

电影院管理系统设计报告

学院：国际学院

专业：物联网工程

设计学生：

笪良润 学号： 班级：2018215119

吕竟豪 学号： 班级：2018215119

冯聪聪 学号： 班级：2018215119

刘宇帆 学号： 班级：2018215119

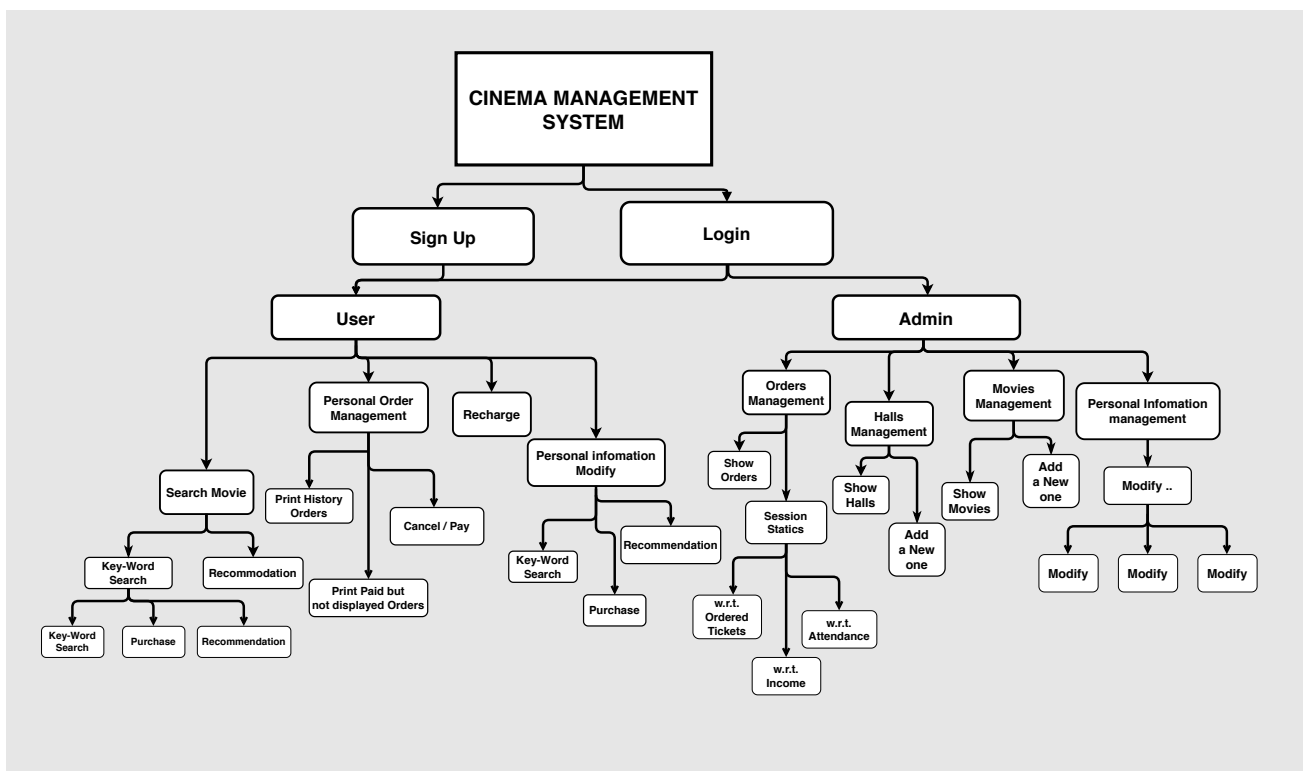
Liangrun Da

目录

目录	2
系统总体设计	3
系统总体功能描述	3
接口定义规范	3
系统的数据结构描述	3
系统的开发环境及运行环境描述：	4
模块详细描述	5
1.管理员模块	5
功能概述	5
处理流程	5
模块的输入与输出	9
模块的接口描述	9
边界情况描述以及处理方法：	13
2.用户模块	14
功能概述	14
处理流程	14
模块的输入和输出：	19
模块的接口描述：	19
边界情况描述以及处理方法：	26

系统总体设计

系统总体功能描述



电影票销售系统主要包括了用户模块和管理员模块。

接口定义规范

本系统源代码的函数分两类，第一类是实现用户名和管理员的功能的函数，这类函数的命名采用“一级功能函数名_二级功能函数名_三级功能函数名”的方式，例如：user是用户功能函数，user_purchase是实现用户购买的函数，user_purchase_recommenseat是用户购买的时候实现座位推荐的函数；第二类是实现数据的转化、输出、输入、搜索、排序等的函数，采用的是“data_功能_操作对象”的命名方式，例如：data_output_movie是将电影的信息（操作对象）输出（功能）到屏幕上的函数。

系统的数据结构描述

- I. 系统需要处理的数据：用户的ID、姓名、性别、电话等数据，管理员的ID、所属影院、姓名等数据，影厅的类型、大小、座位分布等数据，电影的名字、时长、售票情况、价格等数据，订单的下单人、购买座位数、优惠信息等数据。
- II. 这些数据以字符、字符串、整型变量和浮点数等形式存在于程序中，以二进制方式存储在.dat文件中。

本系统用到的文件有：

Admin.dat (管理员信息文件)

Cust.dat (用户信息文件)

Movie.dat (电影信息文件)

Order.dat (订单信息文件)

Hall.dat (影厅信息文件)

Num.dat (历史存储的各项信息的数据个数)

III. 运行过程中使用的的数据结构

采用结构体数组的数据结构存储主要信息，辅以其他结构，例如以one-hot的形式存储座位图等。

系统的开发环境及运行环境描述：

I. 运行环境：

硬件环境：

MacBook Pro (13-inch, 2017, Two Thunderbolt 3 ports)

处理器：2.3 GHz Intel Core i5

内存：8 GB 2133 MHz LPDDR3

图形卡：Intel Iris Plus Graphics 640 1536 MB

软件环境：

macOS Mojave 10.14 (18A391)

II. 开发环境：

Xcode Version 9.4.1 (9F2000)

