int testshu(char []);验证函数

void enter();最初进入页面

void userfunction();用户功能

void uregister();用户注册

void search\_fn();根据电影名查询

void selectkey(); 根据电影名与影院查询

void choose\_sort();根据2D、3D筛选电影

void choose\_kind(); 根据英语、国语筛选电影

void search\_price();根据票价排序所有场次

void search\_margin();根据余票数排序所有场次

void search\_time();根据时间排序所有场次

int test\_changci(char [20]);检验用户输入场次号是否正确

int test\_time(char [],char []);对比电影时间给出提醒

int tuijian1(int [H][L],char [20],char [20]); 推荐一个座位并显示座位图函数

int tuijian2(int [H][L],char [20],char [20]); 推荐两个座位并显示座位图函数

int tuijian3(int [H][L],char [20],char [20]); 推荐三个座位并显示座位图函数

int zizhu1(int [H][L]); 用户自己选一个座位函数

int zizhu2(int [H][L]);用户自己选俩个座位函数

int zizhu3(int [H][L]); 用户自己选三个座位函数

int test\_shuliang(char []);检测用户订单数量

int test\_same(char [],char []);检测用户购买同一部影片

void read\_order(); dingdanA读取函数

void read\_cin();场次信息读取函数

void read\_user();userinformation读取函数

void goupiao(); 购票函数！！

void check\_price(char [20],int,char getci[20]); 检验用户余额函数并充值

int test\_yu(int); 检测余额

void change\_yu(char [20]);充值后改变用户余额

void change\_yupiaoshu(char [],int);改变余票数

void read\_user();

int test\_phone(char []);检验用户手机长度是否为十一位数字

void change\_phone(char []);修改用户手机号码

int test\_email();检验用户手机号码格式

void change\_email(char []);修改用户邮箱

int test\_long(char []);检验密码长度

int testCODE(char[],char[]); 检验确认密码输入正确

void change\_pass(char []);修改用户密码

void PIM(char []);用户个人信息管理

void change\_yue(char []);单独充值

void inquiry\_order(char [20]);

void cancel\_order();取消订单

void admfunction(char []);管理员功能

void admdenglu();管理员登陆

void examine\_order(char []);管理员查询订单

void tongji\_dingdan(char []);管理员统计订单

void px\_dingdan(char [20]);排序场次订单数

void px\_rate(char [20]);排序场次上座率

void px\_changci(char [20]);排序场次总票价

void px\_yingting(char [20]);排序影厅总票价

void read\_admin();读取管理员文件信息

void examine\_hall(char [20]);管理员查询影厅信息

void read\_hall();影厅信息进入函数

void examine\_chang(char [20]);管理员查询所有开设的影厅信息

void admin\_pass(char []);

void admin\_email(char []);

int test\_email2(char []);检验邮箱格式标准

int test\_long2(char [10]);测试密码长度

int testCODE2(char [],char []);检验确认密码输入正确

void increase\_ci(char []); 添加场次

void increase\_room(char []); 添加新影厅

int test\_number(char numbera[]);检验输入的影厅总座位为数字

int nameuser=0,nameode=0,namecin=0,nameadmin=0,namehall=0;

int i,n;

char userID[20]={'\0'};

char admID[20]={'\0'};

# 功能：

# 用户功能

登陆：

输入账号和密码登陆系统

注册：

新用户注册信息

查询（选项栏可容错）

 1.根据影片名查询，默认返回当日场次；

2.根据影片名+影院名查询，默认返回当日场次；

3.可用放映场次类型D如英语、中文等）、放映影厅类型 如3D、4D等）等过滤查询结果

排序

1.根据放映开始时间排序所有场次

 2.根据场次票数排序所有场次

3.根据场次余票数排序所有场次

购票

1.查询之后允许用户购票，购票时具有座位推荐功能

2.同一个订单只包含一个场次，可以包含多个座位最多 3个）；

3.购买的座位与已售出的座位不能相隔仅一个座位；

4.同一名用户购买的多个影片场次的时间冲突时，系统应给出 提示；

5.同一名用户从第二次开始购买相同影片的影票时，系 统应给出提示；

6.规定一名用户一天最多购买五个场次的影票；

7.每个场次购票的座位数不能超过座位总数；

8.座位编号不能冲突；

9.用户账户余额不足时，需要先充值再继续购票

个人信息管理

对个人信息中的电话、密码及邮箱进行修改

取消订单

在未付款之前用户可以取消自己的订单

假设：可模拟付款

查询订单

查看用户曾经的购票记录

# 管理员功能

登陆

输入管理员ID和密码登陆系统。

订单管理

查看用户订单情况： 1.查询管理员自己所属影院的场次的订票情况

各场次已被订票的详细信息，如何时哪个用户购买了哪个场次的哪个座位

1. 统计用户订单信息：

按已购票数排序还未放映的本影院所有的场次信息

按场次上座率排序已经放映结束的所有历史场次信息

按场次总票数收入排序已经放映结束的所有历史场次信息并可计算排序显示出影厅总票数收入

影厅管理

影厅查询：查询所有开设的影厅信息

 添加影厅：添加一个新的影厅

场次管理

 场次查询：查询所有排好的场次信息

添加场次：添加新的场次的排片

排片时间应该考虑错开各影厅入场和散场的时间错开人流，给出智能提示

个人信息管理

对个人信息中的密码及邮箱进行修改