

目录

[一、 问题重述 2](#_Toc28467359)

[二、 问题理解 2](#_Toc28467360)

[三、 理论分析 2](#_Toc28467361)

[1. 内容 3](#_Toc28467362)

[2. 功能 3](#_Toc28467363)

[3. 所包含的外部文件： 3](#_Toc28467364)

[4. 所涉及到的重点知识： 3](#_Toc28467365)

[四、 软件结构 4](#_Toc28467366)

[1. 函数展示 4](#_Toc28467367)

[2. 功能结构 4](#_Toc28467368)

[五、 函数介绍与流程图 5](#_Toc28467369)

[1. int read(Info guest[]); 5](#_Toc28467370)

[2. void meal\_regi(Info guest[],int guest\_number); 6](#_Toc28467371)

[3. void meal\_canc(Info guest[]); 7](#_Toc28467372)

[4. void meal\_inqu(Info guest[]); 8](#_Toc28467373)

[5. void meal\_chan(Info guest[]); 9](#_Toc28467374)

[六、 调试 10](#_Toc28467375)

[七、 心得体会 11](#_Toc28467376)

[八、 亮点 12](#_Toc28467377)

[1. 加入管理员功能 12](#_Toc28467378)

[2. 加入存储到数据库功能 12](#_Toc28467379)

[3. 防御性编程 12](#_Toc28467380)

[4. 交互式界面 13](#_Toc28467381)

[九、 执行过程 14](#_Toc28467382)

[1. 开始界面 14](#_Toc28467383)

[2. 功能一：订餐登记 14](#_Toc28467384)

[3. 功能二：取消订餐 15](#_Toc28467385)

[4. 功能三：查询订餐信息 17](#_Toc28467386)

[5. 功能四：更改订餐信息 18](#_Toc28467387)

[6. 功能五：退出本订餐系统 20](#_Toc28467388)

[7. 功能六：使用管理员权限 20](#_Toc28467389)

[8. 功能七：信息写入到数据库 21](#_Toc28467390)

[十、 源代码 22](#_Toc28467391)

[1. main.c 22](#_Toc28467392)

[2. struct\_info 23](#_Toc28467393)

[3. meal\_tele\_func.h 23](#_Toc28467394)

[4. meal\_tele\_fun.c 24](#_Toc28467395)

[5. sqlite3\_experiment.c 33](#_Toc28467396)

# 问题重述

电话订餐系统

系统包括：姓名，电话，人数，用餐时间，用餐喜好，要求：

（1） 订餐用户登记，给每个订餐用户编号

（2） 取消订餐

（3） 更改订餐时间

（4） 更改订餐人数

（5） 查询订餐情况，根据姓名查询，根据电话查询

（6） 将数据存入文件

# 问题理解

近几年来，随着社会科技不断进步，人民生活水平也在不断提高，而人们在餐饮消费方面的费用投入不断增加。这都刺激着餐饮行业的飞速发展，并激烈竞争。但与需求不断增加所不匹配的是，目前传统餐饮企业还是使用着效率低下的人工管理方式。从菜单设计、餐桌使用管理，到顾客信息登记都还是靠人工纸笔操作。这样不仅会耗费时间和人力成本，还会出现数据错误的现象，造成顾客在消费美食时的满意度下降，直接影响到美食餐饮企业的效益下降。同时，用户的电话订餐登记也大同如此，让人感到难以满意。

为了应对这样的需求不匹配现象，企业急需引进更先进的手段，使用精准而且高效的计算机系统来辅助管理。通过计算机系统来管理餐饮企业的日常运作，给管理者功能辅助的模型，给顾客方便快捷的点餐模型，以此来不断提高订餐效率，完善美食体验。

本系统使用C语言为开发工具，系统完成订餐用户登记，给每个订餐用户编号，取消订餐，更改订餐时间，更改订餐人数，查询订餐情况，根据姓名查询，根据电话查询和将数据存入文件的功能。同时，开发者（也就是作者我）加入了管理员和存入到数据库功能，使得本系统功能更加完备，适用范围更广。

对于具体实现，用结构体存储将方便管理，方便对数据进行整块操作，当结合数组或者链表后将可以表示一定数目的顾客。同时，应该将实现功能的部分在函数中实现，在主函数中仅表示出主要函数的调用，以使代码结构清晰，可重用性提高。

本系统开发需要的知识大体包含：数组的操作，结构体的操作，字符串的操作，文件流的操作，数据库的操作等。

# 理论分析

从问题中提取关键词：姓名，电话，人数，用餐时间，用餐喜好，登记，编号，取消，更改时间，更改人数，查询，存入文件。结合已经掌握的知识和本身程序经验，可以看出本题大概的思路框架。

## **内容**

对于系统包括的内容：姓名，电话，人数，用餐时间，用餐喜好。用结构体存储将方便管理，方便对数据进行整块操作，当结合数组或者链表后将可以表示一定数目的顾客。

## 功能

由于实现功能较多，并且某些功能本质上具有相似性，例如更改订餐时间，更改订餐人数，所以，应该将实现功能的部分在函数中实现，在主函数中仅表示出主要函数的调用，以使代码结构清晰，可重用性提高。

为各个功能编写函数，此处列举表格如下：

|  |  |
| --- | --- |
| 功能 | 函数 |
| 订餐登记 | void meal\_regi(Info guest[],int guest\_number); |
| 取消订餐 | void meal\_canc(Info guest[]); |
| 更改信息 | void meal\_chan(Info guest[]); |
| 查询信息 | void meal\_inqu(Info guest[]); |
| 管理员功能（附加） | void admin(Info guest[]); |
| 存到数据库（附加） | int exp(); |

函数间的调用关系将在后面叙述，[点此跳转](#_软件结构)。

## 所包含的外部文件：

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "sqlite3.h"

#include "sqlite3.c"

## 所涉及到的重点知识：

数组的操作

结构体的操作，包括：typedef, struct等

字符串的操作，包括：strcmp, strcpy等

文件流的操作，包括：fopen, fclose, fseek, fprintf. fscanf等

数据库的操作，包括：sqlite3, sqlite3\_open, sqlite3\_free, sqlite3\_exec, sqlite3\_close等

# 软件结构

## 函数展示

由于本程序主函数简练，高度函数化，因此函数功能和结构略微复杂，现将函数等相关信息用表格列举如下，再做详细介绍：

|  |  |
| --- | --- |
| 函数原型 | 调用的函数(自编写) |
| void interface() |  |
| void print(Info guest) |  |
| void print\_all(Info guest[]); |  |
| void dele\_gue(Info guest[],int guest\_number) |  |
| [void meal\_regi(Info guest[],int guest\_number)](#_void_meal_regi(Info_guest[],int) |  |
| void meal\_rewr(Info guest[]) |  |
| [int main(int argc, char \*argv[])](#_功能结构) | admin, read, interface, meal\_ragi,  meal\_canc, meal\_inqu , meal\_chan, exp |
| int compare(Info guest[]) | print |
| [int read(Info guest[])](#_int_read(Info_guest[]);) | dele\_gue, read |
| void meal\_canc\_option(Info guest[],int i) | interface, meal\_canc, dele\_gue |
| [void meal\_canc(Info guest[])](#_void_meal_canc(Info_guest[]);) | meal\_canc\_option, compare |
| [void meal\_inqu(Info guest[])](#_void_meal_inqu(Info_guest[]);) | print |
| [void meal\_chan(Info guest[])](#_void_meal_chan(Info_guest[]);) | meal\_rewr, compare, meal\_chan |
| void admin(Info guest[]) | admin\_opti |
| void admin\_opti(Info guest[]) | guest\_dele, print\_all, admin\_opti |
| int exp() | read |

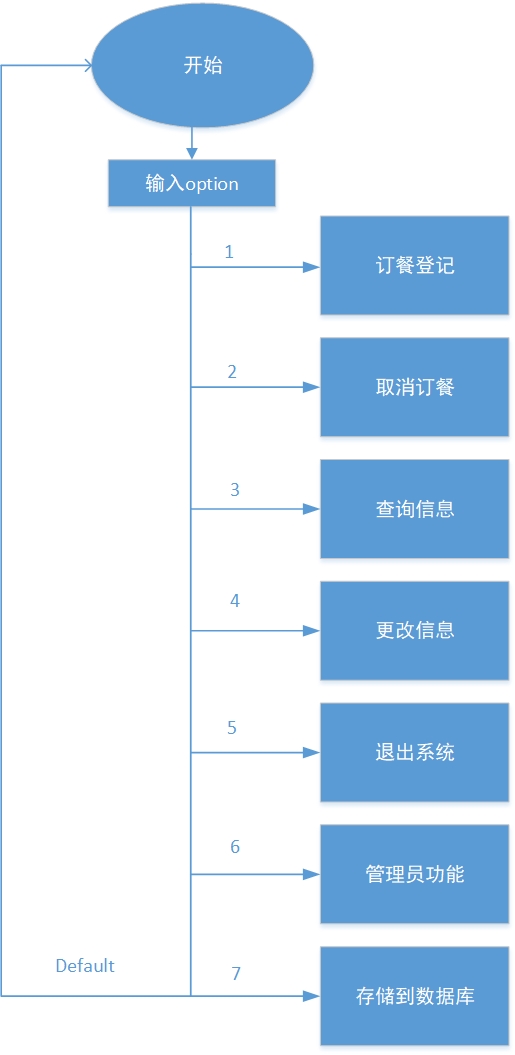
（按住Ctrl并点击链接可转到详细解释部分）

## 功能结构

int main(int argc, char \*argv[])

