cout << "请输入要查找宿舍：";

cin >> psus;

pfind = wei.WeiFind(psus);

cout << "查找宿舍卫生情况：" << endl;

cout << "宿舍排名：" << pfind->wei\_n << " 宿舍号： " << pfind->wei\_sus << " 评分： " << pfind->wei\_score << endl;

break;

case 4:

cout << "宿舍卫生排名升序排序请按1！倒序排序请按2！" << endl;

int p;

cin >> p;

if (p == 1)

{

wei.WeiSort('<');

wei.Weiin();

wei.WeiShow();

}

else if (p == 2)

{

wei.WeiSort('>');

wei.WeiShow();

}

else

{

cout << "输入指令错误！" << endl;

}

break;

case 5:

wei.WeiQuit();

break;

}

}

}

//Shoufei.h

#define MAX\_NUM 10000000

#include <windows.h>

#include <String>

#include<ctime>

#include <fstream>

using namespace std;

struct ShouNode{

int shou\_du = 0;

int shou\_fei;

char shou\_sus[40];

ShouNode \*nextshou;

};

class Shoufei{

ShouNode \*ShouListHead;

public:

Shoufei(); //构造函数

~Shoufei(); //析构函数

void ShouCreat(); //创建宿舍收费表信息

void ShouInsert(char\* sus, int du, int fei); //插入宿舍收费信息信息

ShouNode \*ShouFind(char\* sus); //查找信息，传入参数学号

void ShouRead(); //从文件读入宿舍收费信息

void ShouSave(); //保存信息到文件

void ShouQuit();

void ShouShow(); //遍历输出宿舍收费信息

};

void ShouMenu()

{

time\_t t;

time(&t);

cout << "---------------------------------宿舍水电费模块-------------------------------" << endl;

cout << "\t\t\t 本地时间:" << ctime(&t);

cout << "--------------------------------------------------------------------------------\n" << endl;

cout << "\t\t 1.读入宿舍水电信息 " << endl;

cout << "\t\t 2.录入宿舍水电表格 " << endl;

cout << "\t\t 3.查找已有宿舍信息 " << endl;

cout << "\t\t 4.输出所有收费信息 " << endl;

cout << "\t\t 5.保存宿舍收费信息 " << endl;

cout << "\t\t 0.返回主菜单 " << endl;

cout << "\n\t\n\t\t请选择：";

}

Shoufei::Shoufei() //构造函数

{

ShouListHead = new ShouNode;

ShouListHead->nextshou = NULL;

}

Shoufei::~Shoufei() //析构函数

{

ShouNode \*p;

while (ShouListHead)

{

p = ShouListHead;

ShouListHead = ShouListHead->nextshou;

delete p;

}

ShouListHead = NULL;

}

void Shoufei::ShouRead() //从文件读入数据

{

ShouNode \*p;

p = ShouListHead;

ifstream in("shoufeia.txt");

if (!in) { cout << "文件shoufeia.txt中没有入住信息，请先录入入住信息!" << endl; return; }

while (!in.eof())

{

int du, fei; char sus[40];

in >> sus >> du >> fei;

ShouInsert(sus, du, fei);

}

}

int standard(int du){

int fei = 100;

if (du > 200)

return (100 + (du - 200) \* 3);

else

return fei;

}

void Shoufei::ShouCreat() //创建信息表

{

int n;

ShouNode \*p, \*s;

p = ShouListHead;

cout << "请输入录入宿舍条数：";

cin >> n;

for (int i = 1; i <= n; i++)

{

s = new ShouNode;

cout << "宿舍："; cin >> s->shou\_sus;

cout << "水电使用度："; cin >> s->shou\_du;

s->shou\_fei = standard(s->shou\_du);

s->nextshou = p->nextshou;

p->nextshou = s;

p = p->nextshou;

}

if (p == NULL) //判断信息表是否创建成功

{

cout << "创建失败请重新创建！" << endl;

ShouCreat();

}

}

void Shoufei::ShouInsert(char\* sus, int du, int fei) //插入宿舍收费信息（头插法）

{

ShouNode \*s, \*p;

s = new ShouNode;

s->shou\_du = du;

s->shou\_fei = fei;

strcpy(s->shou\_sus, sus);

p = ShouListHead;

s->nextshou = p->nextshou;

p->nextshou = s;

}

ShouNode \*Shoufei::ShouFind(char\* sus)