Liangrun Da

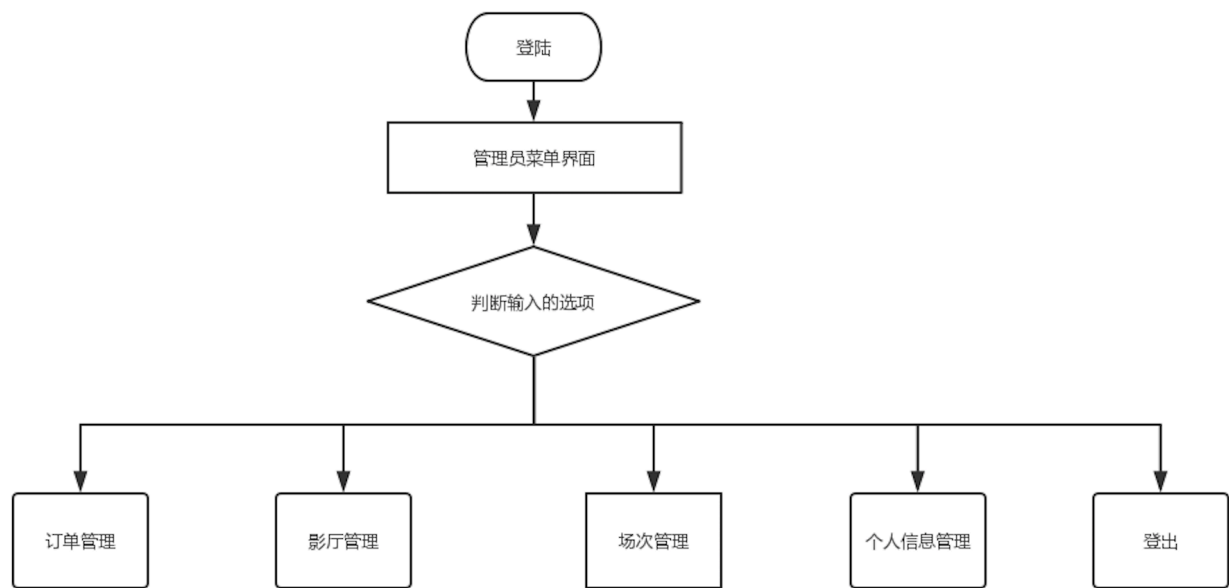
模块详细描述

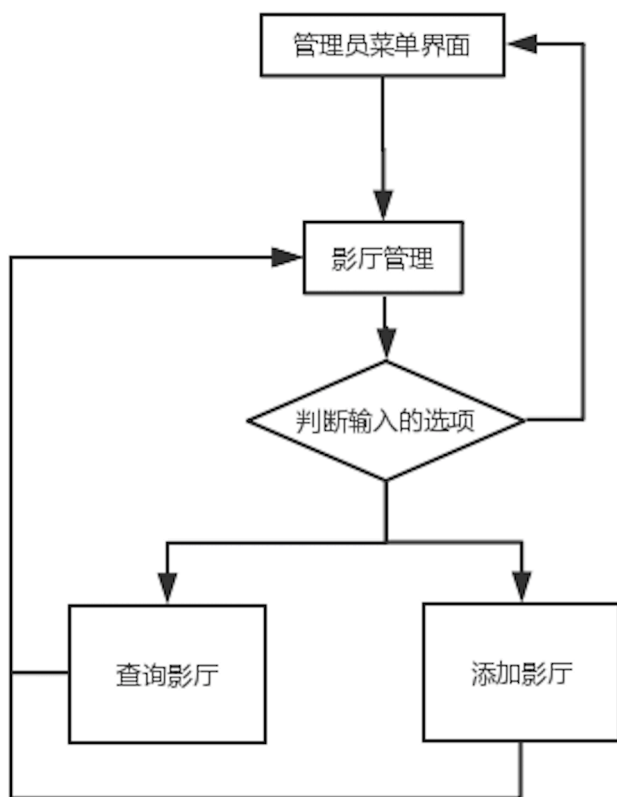
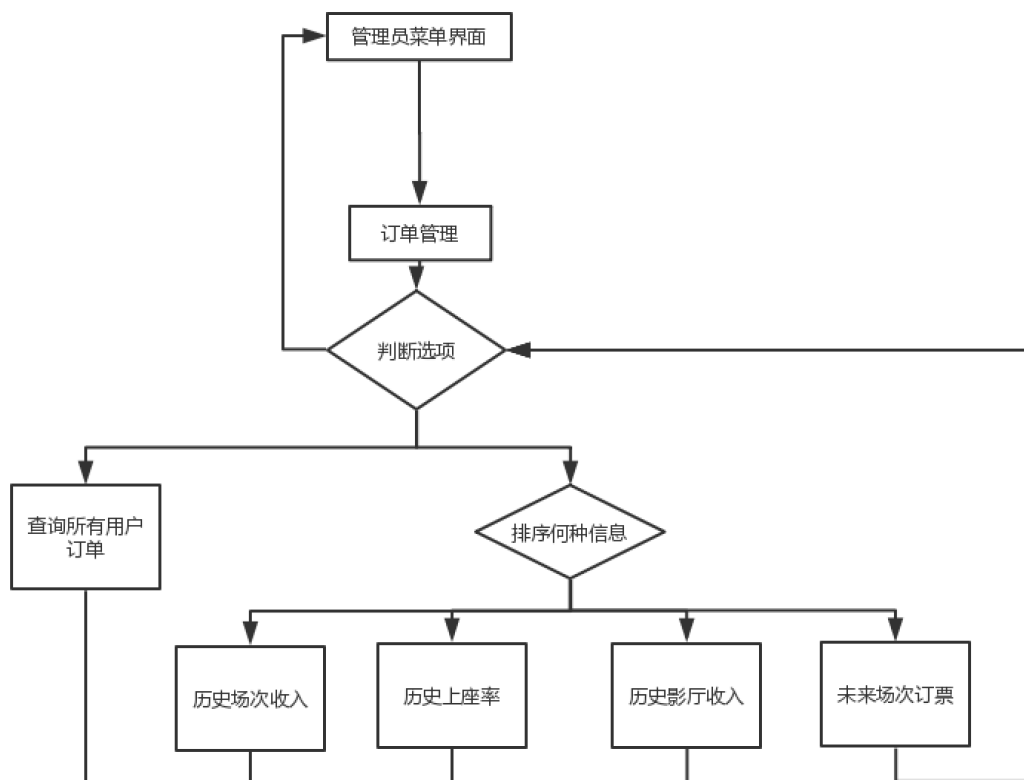
1.管理员模块