功能：仅用来调用函数，显示所使用的功能

结构：

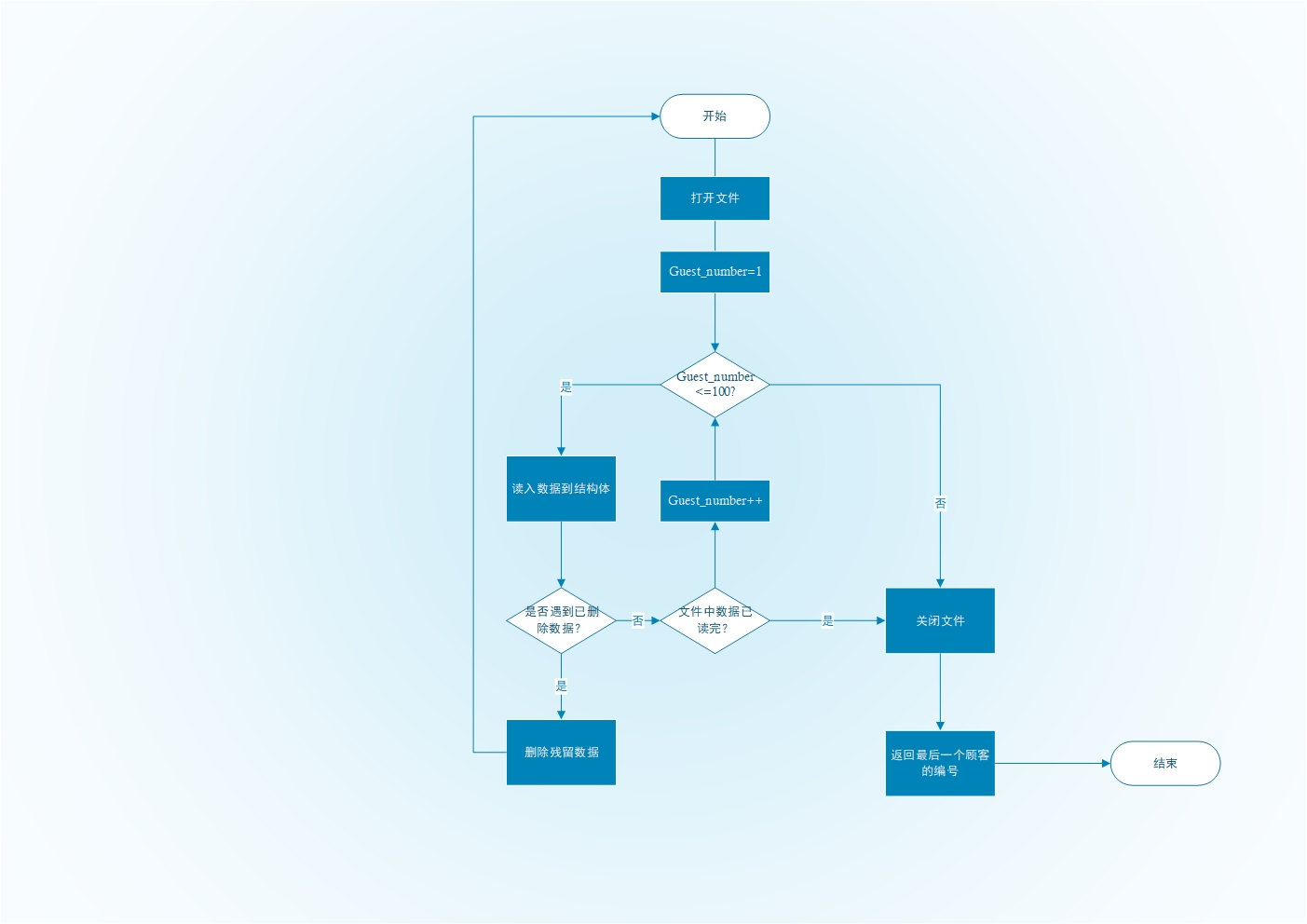


# 函数介绍与流程图

## int read(Info guest[]);

功能：将信息从txt文件中读入到结构体，供其他函数使用

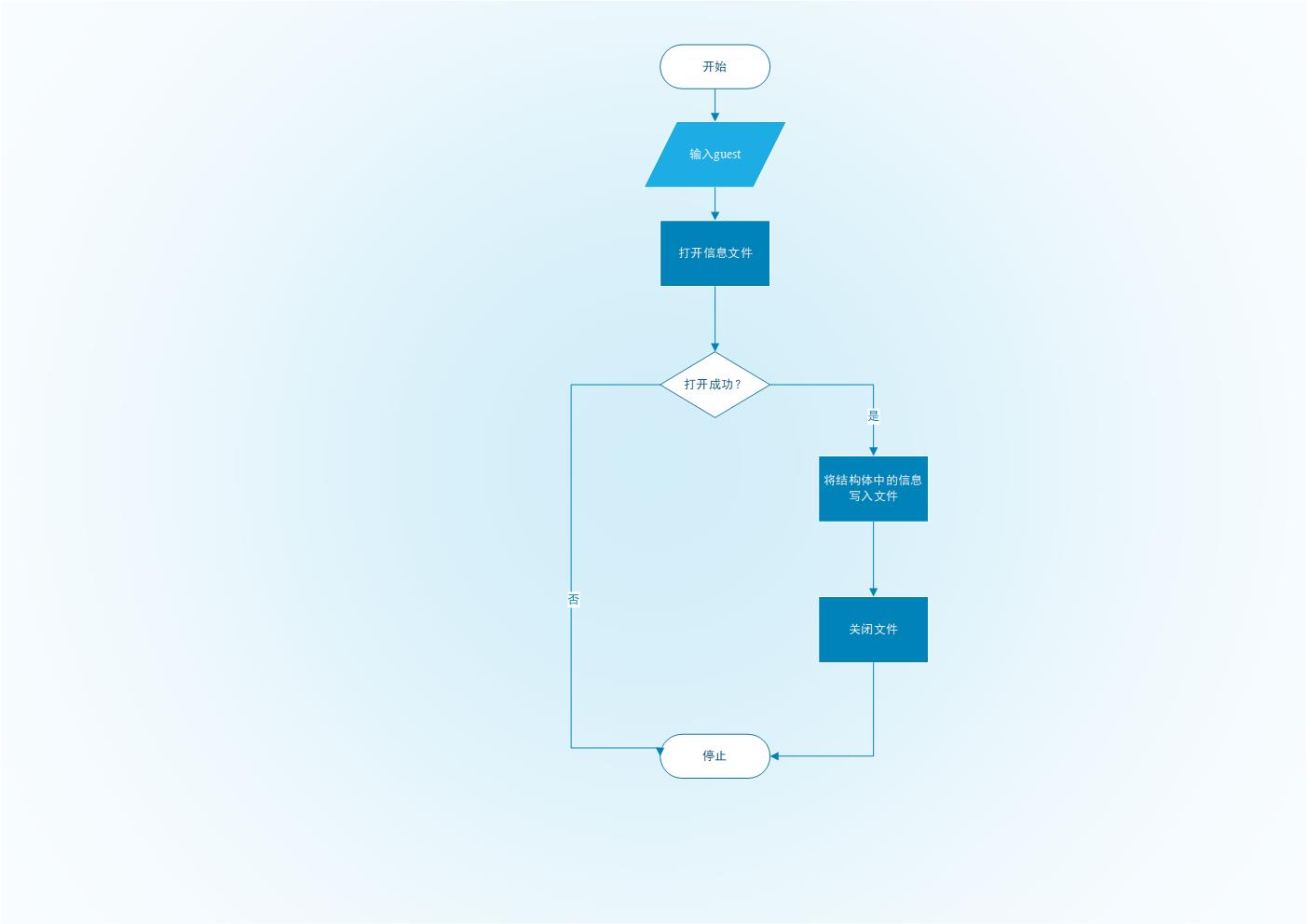
结构：



## void meal\_regi(Info guest[],int guest\_number);

功能：实现订餐登记功能

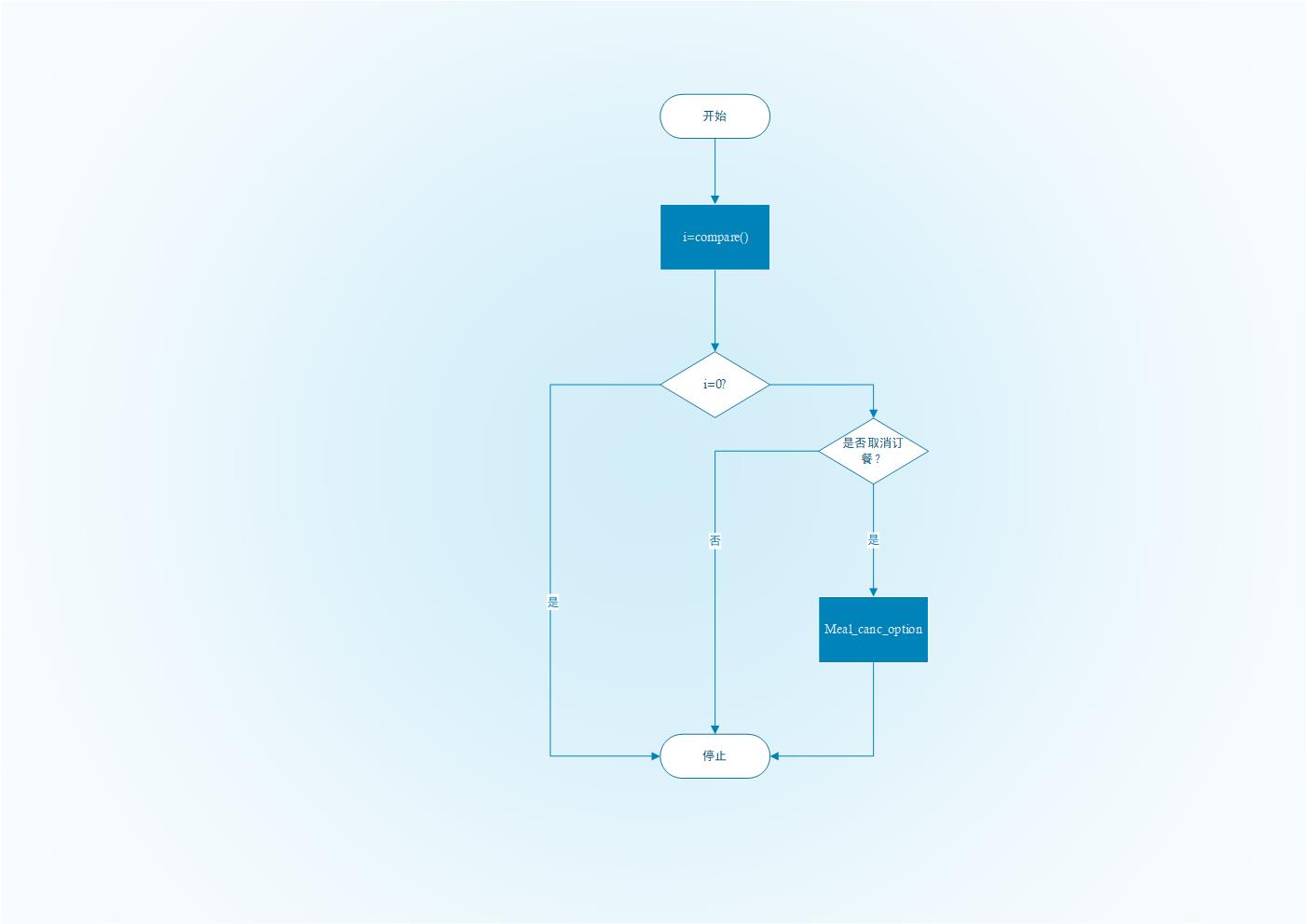
结构：



## void meal\_canc(Info guest[]);

功能：实现订餐取消功能