{

ShouNode \*p;

p = ShouListHead->nextshou;

while (p->nextshou && strcmp(p->shou\_sus, sus) != 0) //循环终止条件为p->nextru不为空 而且没有找到宿舍

{

p = p->nextshou;

}

if (strcmp(p->shou\_sus, sus) == 0)

{

return p;

}

else

{

cout << "未找到该宿舍信息！" << endl;

return NULL;

}

}

void Shoufei::ShouShow() //遍历输出

{

ShouNode \*p;

HANDLE handle;

handle = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(handle, FOREGROUND\_INTENSITY | FOREGROUND\_RED);//设置控制台字体颜色为红色

//printf("这是红色\n");

cout << "提示！规定水电量在200度，收费100元，若超出则按超出部分每平方3元收费" << endl;

//SetConsoleTextAttribute(handle, FOREGROUND\_INTENSITY);//灰色

SetConsoleTextAttribute(handle, FOREGROUND\_RED | FOREGROUND\_GREEN | FOREGROUND\_BLUE);//恢复默认的灰色

//printf("这是灰色\n");

cout << "宿舍" << '\t' << "水电度数" << '\t' << "水电费用" << endl;

for (p = ShouListHead->nextshou; p != NULL; p = p->nextshou)

{

cout << p->shou\_sus << '\t' << " " << p->shou\_du << '\t' << p->shou\_fei << endl;

}

}

void Shoufei::ShouSave() //保存信息

{

ShouNode \*p;

p = ShouListHead->nextshou;

ofstream out("shoufeia.txt");

if (!out) { cout << "不能打开文件shoufeia.txt！" << endl; return; }

while (p != NULL)

{

out << p->shou\_sus << '\t' << p->shou\_du << '\t' << p->shou\_fei << endl;

p = p->nextshou;

}

}

void Shoufei::ShouQuit() //信息写入文件

{

char choice;

cout << "是否保存宿舍水电费收费信息：？（Y/N）";

cin >> choice;

if (choice == 'y' || choice == 'Y')

{

ShouSave();

cout << "宿舍水电收费信息已保存..." << endl;

}

}

void shoufei()

{

char psus[40];

ShouNode \*pfind;

Shoufei sf;

while (1)

{

system("pause");

system("cls"); //清屏

ShouMenu();

int x;

cin >> x;

if (x == 0){

return;

}

switch (x)

{

case 1:

sf.ShouRead();

cout << "读入宿舍收费表：" << endl;

sf.ShouShow();

break;

case 2:

sf.ShouCreat();

cout << "请核对输入宿舍收费信息！" << endl << endl;

sf.ShouShow();

break;

case 3:

cout << "请输入要查找宿舍：";

cin >> psus;

pfind = sf.ShouFind(psus);

cout << "要查找的宿舍：" << pfind->shou\_sus << " 水电度数： " << pfind->shou\_du << " 水电收费： " << pfind->shou\_fei << endl;

break;

case 4:

sf.ShouShow();

break;

case 5:

sf.ShouQuit();

break;

}

}

}

//baoxiu.h

#define MAX\_NUM 10000000

#include <String>

#include<ctime>

#include <fstream>

using namespace std;

struct BaoNode{

char bao\_sus[40];

char bao\_pro[40];

char bao\_tim[40];

char bao\_zhu[40];

BaoNode \*nextbao;

};

class Baoxiu{

BaoNode \*BaoListHead;

public:

Baoxiu(); //构造函数

~Baoxiu(); //析构函数

void BaoCreat(); //创建信息

void BaoInsert(char\* sus, char\* pro, char\* tim, char\* zhu); //插入信息

BaoNode \*BaoFind(char \*sus); //查找信息，传入参数学号

void BaoRead(); //从文件读入信息

void BaoSave(); //保存信息到文件

void BaoQuit();

void BaoShow(); //遍历输出信息

};

void BaoMenu()

{

time\_t t;

time(&t);

cout << "---------------------------------宿舍房屋报修模块-------------------------------" << endl;

cout << "\t\t\t 本地时间:" << ctime(&t);

cout << "--------------------------------------------------------------------------------\n" << endl;

cout << "\t\t 1.读入宿舍报修信息 " << endl;

cout << "\t\t 2.录入宿舍报修信息 " << endl;

cout << "\t\t 3.查找已有宿舍信息 " << endl;

cout << "\t\t 4.输出所有宿舍信息 " << endl;

cout << "\t\t 5.保存宿舍报修信息 " << endl;

cout << "\t\t 0.返回信息主菜单 " << endl;

cout << "\n\t\t请选择：";