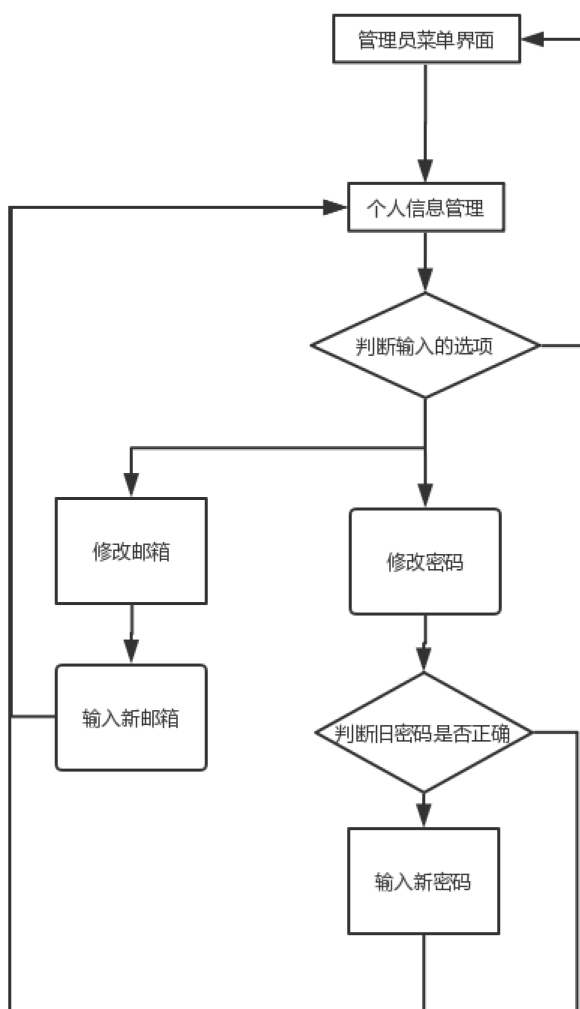
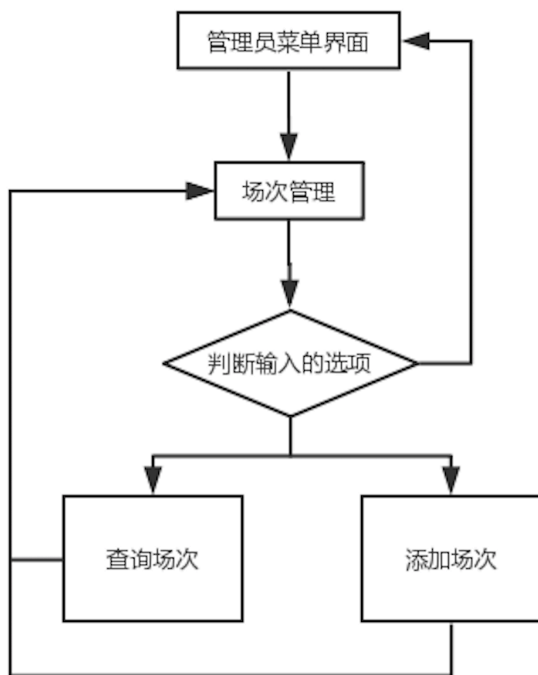
功能概述

管理员可以通过该模块查看所属影院的用户订单情况，管理影厅，管理影片，管理个人信息。

处理流程







管理员登陆以后会进入管理员主界面，并将进入系统的管理员所有信息传入函数admin。输出五个选项，分别是

- 1.订单管理
- 2.影厅管理
- 3.电影管理
- 4.个人信息管理
- 5.登出

以下分别从五个功能分别描述处理流程——

功能一：订单管理

进入订单管理之后，会提示管理员三个选项，分别是：A.展示本影院的所有订单 B.了解本影院的订单统计情况 C.返回

选择A之后，首先会调用data_get_adminorder函数，将所有本影院的订单在订单结构体数组中的下标全部存储到一个int类型的数组里，如果本影院没有订单，就会告知管理员这一信息，否则会将所有本影院订单的详细信息都打印在屏幕上，之后会返回到订单管理，重新选择。

选择B之后，会出现4个选项：(1).按照订票数排序打印所有还没有播放的场次信息 (2).按照场次的收入排序打印所有已经播放的场次信息 (3).按照场次的上座率打印所有已经播放的场次信息 (4).按照影厅的收入排序打印。不管选择了什么选项，都会进入时间选项，即查看从现在开始到未来一段时间或者到过去一段时间的信息。如果选择了选项(1)，则会立即获取当前系统时间，然后在电影的结构体数组里查找所有开始时间大于当前系统时间的电影，并将这些电影在结构体数组中的下标存储到一个整型数组当中，对这个整型数组进行排序，采用的是qsort函数，然后将排序过的数组，其中存储的下标对应的电影信息打出来；如果选择了选项(2)，跟上面的处理类似，只不过在调用qsort函数的时候，需要重新编写排序的标准，即以收入排序；选择了(3)也是一样的，只不过是以上座率进行排序然后打印在屏幕上；选择(4)，则会通过每个场次的收入，将收入加到对应影厅里，然后以影厅收入为标准进行排序。

选择了C选项之后，就会返回到管理员主菜单界面。

功能二：影厅管理

影厅管理分为三个选项，1.查看自己所属影院的所有影厅 2.添加一个影厅 3.返回。选择了1之后，就会在整个影厅结构体数组中查询，然后只要找到跟管理员的影院名一样的影厅，就会将其输出在屏幕上。选择了2之后，进入admin_hall_createhall函数，就会询问要添加的影厅类型和大小，之后就能够在影厅的结构体数组当中去写入这个影厅的信息。选择了3之后，就会返回到管理员主菜单界面。

功能三：电影管理

电影管理分为三个选项，1.查看自己所属影院的所有场次 2.添加一个场次 3.返回。选择了1之后，就会在整个电影的结构体数组中查询，只要找到跟管理员的影院名一样的电影，就会将其输出在屏幕上。选择了2以后，进入排片系统，一开始就会搜索本影院的所有影厅，如果还没有影厅，就会告知管理员不能添加电影并强制返回，如果有的话，就会让管理员依次输入电影名、价格、影厅号等等，接着会去提示管理员输入排片日期，然后会在这个日期里去检索哪些时间段排片子比较多，提示管理员错开这些时间段排片，最后输入小时分钟，会去验证这个小时分钟是否合法（比如是否跟其他场次冲突了影厅，结束时间是否大于开始时间，等等），如果验证都通过，就会让管理员再补齐其他信息，最后将这些信息全部录入电影的结构体数组中，并将新影片的信息打印出来。

功能四：个人信息管理

这一部分只能允许管理员修改自己的密码或者邮箱，如果修改密码的话，需要先去验证旧密码，通过之后才能输入新密码，更新结构体的密码这一属性。邮箱也是类似的处理流程。

功能五：登出

会将登陆状态的变量login_status修改为-1，这样就能够实现登出。

模块的输入与输出

输入：

	数据描述	数据结构
管理员登陆的账号密码	stdin输入	字符数组
影片信息	从movie.dat中输入	结构体数组
影厅信息	从hall.dat中输入	结构体数组
订单信息	从order.dat中输入	结构体数组
顾客信息	从cust.dat中输入	结构体数组
选择菜单的选项	stdin输入	字符变量
选择查看订单的日期	stdin输入	整型变量
添加影厅的信息	stdin输入	字符数组
添加电影的信息	stdin输入	字符数组、整型变量等
修改个人信息	stdin输入	字符数组

输出：

	数据描述	数据结构
菜单信息	输出到屏幕	字符串
查询到的订单信息	输出到屏幕	结构体数组
查询到的影厅信息	输出到屏幕	结构体数组
查询到的场次信息	输出到屏幕	结构体数组
添加的影厅信息	输出到hall.dat文件	结构体数组
添加的影片信息	输出到movie.dat文件	结构体数组
修改成功后的个人信息	输出到admin.dat文件	结构体数组

模块的接口描述

int Login(int *index)