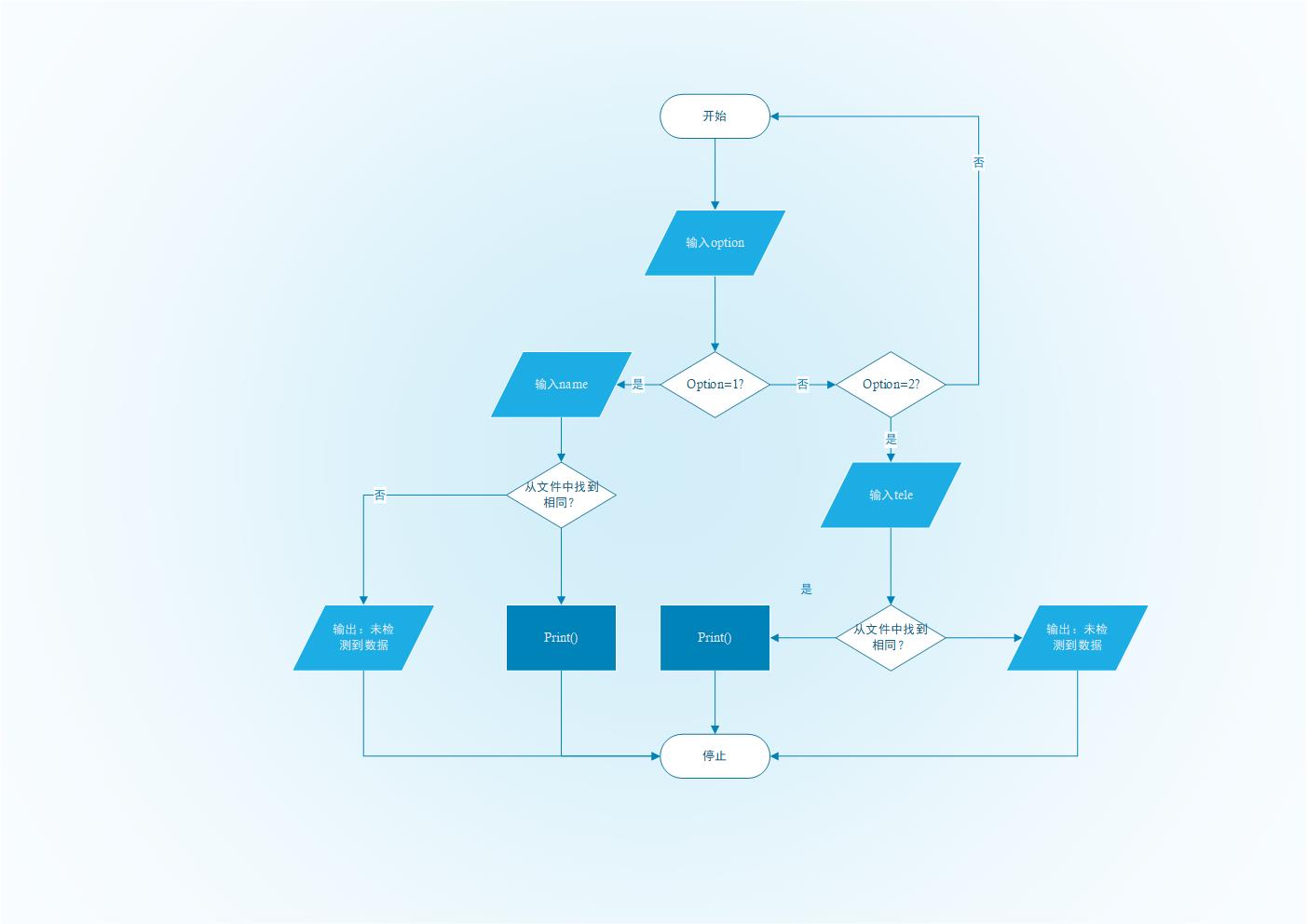
结构：



## void meal\_inqu(Info guest[]);

功能：实现订餐查询功能

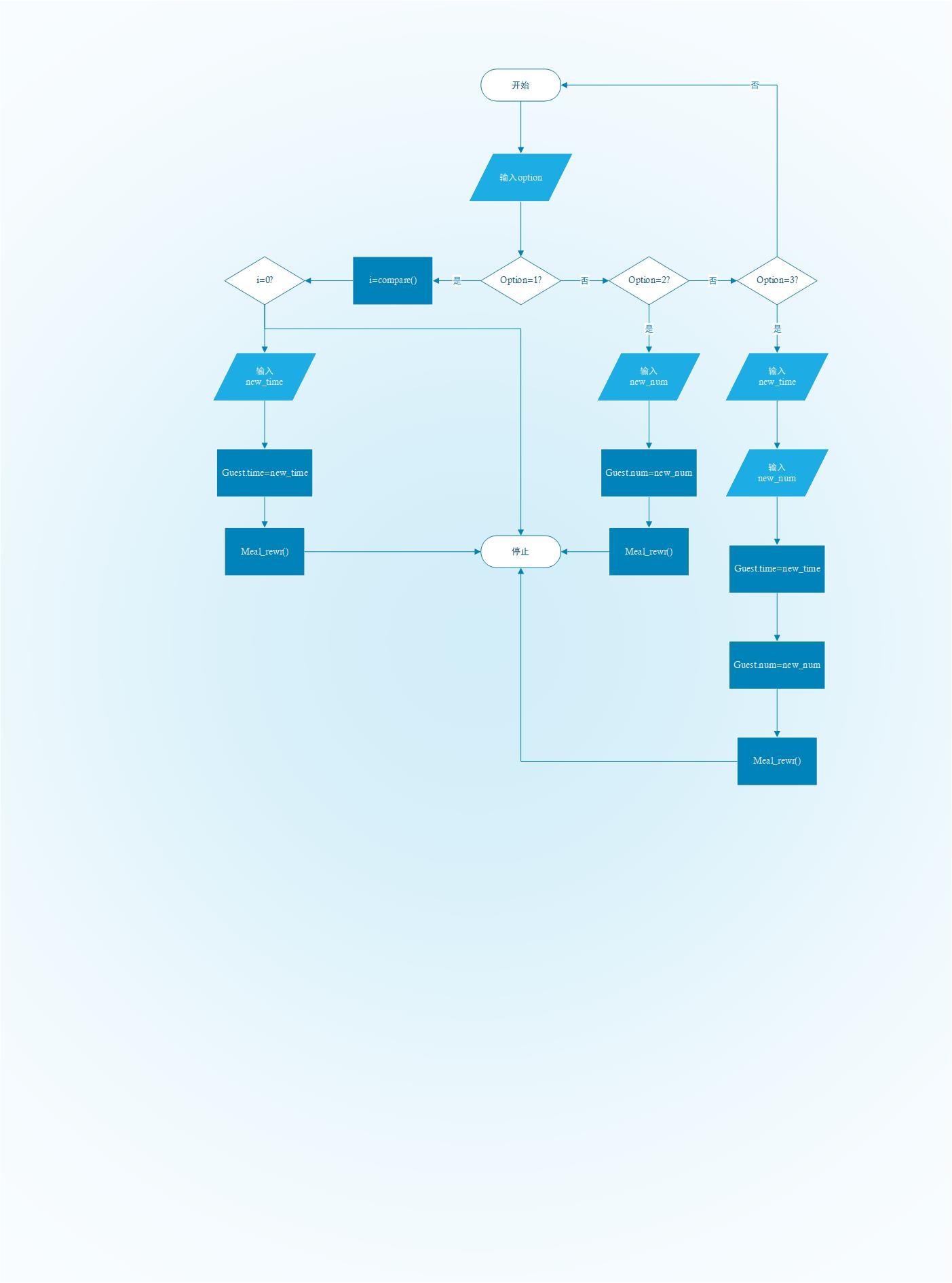
结构：



## void meal\_chan(Info guest[]);

功能：实现更改信息功能

结构：



# 调试

思考软件结构与代码思路上慢慢摸索，解决问题，在实践中发现自己知识结构中的问题，从各种途径获取所需要的知识，此过程实为任务驱动，此处不一一赘述。代码所遇bug成百上千，也不一一赘述，仅列举部分如下：