}

Baoxiu::Baoxiu() //构造函数

{

BaoListHead = new BaoNode;

BaoListHead->nextbao = NULL;

}

Baoxiu::~Baoxiu() //析构函数

{

BaoNode \*p;

while (BaoListHead)

{

p = BaoListHead;

BaoListHead = BaoListHead->nextbao;

delete p;

}

BaoListHead = NULL;

}

void Baoxiu::BaoCreat() //创建学生信息表

{

int n;

BaoNode \*p, \*s;

p = BaoListHead;

cout << "请输入录入宿舍条数：";

cin >> n;

for (int i = 1; i <= n; i++)

{

s = new BaoNode;

cout << "宿舍："; cin >> s->bao\_sus;

cout << "报修项目："; cin >> s->bao\_pro;

cout << "预约时间："; cin >> s->bao\_tim;

cout << "维修状态："; cin >> s->bao\_zhu;

s->nextbao = p->nextbao;

p->nextbao = s;

p = p->nextbao;

}

if (p == NULL) //判断维修表是否创建成功

{

cout << "创建失败请重新创建！" << endl;

BaoCreat();

}

}

void Baoxiu::BaoShow() //遍历输出

{

BaoNode \*p;

cout << "宿舍" << '\t' << "报修项目" << '\t' << "预约时间" << '\t' << "维修状态" << endl;

for (p = BaoListHead->nextbao; p != NULL; p = p->nextbao)

{

cout << p->bao\_sus << '\t' << p->bao\_pro << '\t' << p->bao\_tim << '\t' << p->bao\_zhu << endl;

}

}

void Baoxiu::BaoInsert(char\* sus, char\* pro, char\* tim, char\* zhu) //插入信息（头插法）

{

BaoNode \*s, \*p;

s = new BaoNode;

strcpy(s->bao\_sus, sus);

strcpy(s->bao\_pro, pro);

strcpy(s->bao\_tim, tim);

strcpy(s->bao\_zhu, zhu);

p = BaoListHead;

s->nextbao = p->nextbao;

p->nextbao = s;

}

BaoNode \*Baoxiu::BaoFind(char\* sus)

{

BaoNode \*p;

p = BaoListHead->nextbao;

while (p->nextbao && strcmp(p->bao\_sus, sus) != 0) //循环终止条件为p->nextru不为空 而且没有找到宿舍

{

p = p->nextbao;

}

if (strcmp(p->bao\_sus, sus) == 0)

{

return p;

}

else

{

cout << "未找到该宿舍维修信息！" << endl;

return NULL;

}

}

void Baoxiu::BaoRead() //从文件读入数据

{

BaoNode \*p;

p = BaoListHead;

ifstream in("baoxiua.txt");

if (!in) { cout << "文件baoxiua.txt中没有报修信息，请先录入报修信息!" << endl; return; }

while (!in.eof())

{

char sus[40]; char pro[40]; char tim[40]; char zhu[40];

in >> sus >> pro >> tim >> zhu;

BaoInsert(sus, pro, tim, zhu);

}

}

void Baoxiu::BaoSave() //保存信息

{

BaoNode \*p;

p = BaoListHead->nextbao;

ofstream out("baoxiua.txt");

if (!out) { cout << "不能打开文件baoxiua.txt！" << endl; return; }

while (p != NULL)

{

out << p->bao\_sus << '\t' << p->bao\_pro << '\t' << p->bao\_tim << '\t' << p->bao\_zhu << endl;

p = p->nextbao;

}

}

void Baoxiu::BaoQuit() //信息写入文件

{

char choice;

cout << "是否保存报修信息：？（Y/N）";

cin >> choice;

if (choice == 'y' || choice == 'Y')

{

BaoSave();

cout << "报修信息已保存..." << endl;

}

}

void baoxiu()

{

char psus[40];

BaoNode \*pfind;

Baoxiu bx;

while (1)

{

system("pause");

system("cls");

BaoMenu();

int x;

cin >> x;

if (x == 0){

return;

}

switch (x)