功能：登陆用户界面的函数

参数: index: 用于返回登陆的用户在对应的结构体数组中的下标

返回: 返回登陆状态, -1表示没有登陆, 1表示管理员登陆, 2表示普通顾客登陆

int data_comparmov_purchased(const void *a, const void *b)

功能: 比较电影的购票数目

参数: a: 影片的数组下标 b: 影片的数组下标

返回: 返回正数表示 $a > b$, 0表示 $a = b$, 负数表示 $a < b$

int data_compar_movattendance(const void *a, const void *b)

功能: 比较电影的上座率

参数: a: 影片的数组下标 b: 影片的数组下标

返回: 返回正数表示 $a > b$, 0表示 $a = b$, 负数表示 $a < b$

int data_compar_movincome(const void *a, const void *b)

功能: 比较电影的收入

参数: a: 影片的数组下标 b: 影片的数组下标

返回: 返回正数表示 $a > b$, 0表示 $a = b$, 负数表示 $a < b$

void data_print_movie(int index)

功能: 打印一个电影

参数: index: 需要打印的电影信息下标

返回: 无返回值

char data_print_rootmenu(void)

功能: 打印管理员总菜单

参数: 无参数

返回: 选择的选项

void data_print_halllist(int hallIndex[], int num)

功能: 打印影厅列表

参数: hallIndex: 影厅下标 num: 需要打印的影厅个数

返回: 无返回值

void data_print_hall(hall *input)

功能: 打印一个影厅的信息

参数: input: 需要打印的影厅结构体的指针

返回: 无返回值

void admin_hall_createhall(int adminNO)

功能: 创建一个新影厅

参数: adminNO: 创建影厅的管理员在管理员数组中的下标

返回: 无返回值

`void data_transformgraph(char size, char *graph)`

功能：将一个one-hot结构转化为一个空的座位图

参数：size：影厅的大小 graph：座位图

返回：无返回值

`int admin_movie_createmovie(int adminNO)`

功能：创建新场次

参数：adminNO: 管理员下标

返回：返回1表示创建成功，0表示不成功

`int data_compar_admin_cinema(const void *a, const void *b)`

功能：比较管理员的字典序

参数：a：订单下标 b.订单下标

返回：正数 $a > b$ ，0表示 $a = b$,负数表示 $a < b$

`int data_compar_adminNO(const void *a, const void *b)`

功能：比较管理员

参数：a：订单下标 b.订单下标

返回：正数 $a > b$ ，0表示 $a = b$,负数表示 $a < b$

`int admin_order_binarysearch_order_lower_bound(int index[], int key)`

功能：二分查找左边界

参数：index: 订单下标 key: 搜索的关键字

返回：返回左边界

`void data_input_fixedlength(int lower, int upper, char input[])`

功能：输入定长的字符串

参数：lower:下界 upper:上界 input:存储输入的字符串

返回：正数 $a > b$ ，0表示 $a = b$,负数表示 $a < b$

`double data_input_double()`

功能：输入一个浮点数

参数：无参数

返回：返回读入的浮点数

`int data_get_month()`

功能：获得月份信息

参数：无参数

返回：返回当前月份

`int data_get_day()`

功能：获得日期

参数：无参数

返回：返回当前日期

void data_print_orderlist(int orderIndex[], int num)

功能：打印订单信息

参数：orderIndex: 订单下标 num: 需要打印的订单个数

返回：无返回值

void data_print_order(int orderNO)

功能：打印订单信息

参数：订单下标

返回：无返回值

int admin_movie_checkvalidtime(int i, time_t end_sec, time_t start_sec)

功能：检测时间是否冲突

参数：i: 下标, end_sec; 结束时间 start_sec: 开始时间

返回：返回当前月份

void data_digit2NO(int digit, char NO[])

功能：数字到字符串

参数：digit: 输入的数字 NO: 存储的字符串

返回：无返回值

void data_string2abbr(char string[], char Abbr[])

功能：获得字符串的首字母缩写

参数：string: 字符串 Abbr: 缩写

返回：无返回值

void data_fillzero(char input[], int num)

功能：填充0，来获得编号信息

参数：input: 字符串 num: 这个数组的大小，并非字符串的长度

返回：无返回值

void admin_movie_timerecommend(int adminNO, int month, int day)

功能：推荐排片时间

参数：adminNO: 当前登录的管理员下标 month: 推荐月 day: 推荐日

返回：无返回值

int data_compar_hallincome(const void *a, const void *b)

功能：比较影厅的收入

参数：a, b: 影厅的下标

返回：1: a > b 0: a = b -1 a < b

```
void data_get_adminhall(int adminHallIndex[], int *numOfAdminHall, int adminNO)
```

功能：获得当前管理员的所有影厅下标

参数：Index: 存放影厅下标 num：搜寻到的影厅个数 adminNO: 当前管理员

返回：无返回值

```
void data_get_adminorder(int orderIndex[], int *numOfOrder, int adminNO)
```

功能：获得当前管理员的所有订单

参数：Index：订单下标 num： 订单个数 adminNO: 当前管理员

返回：无返回值

```
int data_input_hour()
```

功能：输入小时

参数：无参数

返回：返回小时

```
int data_input_min()
```

功能：输入分钟

参数：无参数

返回：返回分钟

边界情况描述以及处理方法：

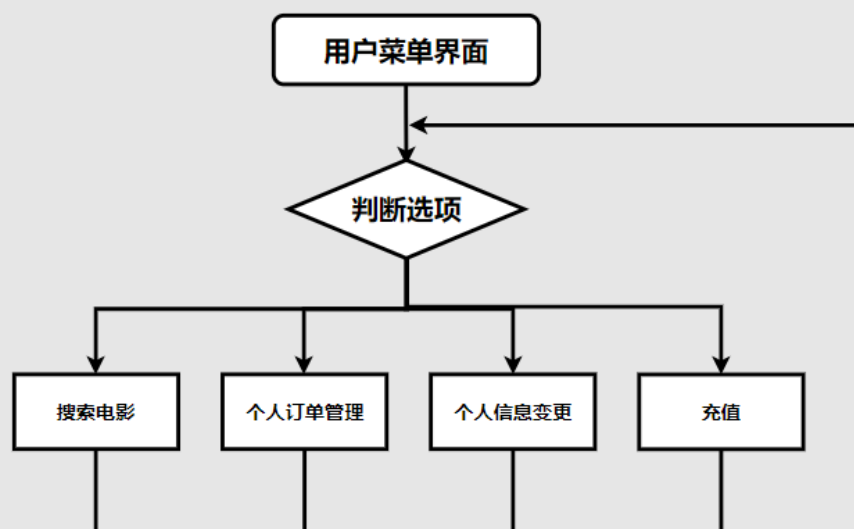
1. 输入整型数据时，先利用fgets读入存储到字符串中，再判断是否是一个有效的整型数据，最后判断是否超过int类型存储范围，如果都满足则将其通过atoi标准库函数转化为整型数据，不满足重新输入，以上流程集成在data_input_digit这一函数中
2. 输入浮点数时，先利用fgets读入存储到字符串中，再判断是否是一个有效的浮点数，如果都满足则将其通过atoi标准库函数转化为整型数据，不满足重新输入，以上流程集成在data_input_double这一函数中
3. 输入选项时，先利用fgets读入存储到字符串中，再判断是否超过一个字符，如果超过提示重新输入，再接着判断是否在给定的选项的范围内，如果不满足重新输入，满足则会返回该选项，以上流程集成在data_input_option这一函数中。
4. 输入小时和分钟时，也会检验是否是有效的时间，集成在data_input_min和data_input_hour这两个函数中。
5. 添加影片时检测是否有影厅存在，没有则停止添加
6. 添加影片时输入时间，会检验是否跟该影厅的其他电影冲突，集成在admin_movie_checkvalidtime函数中，并且还会检验是否结束时间大于开始时间，不合理就会提示重新输入
7. 验证新邮箱是否为有效的邮箱，集成在data_isvalid_email函数中
8. 添加影片的时候，要求输入影厅号码，如果没有该影厅存在，要求重新输入