SQL error: no such table: mystruct

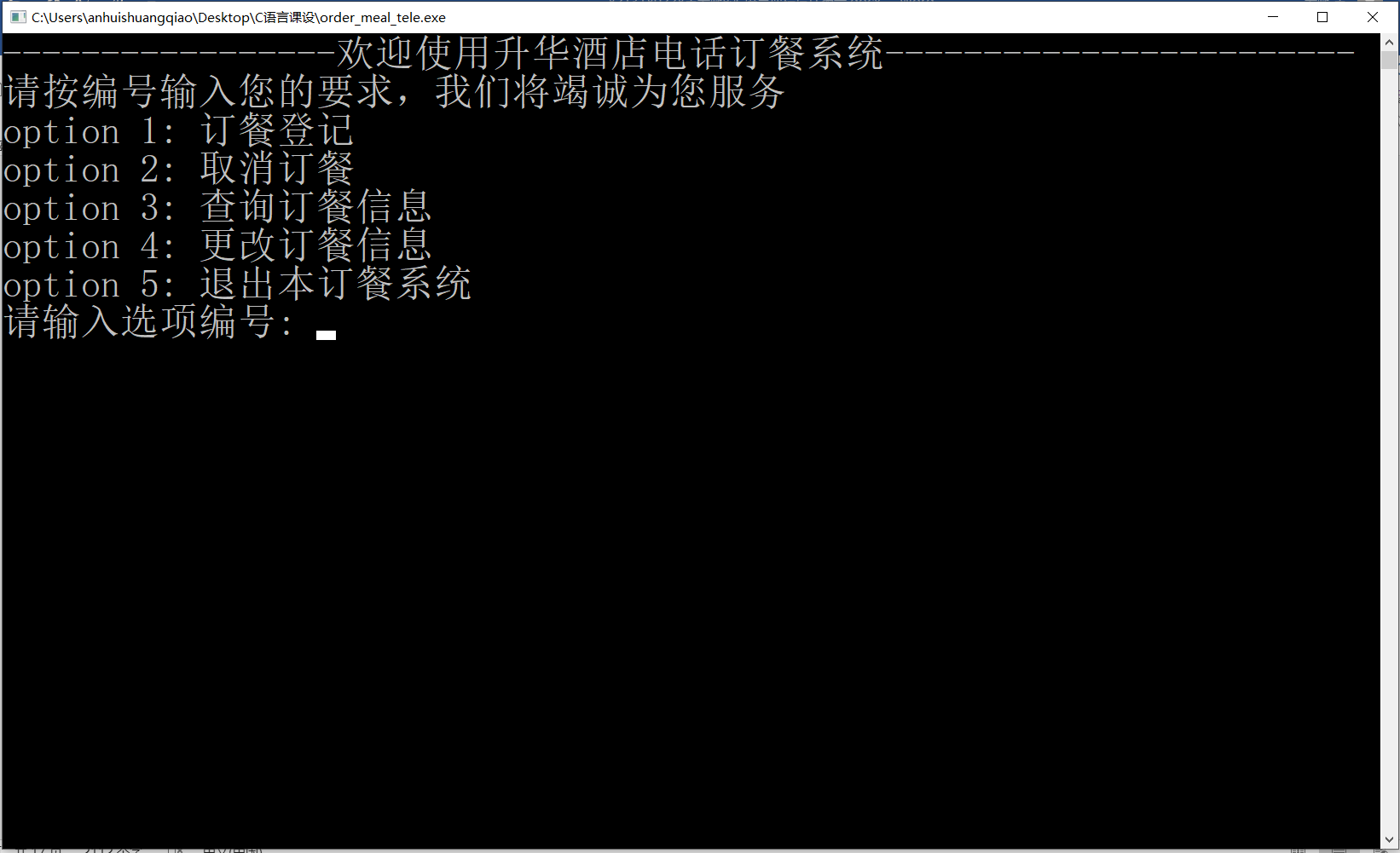
SQL error: near "test": syntax error

[Error] expected declaration or statement at end of input

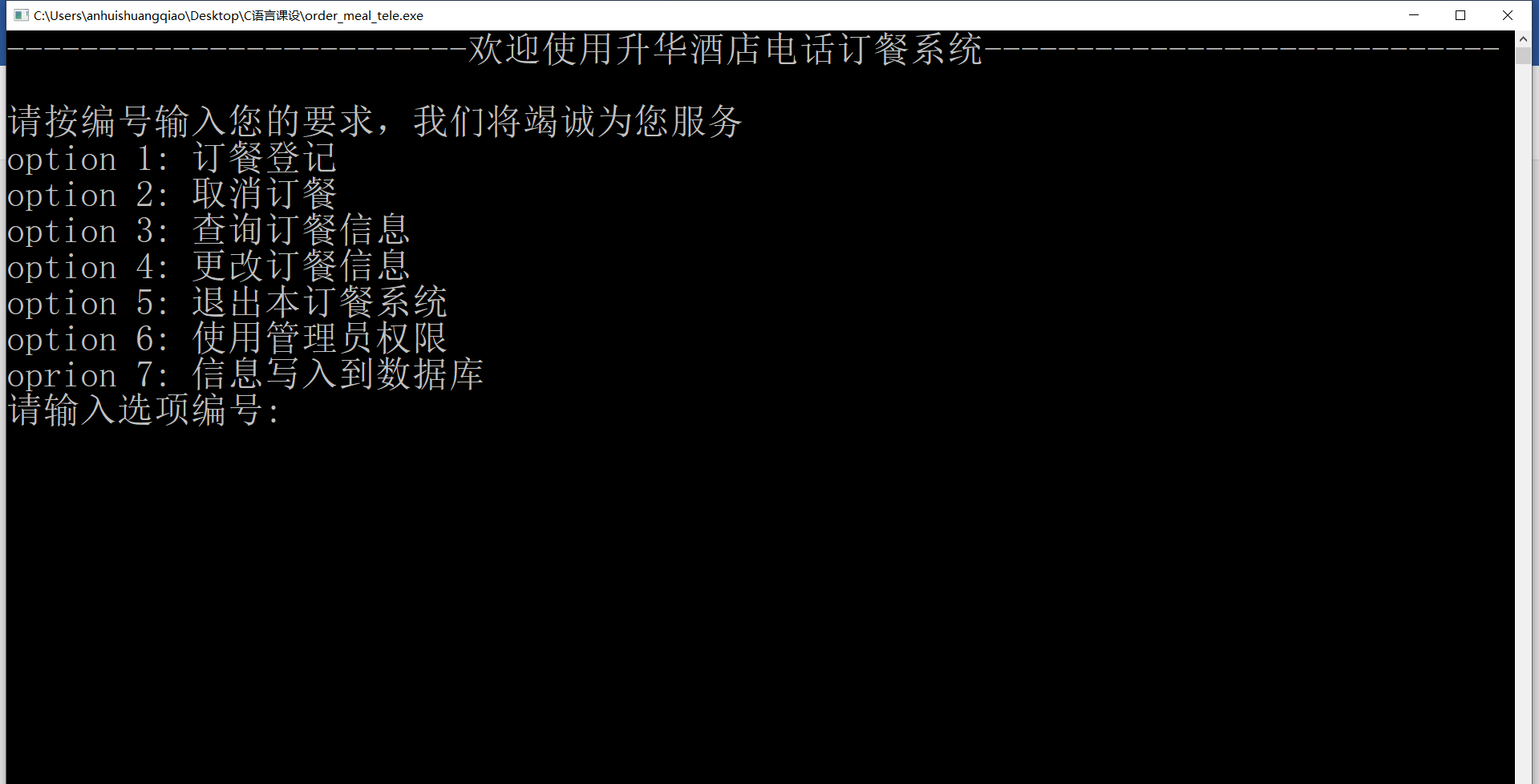
在经过头悬梁，锥刺股，凿壁偷光，萤囊映雪，批阅十载，增删五次之后，终于使程序运行完美。呈现给C语言老师看，询问意见，老师建议我加入管理员功能，优化输出所有数据时的界面，同时强调希望我实现sqlite数据库功能。

所以，此处呈现改动前后的界面和功能。

改动前，仅含有要求中的五个功能，呈现如下：



改动后，除了包含要求中的五个功能外，加入了 功能六：管理员选项、功能七：存储到数据库，并且显示出较好的运行效果，呈现如下：



# 心得体会

虽然已不是第一次做课程设计，但对于要求功能完备的DOS界面设计倒还是头一次，这次的课程设计是一次既让人痛苦又让人开心的事情。开始的时候不是很清楚课程设计要设计什么，不知道到底怎样设计，因为平时所写的c程序都是一些简单而且基本的C程序，往往只是针对某个特殊的问题来解答，根本不是解决实际情况，也不是符合顾客要求的工程程序，所以觉得无从下手，觉得很困难，甚至没有一个思路，而且对于组建工程，构建模块等工作无法理解。这些情况都使自己陷入了苦恼的深渊。

后来，从最基本的看起，从《C语言程序设计现代方法》开始，一点点了解怎样组建工程，怎么样构建模块，对于每个模块应该怎么样进行设计，逻辑思维是什么，然后开始构建。

在设计过程中一直都有错误存在，困难重重。遇到困难就开始用大量的时间进行修改以及整理，理出思路。最终将工程调试成功，添加模块成功，而这些成功的瞬间是最快乐最开心的时刻。

在编写代码的过程中，常遇到各种bug，语法错误的相对来说容易修正，但碰到偶尔什么稀奇古怪的error，是在让人摸不着头脑，在求助于网络之后，绝大部分的问题都可解决。偶尔的一个符号缺少，会让bug的error提示出现在另外的地方，甚至使.o文件中，这个时候就要静下心来，从最开始的地方看起，慢慢寻找bug，只要多进行输出，终究会发现错误的根源。这个时候不得不感叹DEV-C的不方便之处，matlab在显示变量的值时极为方便清楚，容易发现错误，方便debug。

思考软件结构与代码思路也需要慢慢摸索，解决问题，在实践中发现自己知识结构中的问题，从各种途径获取所需要的知识，此过程实为任务驱动，让人获益匪浅。

在经过头悬梁，锥刺股，凿壁偷光，萤囊映雪，批阅十载，增删五次之后，终于使程序运行完美。呈现给C语言老师看，询问意见，老师建议我加入管理员功能，优化输出所有数据时的界面，同时强调希望我实现sqlite数据库功能。经过修改之后，在展示过程中得到了检查人员的肯定。

总的来说，通过这次课程设计我学会了很多，觉得自己设计的东西更加接近实际的程序要求，获益匪浅。

# 亮点

## 加入管理员功能

考虑到商家可能会对顾客信息进行管理，特加入管理员功能。

[按住Ctrl点此跳转到本文档执行过程部分查看。](#_功能六：使用管理员权限)