{

case 1:

bx.BaoRead();

cout << "读入宿舍报修表：" << endl;

bx.BaoShow();

break;

case 2:

bx.BaoCreat();

cout << "请核对输入宿舍报修信息！" << endl << endl;

bx.BaoShow();

break;

case 3:

cout << "请输入要查找宿舍：";

cin >> psus;

pfind = bx.BaoFind(psus);

cout << "要查找的宿舍：" << pfind->bao\_sus << " 报修项目:" << pfind->bao\_pro << " 预约时间：" << pfind->bao\_tim << " 维修状态：" << pfind->bao\_zhu << endl;

break;

case 4:

bx.BaoShow();

break;

case 5:

bx.BaoQuit();

break;

}

}

}

//Dengji.h

#define MAX\_NUM 10000000

#include <String>

#include<ctime>

#include <fstream>

using namespace std;

struct DenNode{

char den\_nam[40];

char den\_tel[40];

char den\_op[40];

char den\_intime[40];

char den\_outtime[40];

DenNode \*nextden;

};

class Dengji{

DenNode \*DenListHead;

public:

Dengji(); //构造函数

~Dengji(); //析构函数

void DenInsert(char\* nam, char\* op, char\* tel, char\* intime, char\* outtime); //插入信息

DenNode \*DenFind(char \*nam); //查找信息，传入参数学号

void DenRead(); //从文件读入信息

void DenSave(); //保存信息到文件

void DenQuit();

void DenShow(); //遍历输出信息

};

void DenMenu()

{

time\_t t;

time(&t);

cout << "---------------------------------宿舍房屋报修模块-------------------------------" << endl;

cout << "\t\t\t 本地时间:" << ctime(&t);

cout << "--------------------------------------------------------------------------------\n" << endl;

cout << "\t\t 1.读入人员登记信息 " << endl;

cout << "\t\t 2.进入登记信息 " << endl;

cout << "\t\t 3.离开登记信息 " << endl;

cout << "\t\t 4.查找外来登记信息 " << endl;

cout << "\t\t 5.输出所有登记信息 " << endl;

cout << "\t\t 6.保存登记信息 " << endl;

cout << "\t\t 0.返回信息主菜单 " << endl;

cout << "\n\t\t请选择：";

}

Dengji::Dengji() //构造函数

{

DenListHead = new DenNode;

DenListHead->nextden = NULL;

}

Dengji::~Dengji() //析构函数

{

DenNode \*p;

while (DenListHead)

{

p = DenListHead;

DenListHead = DenListHead->nextden;

delete p;

}

DenListHead = NULL;

}

void Dengji::DenShow() //遍历输出

{

DenNode \*p;

cout << "来访人姓名" << '\t' << "来访理由" << '\t' << "联系方式" << '\t' << "来访时间" << '\t' << "离开时间" << endl;

for (p = DenListHead->nextden; p != NULL; p = p->nextden)

{

cout << p->den\_nam << '\t' << p->den\_op << '\t' << p->den\_tel << '\t' << p->den\_intime << '\t' << p->den\_outtime << endl;

}

}

void Dengji::DenInsert(char\* nam, char\* op, char\* tel, char\* intime, char\* outtime) //插入信息（头插法）

{

DenNode \*s, \*p;

s = new DenNode;

strcpy(s->den\_nam, nam);

strcpy(s->den\_op, op);

strcpy(s->den\_tel, tel);

strcpy(s->den\_intime, intime);

strcpy(s->den\_outtime, outtime);

p = DenListHead;

s->nextden = p->nextden;

p->nextden = s;

}

DenNode \*Dengji::DenFind(char\* nam)

{

DenNode \*p;

p = DenListHead->nextden;

while (p->nextden && strcmp(p->den\_nam, nam) != 0) //循环终止条件为p->nextru不为空 而且没有找到宿舍

{

p = p->nextden;

}

if (strcmp(p->den\_nam, nam) == 0)

{

return p;

}

else

{

cout << "未找到外来登记信息！" << endl;

return NULL;

}

}

void Dengji::DenRead() //从文件读入数据

{

DenNode \*p;

p = DenListHead;

ifstream in("dengjia.txt");

if (!in) { cout << "文件dengjia.txt中没有报修信息，请先录入报修信息!" << endl; return; }

while (!in.eof())

{

char nam[40]; char op[40]; char tel[40]; char intime[40]; char outtime[40];

in >> nam >> op >> tel >> intime >> outtime;

DenInsert(nam, op, tel, intime, outtime);