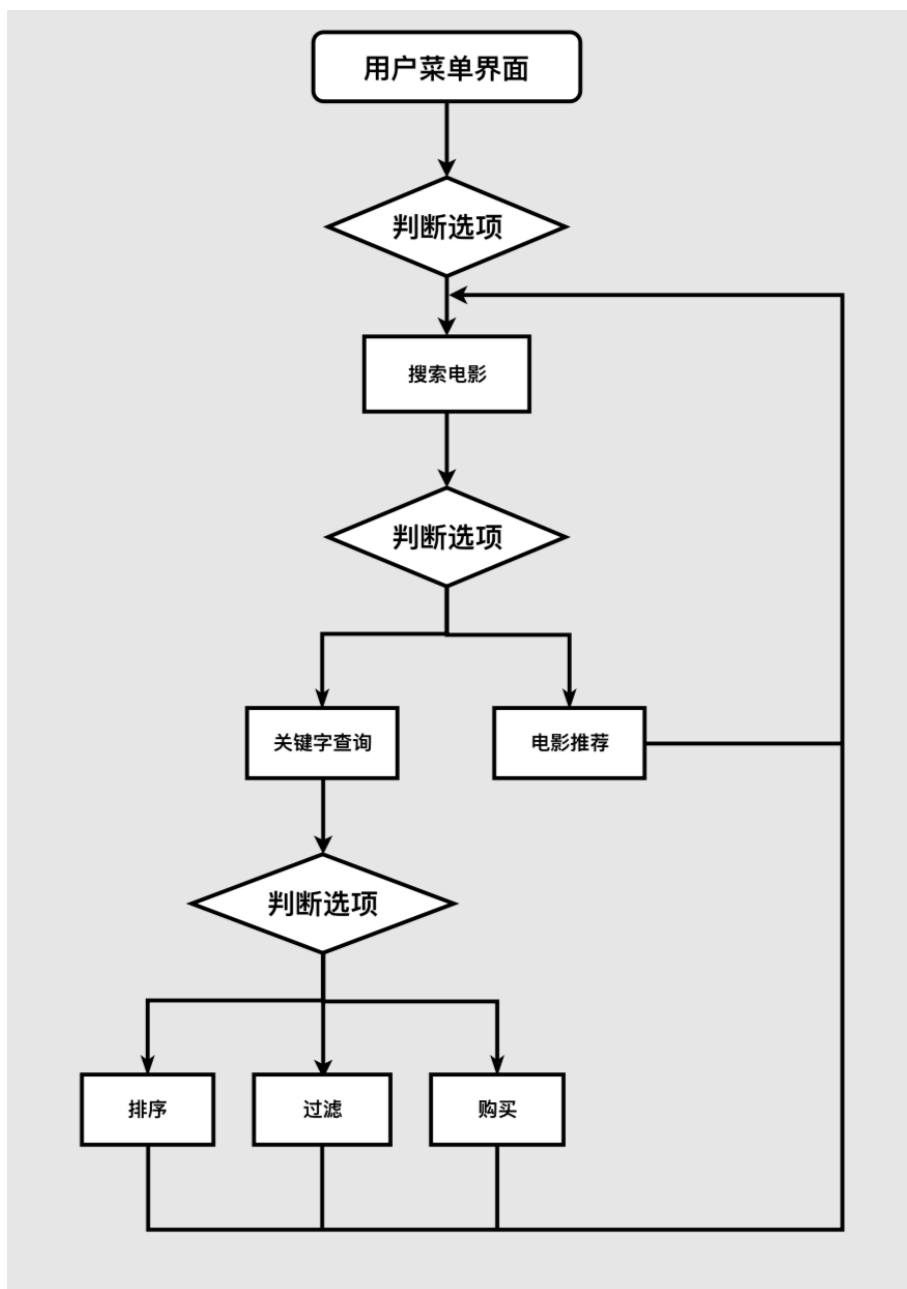
2.用户模块

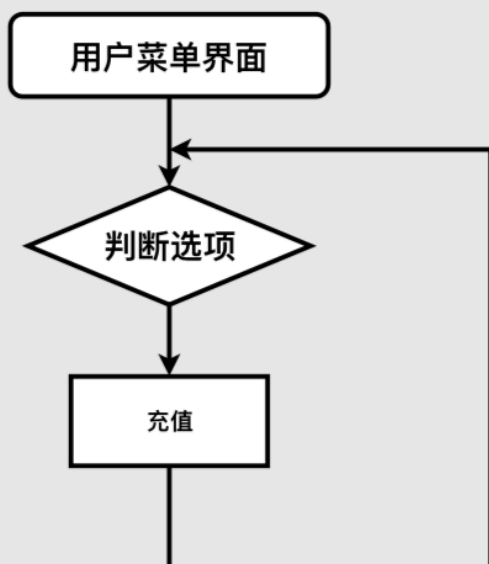
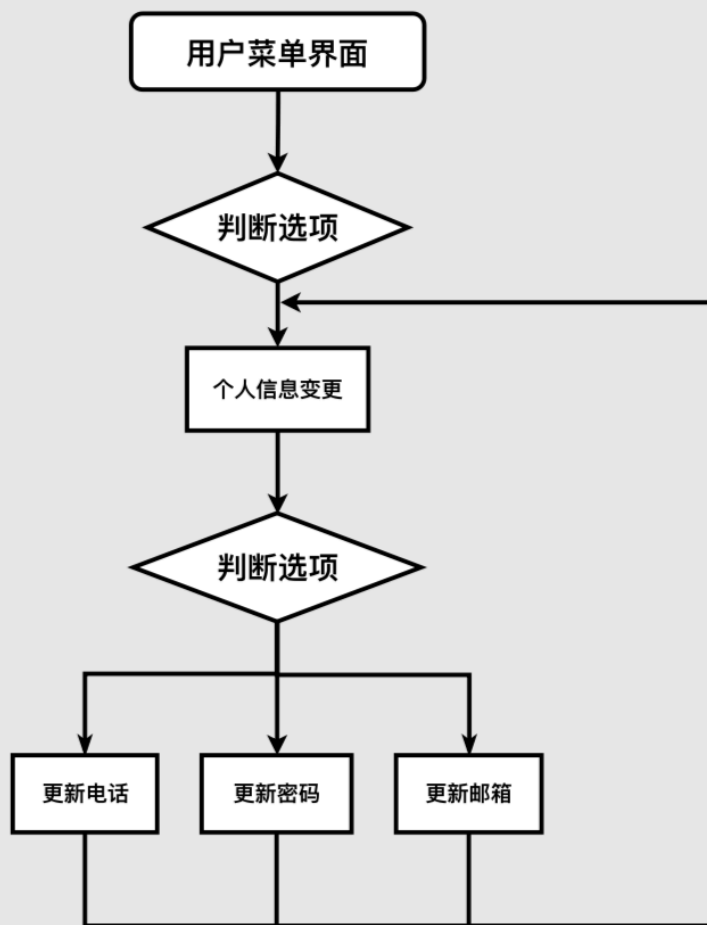
功能概述