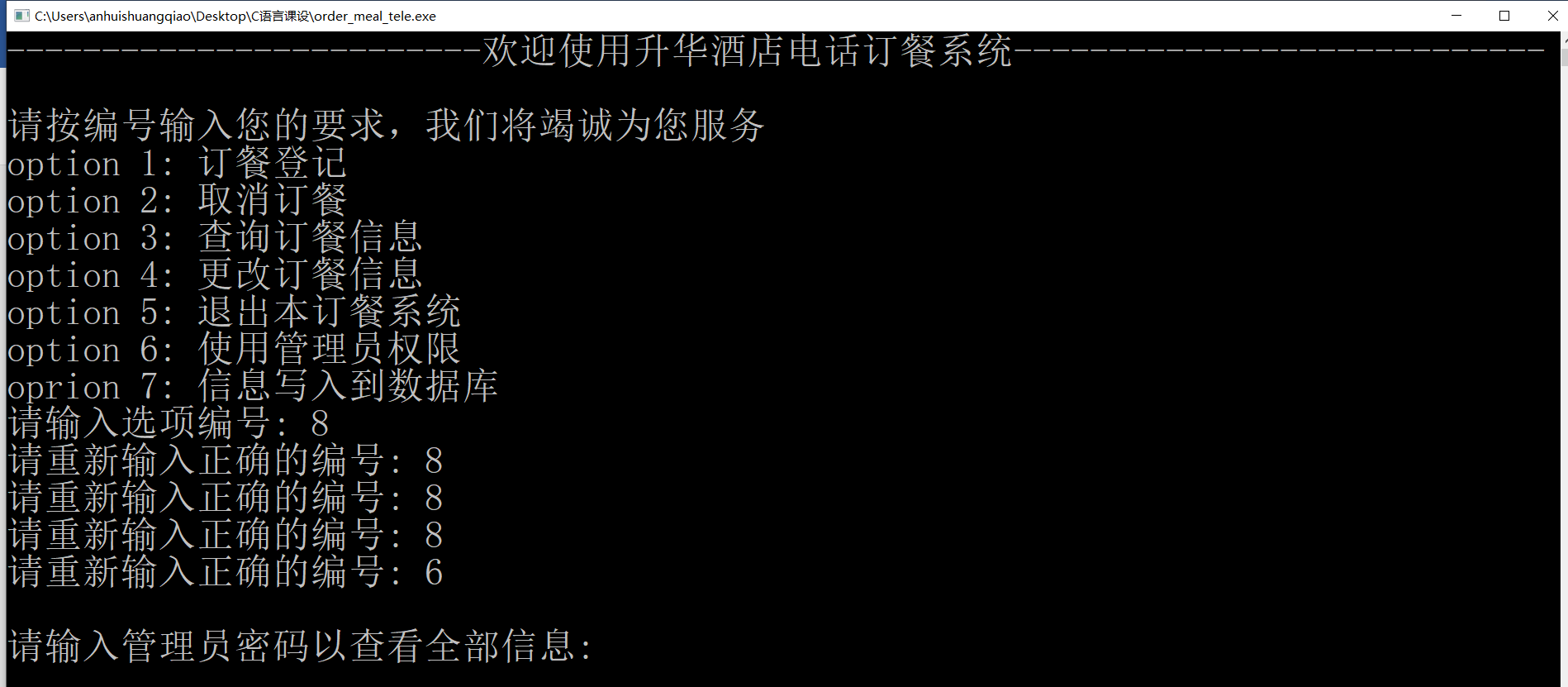
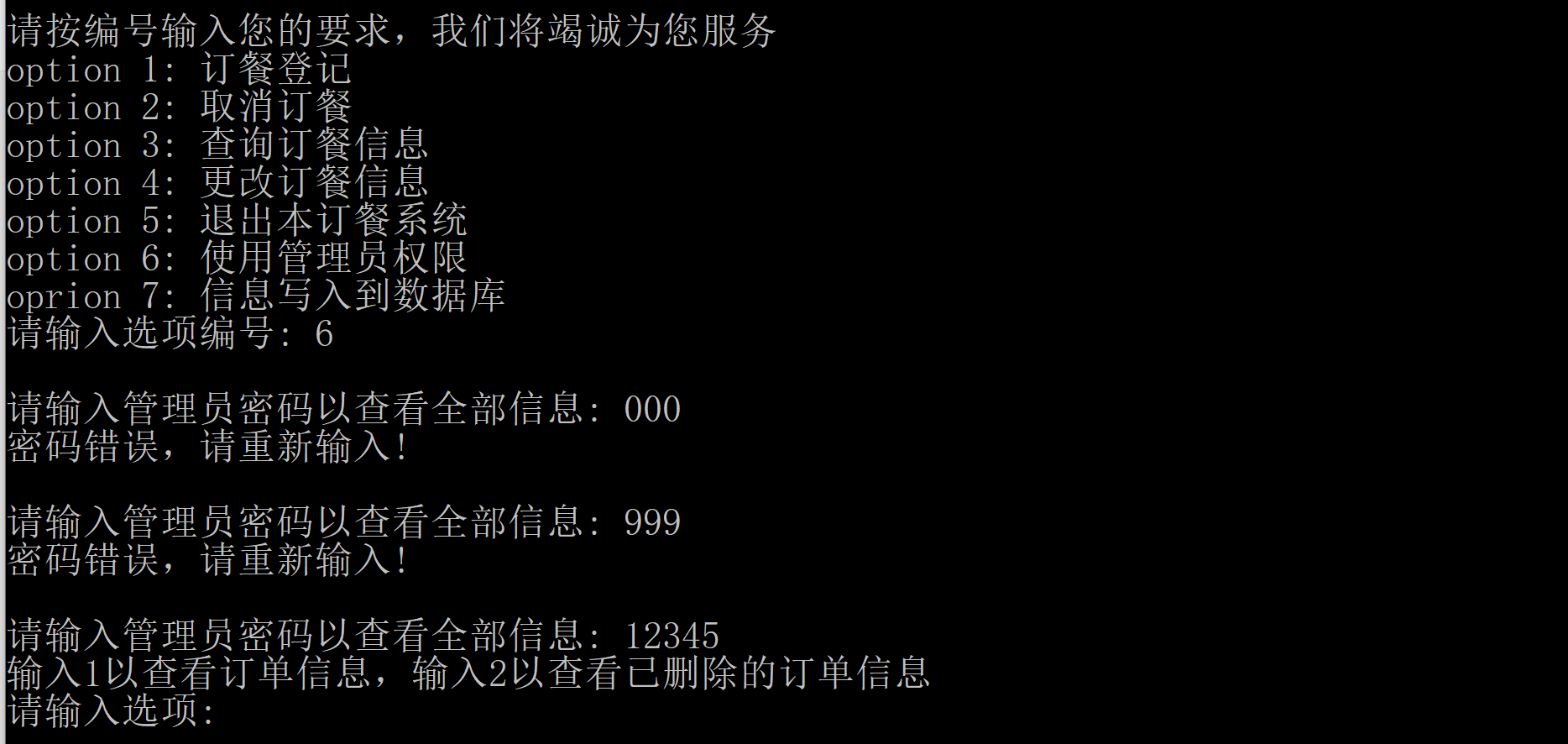
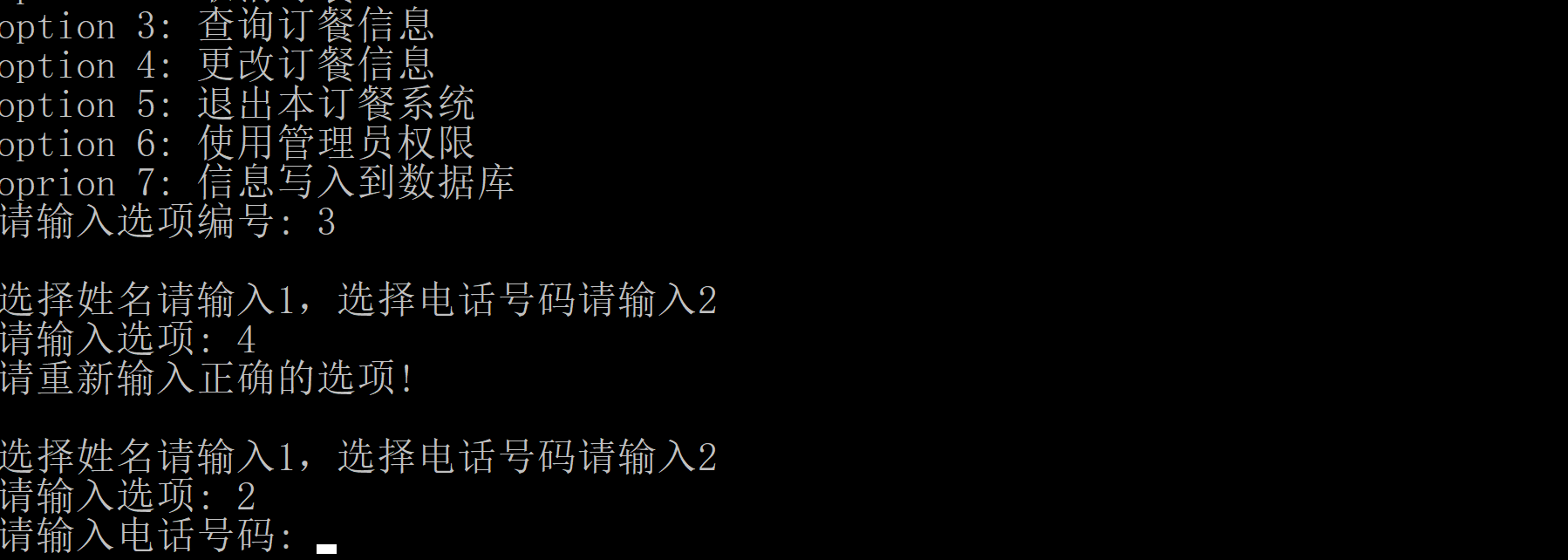
## 加入存储到数据库功能

为方便数据管理，特加入“存储到数据库功能”。

[按住Ctrl点此跳转到本文档执行过程部分查看。](#_功能七：信息写入到数据库)