}

}

void Dengji::DenSave() //保存信息

{

DenNode \*p;

p = DenListHead->nextden;

ofstream out("dengjia.txt");

if (!out) { cout << "不能打开文件dengjia.txt！" << endl; return; }

while (p != NULL)

{

out << p->den\_nam << '\t' << p->den\_op << '\t' << p->den\_tel << '\t' << p->den\_intime << '\t' << p->den\_outtime << endl;

p = p->nextden;

}

}

void Dengji::DenQuit() //信息写入文件

{

char choice;

cout << "是否保存登记信息：？（Y/N）";

cin >> choice;

if (choice == 'y' || choice == 'Y')

{

DenSave();

cout << "登记信息已保存..." << endl;

}

}

void dengji()

{

char nam[40], op[40], tel[40], intime[40];

char outtime[40] = "未离开";

DenNode \*pfind;

Dengji dj;

while (1)

{

system("pause");

system("cls");

DenMenu();

int x;

cin >> x;

if (x == 0){

return;

}

switch (x)

{

case 1:

dj.DenRead();

cout << "读入宿舍外来人员登记表：" << endl;

dj.DenShow();

break;

case 2:

time\_t a;

time(&a);

cout << "来访人姓名："; cin >> nam;

cout << "来访理由："; cin >> op;

cout << "联系方式："; cin >> tel;

strcpy(intime, ctime(&a));

dj.DenInsert(nam, op, tel, intime, outtime);

cout << "请核对输入外来人员登记信息！" << endl;

dj.DenShow();

break;

case 3:

time\_t b;

time(&b);

cout << "来访人姓名："; cin >> nam;

pfind = dj.DenFind(nam);

strcpy(pfind->den\_outtime, ctime(&b));

cout << "离开时间已登记！" << ctime(&b) << endl;

break;

case 4:

cout << "请输入要查找来访人姓名：";

cin >> nam;

pfind = dj.DenFind(nam);

cout << "要查找的人员：" << pfind->den\_nam << " 来访理由:" << pfind->den\_op << " 联系方式：" << pfind->den\_tel << " 来访时间：" << pfind->den\_intime << " 离开时间：" << pfind->den\_outtime << endl;

break;

case 5:

dj.DenShow();

break;

case 6:

dj.DenQuit();

break;

}

}

}

//Chaqin.h

#define MAX\_NUM 10000000

#include <String>

#include<ctime>

#include <fstream>

using namespace std;

void chaqin()

{

int n;

cout << "请输入楼层数："; cin >> n;

for (int i = 1; i <= n; i++){

int a, b;

cout << "请输入" << i << "楼层查寝宿舍区间：";

cin >> a >> b;

cout << i << "楼查寝宿舍" << endl;

suijishu(a, b);

cout << endl << endl;

}

}

//main.cpp

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<iostream>

#include<windows.h>

#include "sushe.h"

#include "Ruzhu.h"

#include"Fuzhu.h"

#include"Weisheng.h"

#include"Shoufei.h"

#include"baoxiu.h"

#include"Dengji.h"

#include"Chaqin.h"

#include<cstdlib>

using namespace std;

int main()

{

Systemdoor();

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

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

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

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

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

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

cout << " \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 欢迎进入学生宿舍管理信息管理系统 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

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

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

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

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

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

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

while（1）{

system("pause");

system("cls"); //清屏

time\_t t;

time(&t);

cout << "---------------------------------学生宿舍管理信息系统主菜单-------------------------------" << endl;

cout << "\t\t 本地时间:" << ctime(&t);

cout << "------------------------------------------------------------------------------------------\n" << endl;

cout << "\t\t 0.安全退出系统 " << endl;

cout << "\t\t 1.宿舍信息模块 " << endl;

cout << "\t\t 2.学生入住模块 " << endl;

cout << "\t\t 3.卫生检查模块 " << endl;

cout << "\t\t 4.水电收费模块 " << endl;

cout << "\t\t 5.房屋报修模块 " << endl;

cout << "\t\t 6.外来人员登记 " << endl;

cout << "\t\t 7.宿舍随机查寝 " << endl;

cout << "\n\n\t请输入选择：";

int x;

cin >> x;

if (x == 0)

return 0;

switch (x)

{

case 1:

sushe();

break;

case 2:

ruzhu();

break;

case 3:

weisheng();

break;

case 4:

shoufei();

break;

case 5:

baoxiu();

break;

case 6:

dengji();

break;

case 7:

chaqin();

break;

}

}

cout << "退出成功！";

return 0;

}