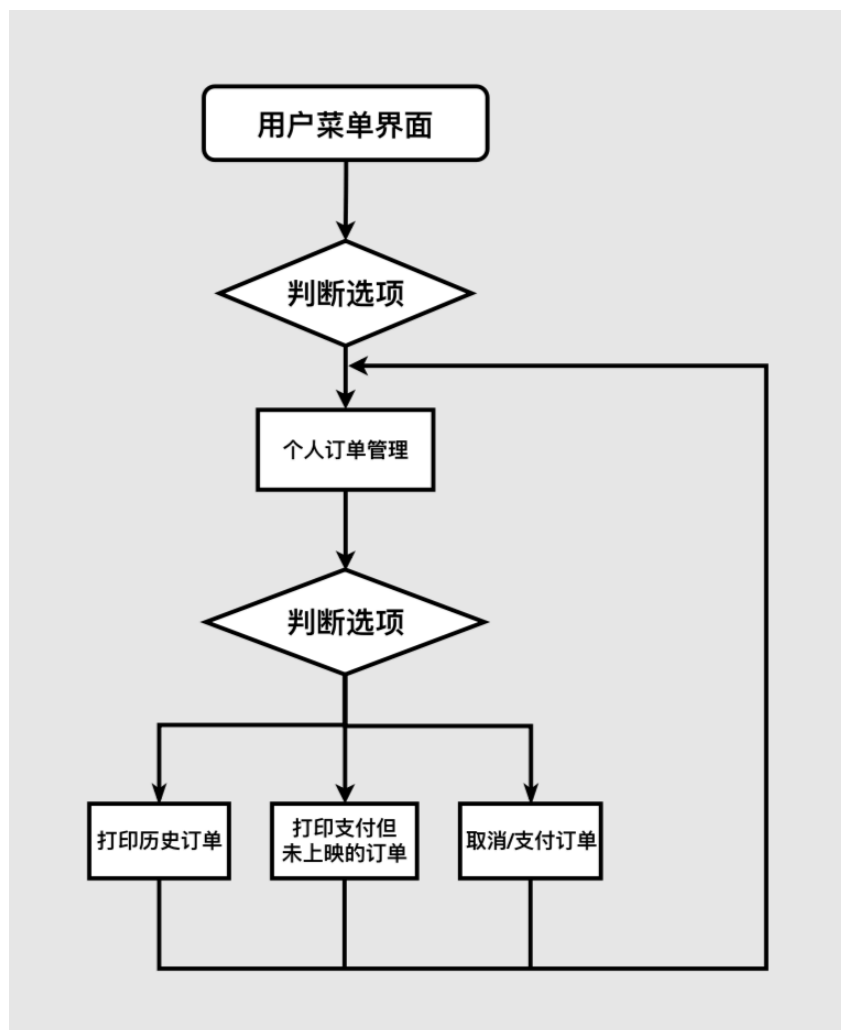
用户可通过此模块实现登陆、注册，搜索影片，依据自己可选的规则排序、筛选结果，选择电影购票下单，同时能够支付、取消订单，也能查看自己的历史订单，修改个人信息。

处理流程









用户登陆以后进入user函数，并将进入系统的用户所有信息传入函数user，首先展示给用户五个选项：

- 1.搜索、购买电影
- 2.个人订单管理
- 3.个人信息管理
- 4.充值
- 5.登出

以下分别从五个功能来阐述处理流程。

功能一：搜索、购买电影

用户进入搜索、购买电影的函数user_search之后，会提示用户三个选项: A.通过关键字搜索电影 B.热门电影推荐 C返回

选择了A之后，首先会按照字典序进行qsort排序，方便后续多次查找，接着会提示用户输入电影名[+影院名]，其中影院名是可选择项，之后进入data_input_search函数，输入之后读取到一个字符串中，然后可以在该函数内部实现这两个名称的分离，分别存入两个字符数组中。

接着对影片场次的结构体数组进行二分查找左边界，如果没有找到左边界就会直接提示用户没有该影片，如果找到了就会继续向右查找右边界，最终将这一区段的结构体数组下标全部存入一个int类型的数组中，再在这个数组中查找对应影院名的所有场次（如果没有输入影院名则不进行这一流程），再在其中默认筛选出今天的场次展示给用户，并询问用户是否需要查看明天或者后天的电影，如果需要的话再在之前还没有进行日期筛选的数组中重新进行一遍筛选。

至此初步筛选已经完成，用户可以按照屏幕上打出来的号码选择购买，也可以选择继续排序或者筛选，提供给用户四个选项：1.排序 2.筛选过滤 3.购买 4.返回。如果选择了1选项之后，会进入到排序函数`user_search_sortmovie`，提供给用户三个选项：(1)按照开始时间排序 (2)按照价格排序 (3)按照剩余票数排序。选择了三个的任意一个以后，都会在之前查找到的结果中调用`qsort`函数按照其选择的标准进行排序，再展示给用户，在此基础上返回到之前的菜单可以选择序号进行购买。如果选择了2选项之后，会进入到过滤函数`user_search_filtermovie`，提供给用户两个选项：(1)按照语言过滤 (2)按照影厅放映类型过滤，选择了任意一个之后，都会在之前的搜索结果里进行线性查找，将符合条件的场次输出到屏幕，然后供用户选择购买。