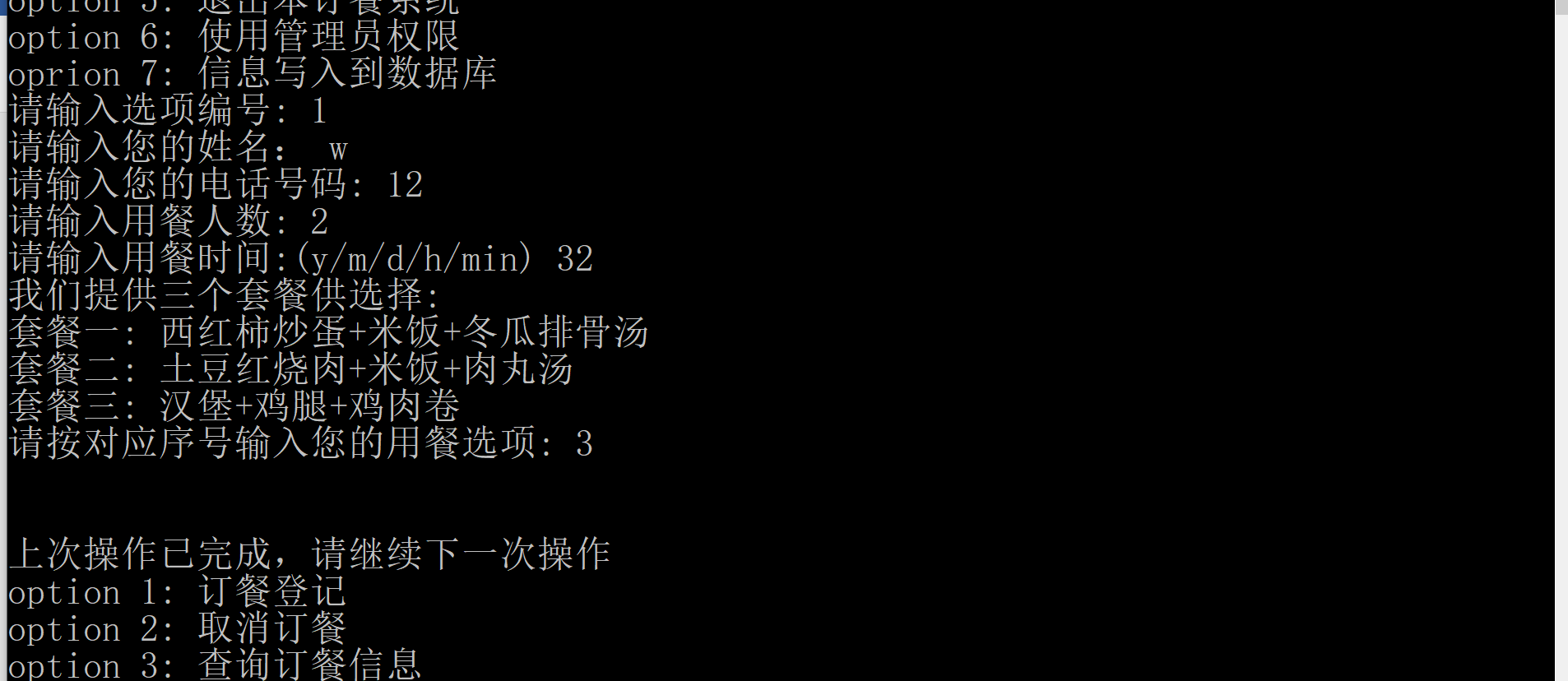
## 防御性编程

用户输入错误选项后将得到提示，并重新输入，列举几项如下。

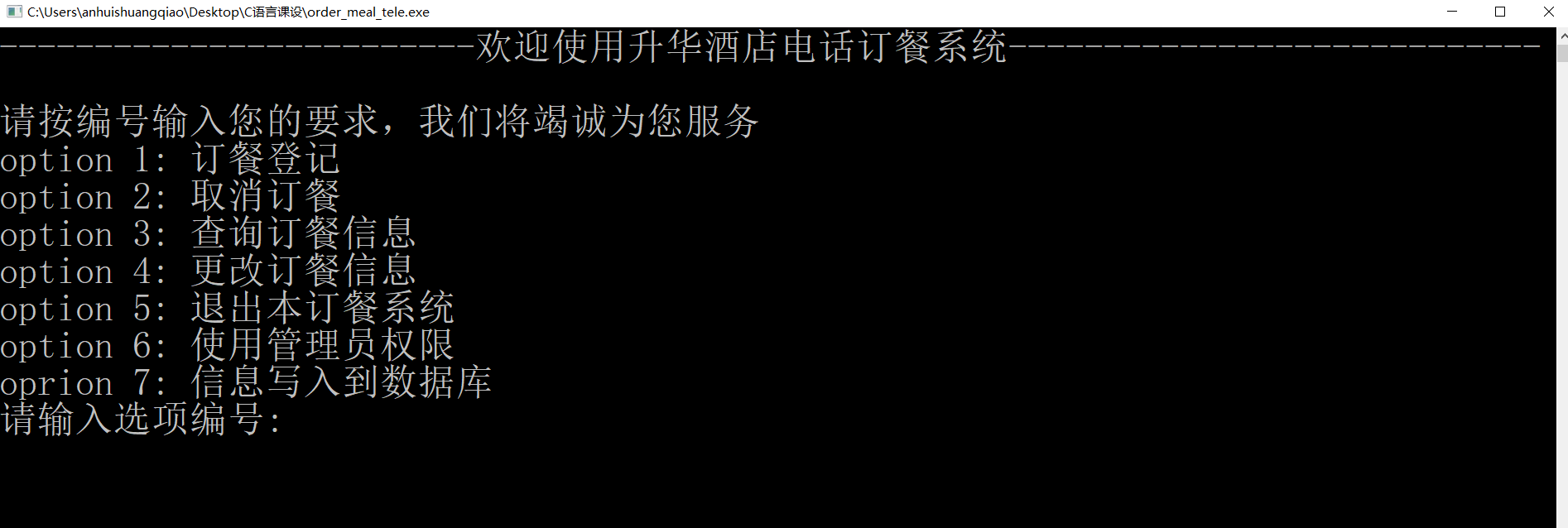
## 交互式界面

为改善用户体验，特让当前用户之前输入的痕迹留存，方便查看。



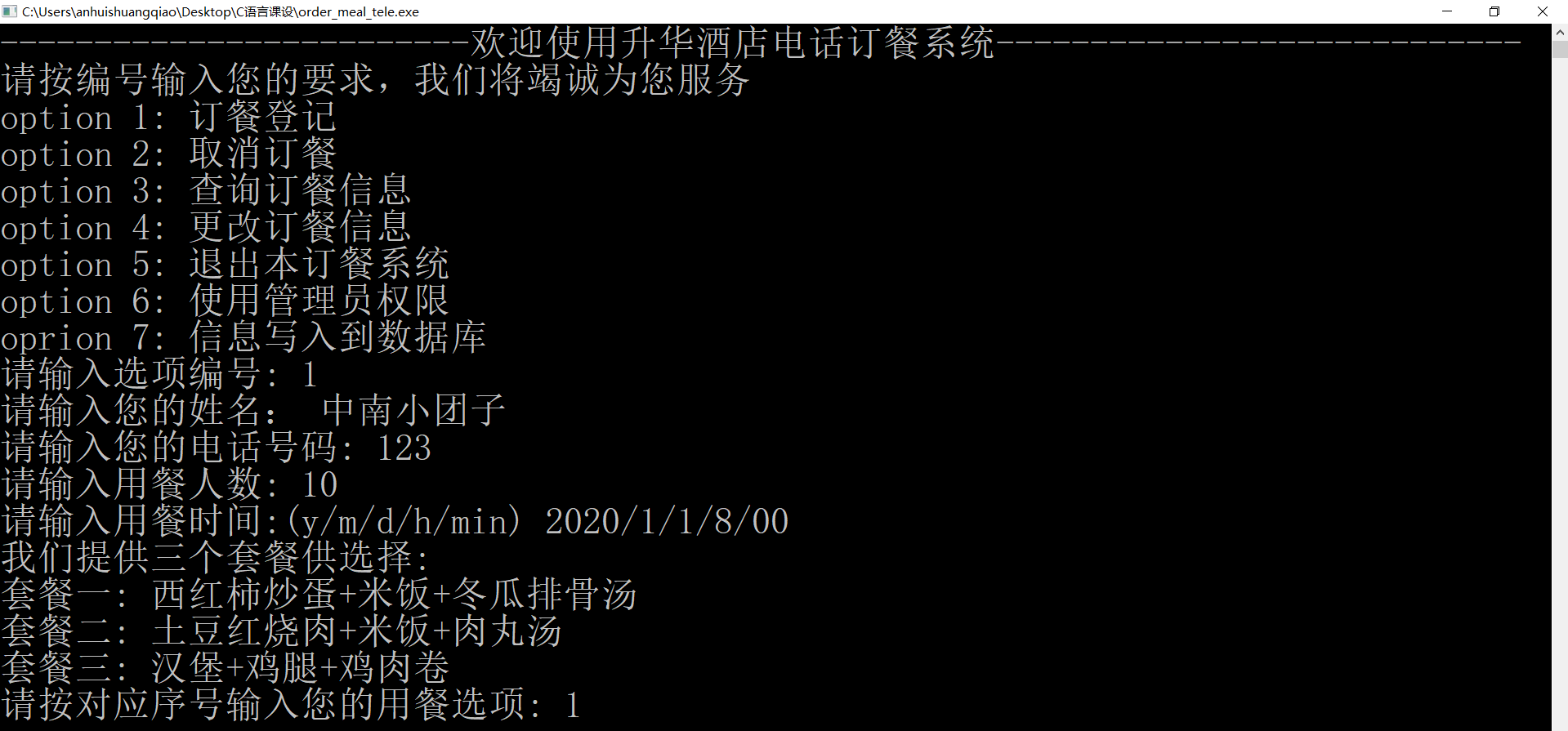
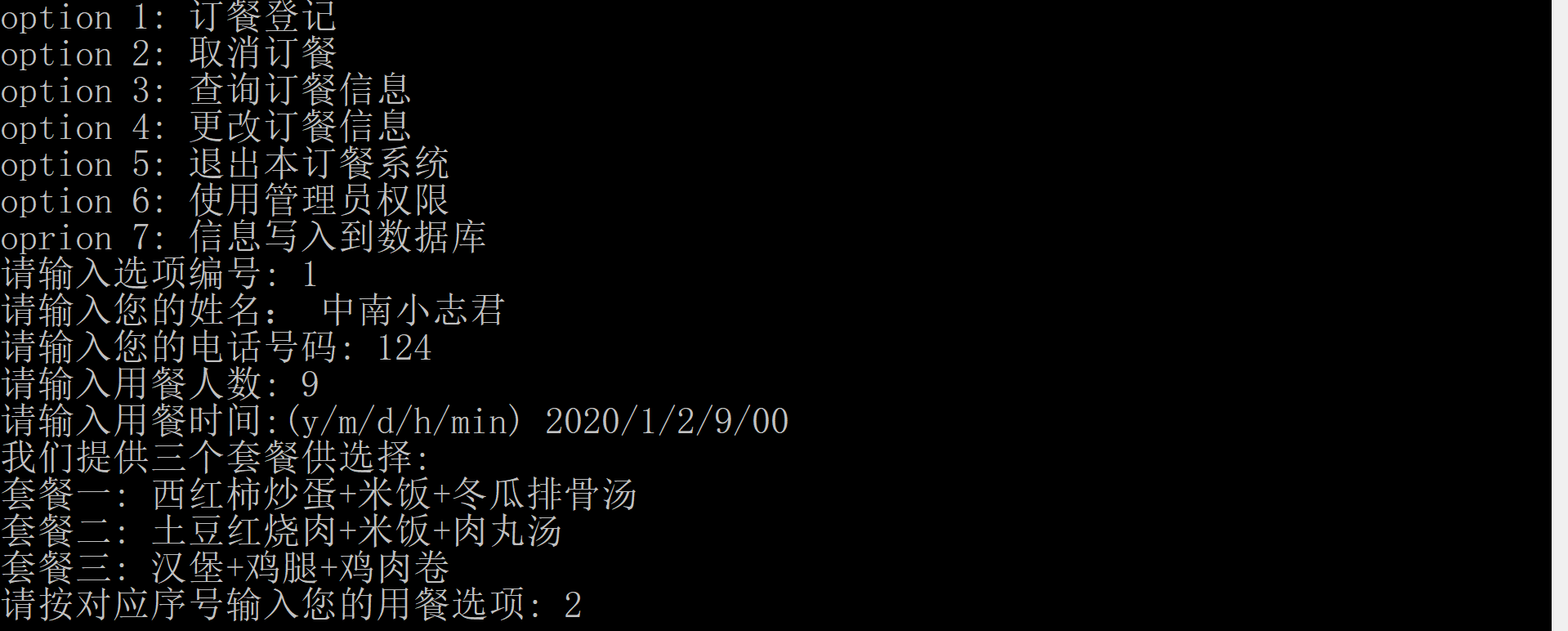
# 执行过程