如果用户选择了购买，就会提示用户选择之前在屏幕上输出的影片序号，之后会判断该影片序号是否存在，如果存在的话，进入`user_purchase`函数，进入函数之后，首先会在订单数组中检索该用户的订单，如果在这天购买的订单数量超过5，会直接提示用户订单过多不能继续购买，并返回到上一级菜单；接着会检索用户是否已经购买过该场次的票，如果已经购买会提示用户；最后检索用户购买的其他场次的票是否和该场次的票冲突，如果冲突会提示用户。完成以上前置任务以后，提示用户输入购买的座位个数，然后判断用户的余额是否足够，如果不够会提示用户充值，并返回到上一级菜单。如果以上检验都通过了，会进入`user_purchase_recommandseat`函数进行座位推荐，座位推荐的方式是，从头到尾遍历所有的座位，在检验座位有效的前提下，以(座位行数-中心行数)+(座位列数-中心列数)作为评价标准，找到离中心位置最近的座位，并且推荐给用户。之后提示用户输入想要购买的座位，将用户输入的座位情况输入`data_isvalid_seat`函数，返回的结果可以得知是否是有效的座位，如果是有效座位提示用户下单成功，如果无效则提示用户重新输入。下单成功之后，会在订单数组中进行`realloc`操作，分配新的内存空间，以供新的订单信息录入。至此下单全部完成。

功能二：个人订单管理

选择该功能之后进入`user_order`函数，该函数首先会打印出四个选项：A.展示所有的历史订单 B.支付或者取消还未付款的未来订单 C.展示所有支付过的未来订单。选择A之后，会遍历整个订单的结构体数组，找到个人的历史订单，并将其打印出来；选择B之后，会创建一个动态分配的数组，用于存放订单结构体数组的下标，然后遍历所有的订单，找到个人的所有订单，并且是没有支付的，打印出来搜索到的订单，并提供用户选择订单的序号，然后用户输入序号，就可以进入对应的订单，然后选择1.支付 2.取消 3.返回。选择C之后，会将遍历订单结构体数组，找到已经付了款但是还没有播放的订单，然后将订单信息输出在屏幕上。

功能三：个人信息管理

选择该功能之后进入`user_information_modify`函数，然后有四个选项：A.修改电话号码 B.修改密码 C.修改邮箱 D.返回。选择了A之后，会提示用户输入新电话号码，然后通过`data_isvalid_phone`函数判断这个电话是否是合法的电话号码，如果不合法提示用户重新输入；选择了B之后，会提示用户输入旧密码，验证通过之后，可以输入新密码（如果不通过，会返回），并将该结构体的密码信息修改掉；选择了C之后，会提示用户输入新邮箱，并判断邮箱是否是合法的邮箱，如果合法就会通过，并将该结构体的邮箱信息修改掉。

功能四：充值

选择该功能之后，会提示用户充值的数目，然后更改该用户结构体的余额数值。

功能五：登出

登出之后，将`login_status`值改为-1，标记为未登录状态。

模块的输入和输出：

输入

	数据描述	数据结构
用户名和密码	stdin输入	字符数组
选项输入	stdin输入	字符变量
搜索关键字	stdin输入	字符数组
输入语言过滤	stdin输入	字符数组
输入日期	stdin输入	整型变量
用户信息	从cust.dat读入	结构体数组
影片信息	从movie.dat读入	结构体数组
影厅信息	从hall.dat读入	结构体数组
订单信息	从order.dat读入	结构体数组
需要购买的座位	stdin输入	整型变量
需要修改的密码、邮箱等信息	stdin输入	字符数组
需要充值的数目	stdin输入	浮点数变量

输出

	数据描述	数据结构
用户信息	写入到cust.dat	结构体数组
订单信息	写入到order.dat	结构体数组
菜单等提示信息	输出到屏幕	字符串
影片信息	输出到屏幕	结构体
订单信息	输出到屏幕	结构体

模块的接口描述：

```
void user(int userNO)
```

功能：登陆用户界面

参数：当前登陆用户在结构体数组中的下标

返回：无返回值

```
void user_recharge(int userNO)
```

功能：用户充值函数

参数：userID: 当前登陆用户在结构体数组中的下标

返回：无返回值

`void user_order(int userNO)`

功能：用户订单管理函数

参数：userID: 当前登陆用户在结构体数组中的下标

返回：无返回值

`void user_search(int userNO)`

功能：用户搜索电影功能函数

参数：userID: 当前登陆用户在结构体数组中的下标

返回：无返回值

`int Login(int *index)`

功能：登陆用户界面的函数

参数：index: 用于返回登陆的用户在对应的结构体数组中的下标

返回：返回登陆状态，-1表示没有登陆，1表示管理员登陆，2表示普通顾客登陆

`void data_init_cust()`

功能：初始化顾客信息

参数：无参数

返回：无返回值

`void data_init_num()`

功能：初始化各个结构体数组当中的元素个数信息

参数：无参数

返回：无返回值

`void data_init_hall()`

功能：初始化影厅的信息

参数：无参数

返回：无返回值

`void data_init_mov()`

功能：初始化电影信息

参数：无参数

返回：无返回值

`void data_init_order()`

功能：初始化订单信息

参数：无参数

返回：无返回值

`void data_input_digit(int *digit)`

功能：用于输入一个整数，防止1.超出int类型的存储范围 2.输入的不是int类型的数

参数: digit: 用于存放这个输入的整数

返回: 无返回值

void data_input_char(char *input)

功能: 字符输入

参数: input: 将读到的字符存进去

返回: 无返回值

void data_input_string(char info[], int bound)

功能: 用于读入字符串，避免越界

参数: info表示将字符串存储到这个字符数组中，bound表示info最多存储的字符数量

返回: 无返回值

char data_input_option(char high)

功能: 用于读取用户的选项

参数: high: 表示选项的最大值

返回: 返回读到的选项，一定会返回一个1到high的字符

void data_input_search(char movieKey[], char theaterKey[])

功能: 用于分离用户影片和影院的搜索字符串

参数: movieKey: 存放影片名 theaterKey: 存放影院名

返回: 无返回值

int data_size2seats(char size)

功能: 影院的大小转化为座位数

参数: size: 表示影院的大小

返回: 返回对应的座位数

int data_compar_movname(const void *a, const void *b)

功能: 比较电影的字典序大小

参数: a: 影片的数组下标 b: 影片的数组下标

返回: 返回正数表示 $a > b$, 0表示 $a = b$, 负数表示 $a < b$

int data_compar_movtime(const void *a, const void *b)

功能: 比较电影的开始时间

参数: a: 影片的数组下标 b: 影片的数组下标

返回: 返回正数表示 $a > b$, 0表示 $a = b$, 负数表示 $a < b$

int data_compar_movprice(const void *a, const void *b)

功能: 比较电影的价格

参数: a: 影片的数组下标 b: 影片的数组下标

返回: 返回正数表示 $a > b$, 0表示 $a = b$, 负数表示 $a < b$

int data_compar_movremain(const void *a, const void *b)

功能：比较电影的剩余票数

参数：a: 影片的数组下标 b: 影片的数组下标

返回：返回正数表示 $a > b$, 0表示 $a = b$, 负数表示 $a < b$

int data_binarysearchmov_lower_bound(int index[], char key[])

功能：二分查找左边界

参数：index: 电影的下标数组, key: 搜索的关键字

返回：返回搜索到的数组左边界, -1表示没有搜索到

int data_searchmov_upper_bound(int index, char key[], int movieIndex[])