## 开始界面

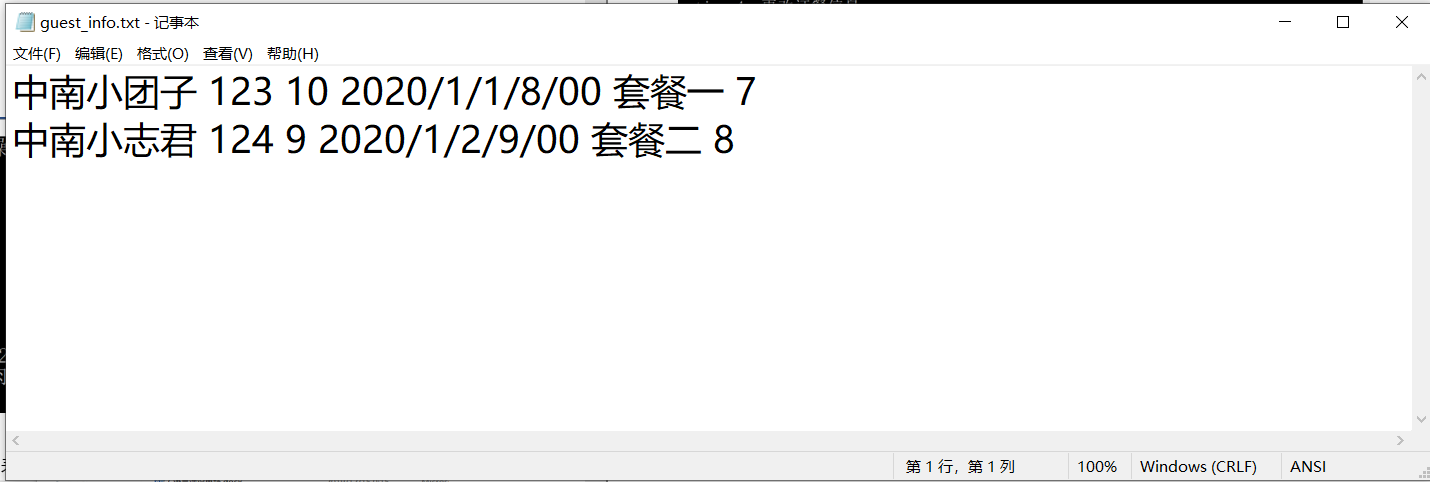


## 功能一：订餐登记

根据提示输入信息（下图为“中南小团子”、“中南小志君”分别订餐）

信息被存入TXT文件，在TXT文件中表示如下（订单编号即为每行最后一列数字）



## 功能二：取消订餐

取消订餐有三个功能选项，分别为

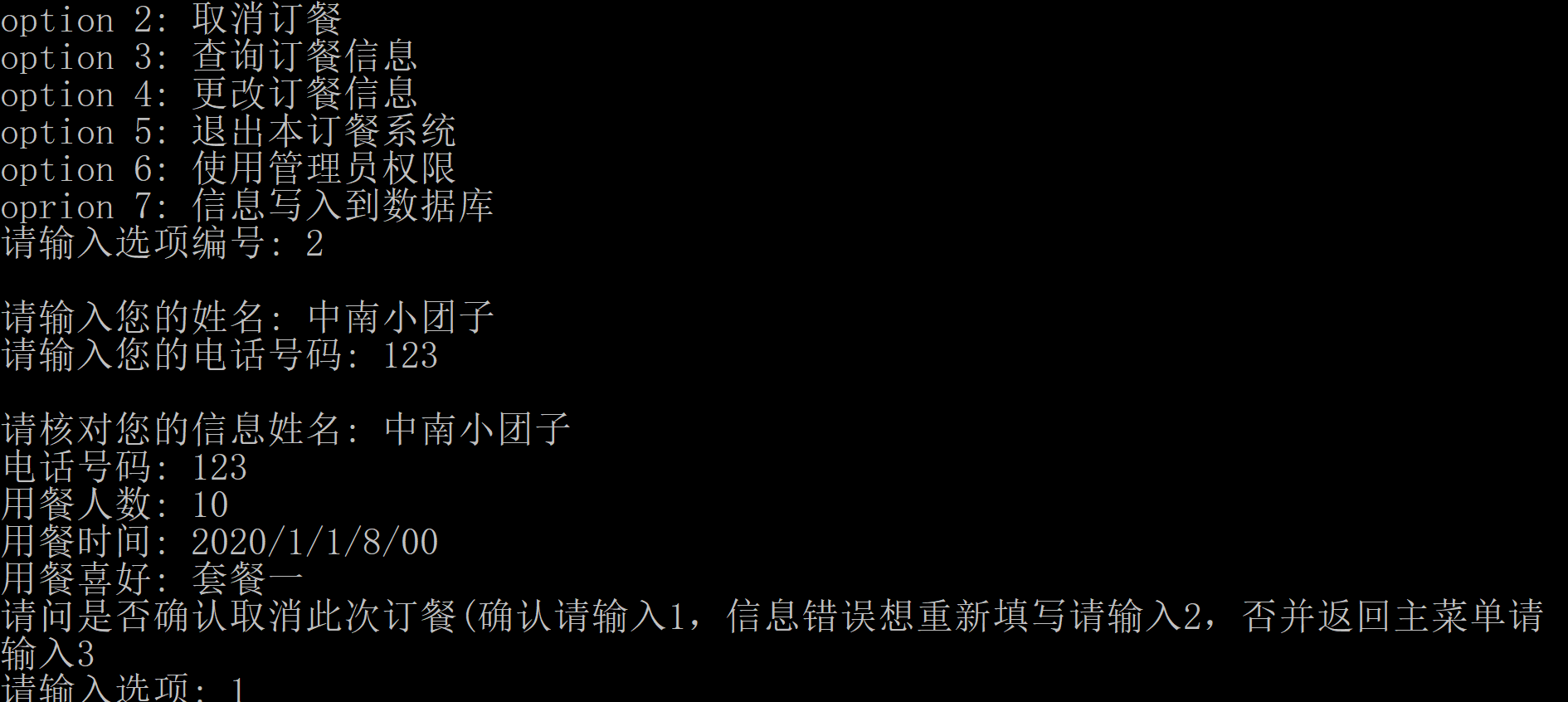
1、 确认取消订餐

2、 拒绝取消订餐，返回主菜单

截图如下

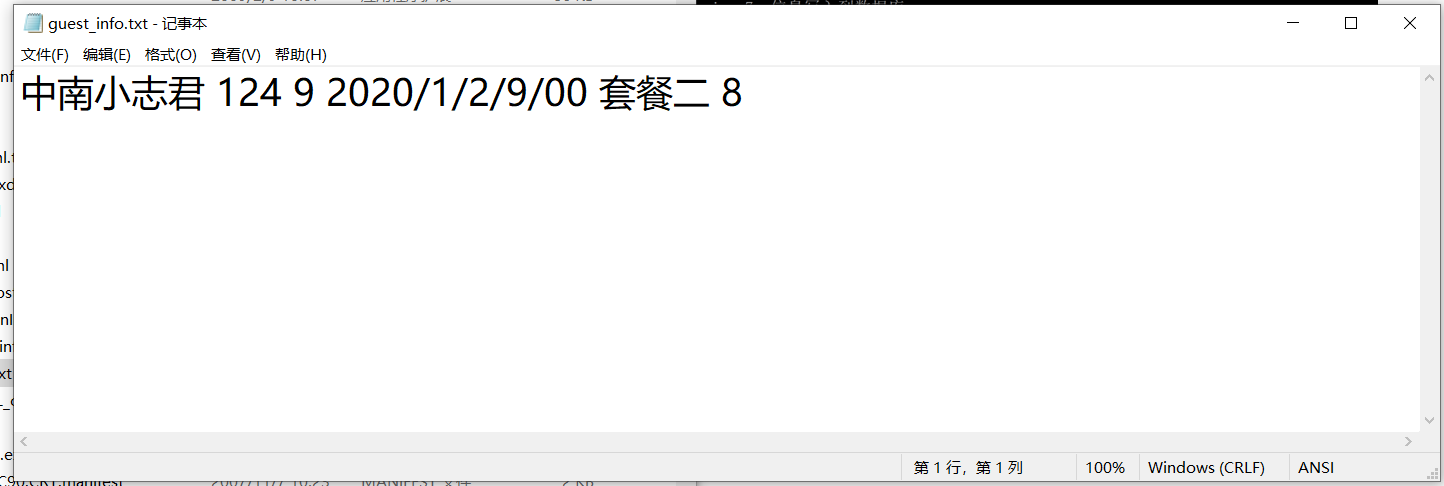