功能：查找右边界

参数：index: 左边界, key: 搜索的关键字, movieIndex: 影片的下标

返回：返回搜索到的右边界

void data_print_movielist(int movieIndex[], int num)

功能：打印传入的电影

参数：movieIndex: 传入的电影下标数组 num: 需要打印的电影个数

返回：无返回值

void data_print_movie(int index)

功能：打印一个电影

参数：index: 需要打印的电影信息下标

返回：无返回值

int user_search_filtertheater (int filtered_index[], char theater[], int num)

功能：按照电影院筛选电影

参数：filtered_index 需要筛选的电影的下标数组 theater: 筛选电影的关键字

返回：返回筛选出的电影个数

int user_search_filterlanguage(int filtered_index[], char language[], int num)

功能：按照语言筛选电影

参数：filtered_index 需要筛选的电影的下标数组 language: 筛选电影的关键字

返回：返回筛选出的电影个数

int user_search_filterdate(int filtered_index[], int month, int day, int num)

功能：按照日期筛选电影

参数：filtered_index 需要筛选的电影的下标数组 mon, day: 筛选电影的日期

返回：返回筛选出的电影个数

int user_search_filterhalltype(int filtered_index[], char halltype[], int num)

功能：按照电影院类型筛选电影

参数: filtered_index 需要筛选的电影的下标数组 halltype:筛选电影的影院名

返回: 返回筛选出的电影个数

char data_print_rootmenu(void)

功能: 打印管理员总菜单

参数: 无参数

返回: 选择的选项

void data_print_halllist(int hallIndex[], int num)

功能: 打印影厅列表

参数: hallIndex: 影厅下标 num: 需要打印的影厅个数

返回: 无返回值

void data_print_hall(hall *input)

功能: 打印一个影厅的信息

参数: input: 需要打印的影厅结构体的指针

返回: 无返回值

void admin_hall_createhall(int adminNO)

功能: 创建一个新影厅

参数: adminNO: 创建影厅的管理员在管理员数组中的下标

返回: 无返回值

void outputGraph(char size, char graph[])

功能: 打印一个座位图

参数: size: 影厅的大小 graph: 用one-hot结构存储的座位信息

返回: 无返回值

int user_purchase(int movNO, int userNO)

功能: 用户购买函数

参数: movNO:需要购买的电影索引 userNO:用户号

返回: 返回1表示购买成功, 0不成功

int data_isvalid_seat(char seats[], int chosen[6], char size, int num)

功能: 检测座位是否有效, 包括是否已经有人购买, 是否越界, 是否制造了一个空座位

参数: seats: one-hot座位图, chosen选择的座位, size:影厅大小 num:选择的座位个数

返回: 返回1表示有效, 返回0表示无效

int data_isvalid_phone(char phone[])

功能: 检测是否是有效的电话号码

参数: phone: 手机号码

返回: 返回1表示有效, 返回0表示无效

`void data_output2File()`

功能：将数组中的信息存储到文件当中

参数：无参数

返回：无返回值

`int user_purchase_search_ordernum(int day, int month, int orderNO[], int orderNum)`

功能：查找订单个数

参数：day, month : 日期, orderNO:订单, orderum:订单个数

返回：返回查找到的订单个数

`void data_input_fixedlength(int lower, int upper, char input[])`

功能：输入定长的字符串

参数：lower:下界 upper:上界 input:存储输入的字符串

返回：正数 $a > b$, 0表示 $a = b$,负数表示 $a < b$

`double data_input_double()`

功能：输入一个浮点数

参数：无参数

返回：返回读入的浮点数

`void user_signup()`

功能：注册函数

参数：无参数

返回：无返回值

`int data_isvalid_email(char* curEmail)`

功能：验证邮箱是否合法

参数：curEmail: 输入的邮箱

返回：返回1表示有效, 0表示无效

`void user_purchase_recommandseat(char seat[], char size, int seats, int recommand[])`

功能：推荐座位

参数：Seat: 座位信息, size: 影厅大小, seats: 需要推荐的座位数, recommand: 推荐的座位

返回：无返回值

`int data_absolute(int a, int b)`

功能：计算int类型的绝对值

参数：a, b: int值

返回：返回差的绝对值

`void user_search_filtermovie(int *num_filtered_date, int filtered_index_date[])`

功能：过滤电影

参数: num: 已经经过日期排序的 index: 下标数组

返回: 无返回值

int user_purchase_option (int filtered_index_date[], int num_filtered_date, int userNO)

功能: 提示用户购买

参数: index: 已经经过日期筛选的电影 num : 电影个数 userNO: 购买人在数组中的下标

返回: 返回1表示电影购买成功

void dateOptionFilter(int *num_filtered_date, int filtered_index_date[], int num_filtered)

功能: 选择是否需要改变查询电影的日期

参数: num: 已经经过日期筛选的电影 index: 已经经过日期筛选的电影

返回: 无返回值

int data_get_month()

功能: 获得月份信息

参数: 无参数

返回: 返回当前月份

int data_get_day()

功能: 获得日期

参数: 无参数

返回: 返回当前日期

void data_print_orderlist(int orderIndex[], int num)

功能: 打印订单信息

参数: orderIndex: 订单下标 num: 需要打印的订单个数

返回: 无返回值

void data_print_order(int orderNO)

功能: 打印订单信息

参数: 订单下标

返回: 无返回值

void user_information_modify(int userNO)

功能: 修改个人信息

参数: userNO: 当前的用户号码

返回: 无返回值

边界情况描述以及处理方法：

1. 输入整型数据时，先利用fgets读入存储到字符串中，再判断是否是一个有效的整型数据，最后判断是否超过int类型存储范围，如果都满足则将其通过atoi标准库函数转化为整型数据，不满足重新输入，以上流程集成在data_input_digit这一函数中
2. 输入浮点数时，先利用fgets读入存储到字符串中，再判断是否是一个有效的浮点数，如果都满足则将其通过atoi标准库函数转化为整型数据，不满足重新输入，以上流程集成在data_input_double这一函数中
3. 输入选项时，先利用fgets读入存储到字符串中，再判断是否超过一个字符，如果超过提示重新输入，再接着判断是否在给定的选项的范围内，如果不满足重新输入，满足则会返回该选项，以上流程集成在data_input_option这一函数中。
4. 输入小时和分钟时，也会检验是否是有效的时间，集成在data_input_min和data_input_hour这两个函数中。
5. 如果用户在搜索时没有输入关键字就直接回车，会提示用户没有输入，需要重新输入
6. 如果没有搜索到电影，会提示用户没有搜索到，询问是否需要重新搜索
7. 在订单购买的时候，如果进入预定的座位数大于剩余座位数，就会提示用户余票不足，选择返回搜索界面或者重新输入购票数目
8. 如果检测用户要购买的座位无效，就会提示重新输入
9. 在修改邮箱的时候，验证新邮箱的格式是否有效

Liangrun Da