1、 确认取消订餐（下图为删除“中南小团子”的订餐）

检测到用户信息，提示是否确认

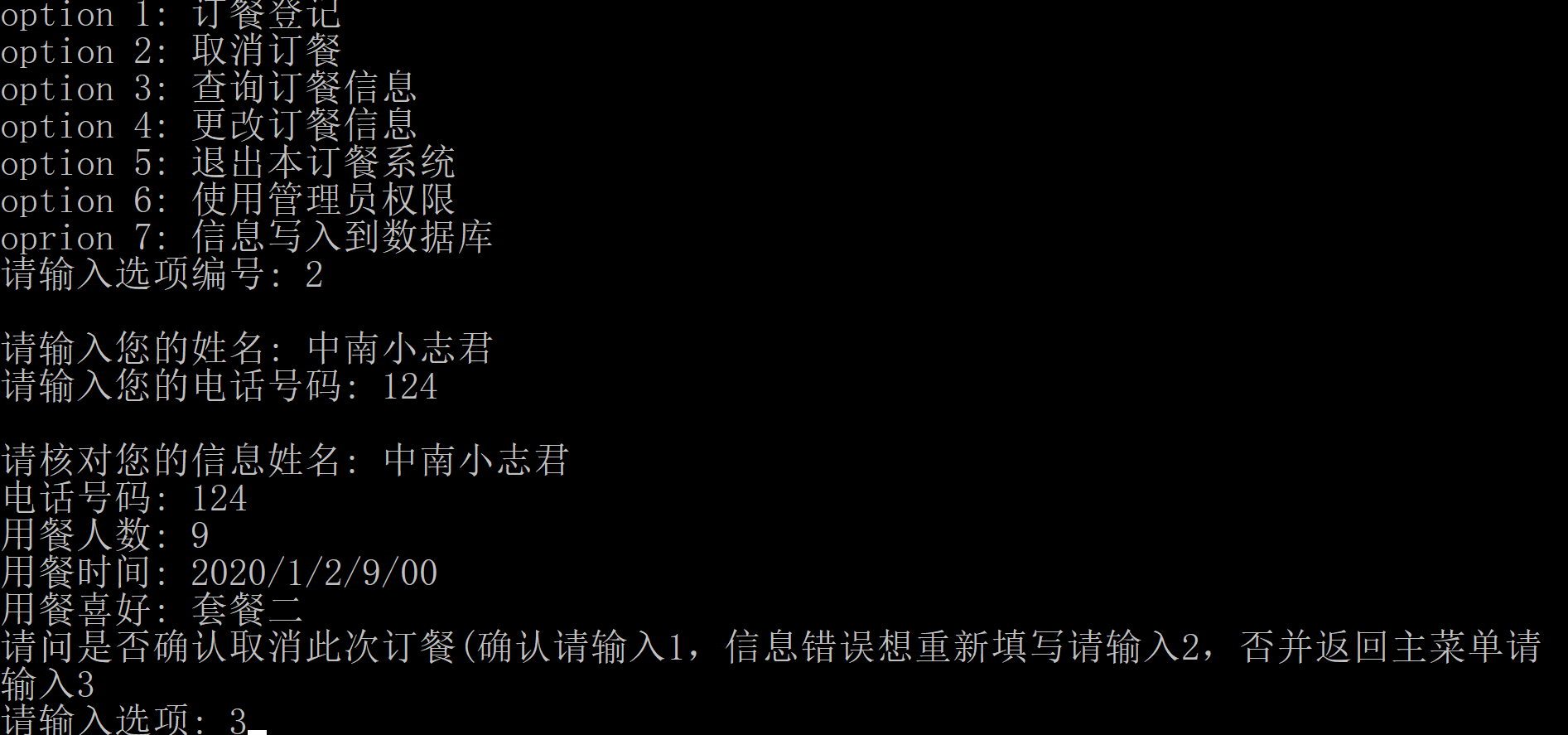


确认删除后，在TXT文件中表现如下

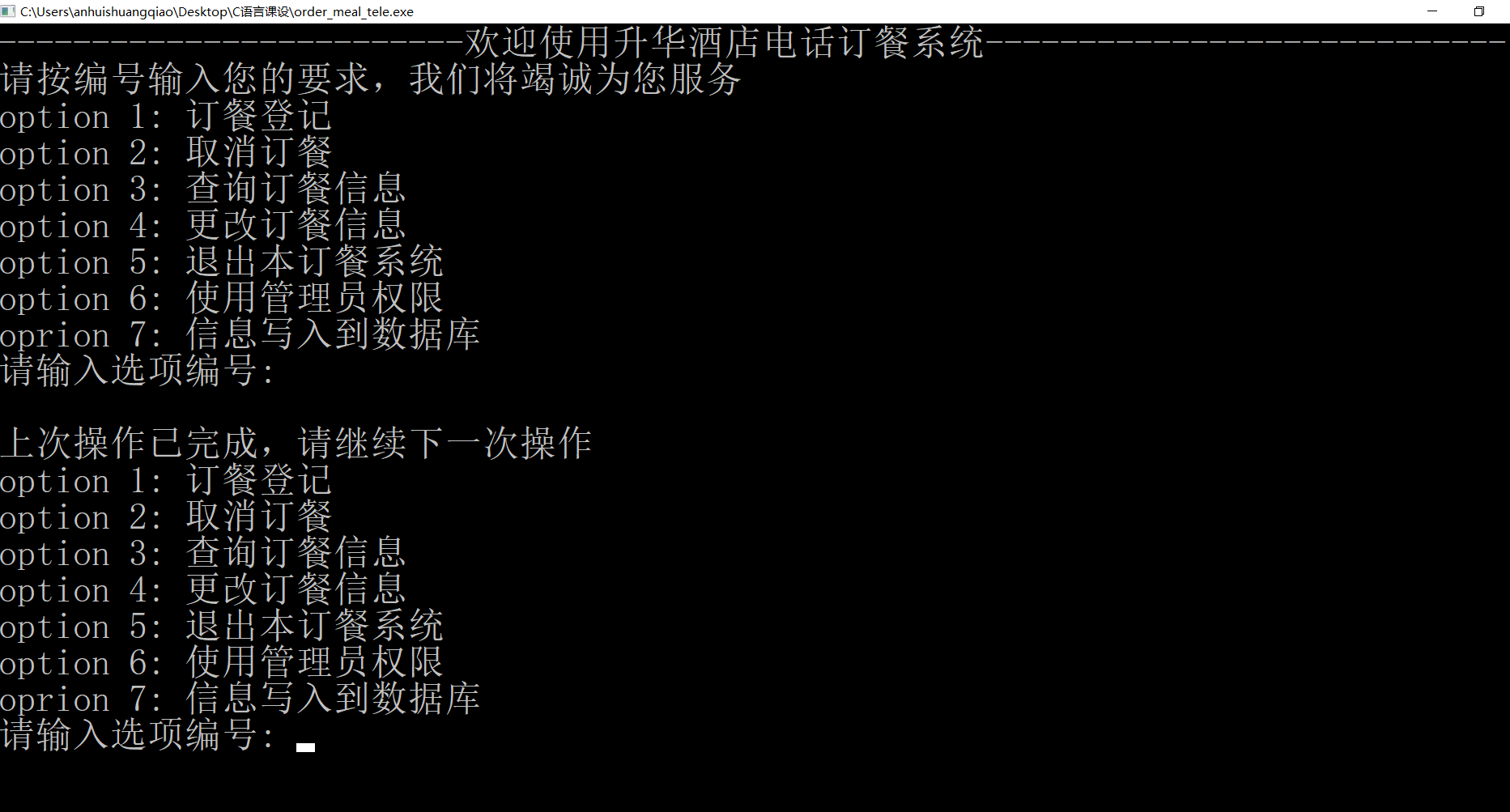
可以看到，中南小团子的订餐信息已被删除



2、 拒绝取消订餐，返回主菜单（用户输入3，拒绝取消，返回主菜单）



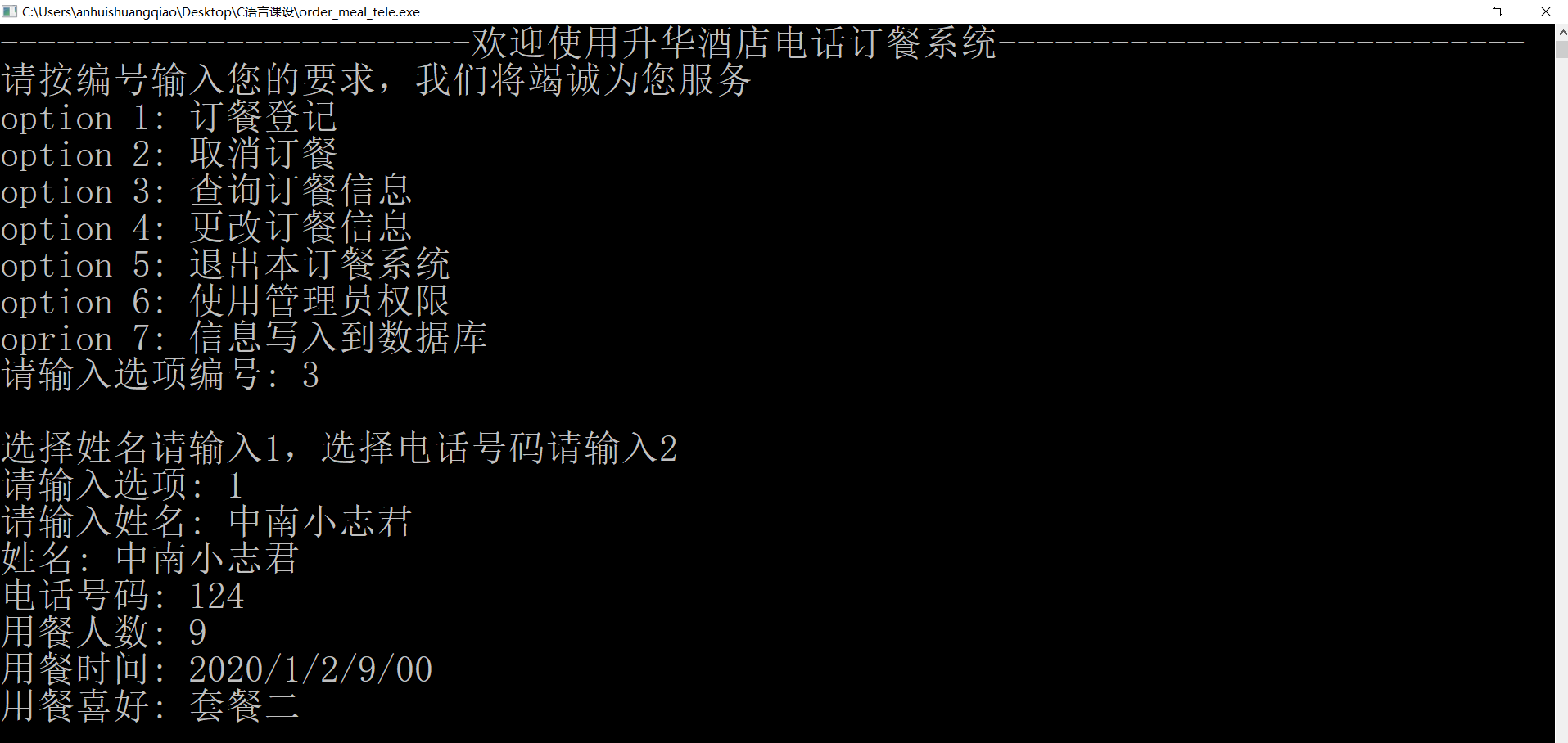
按下回车后



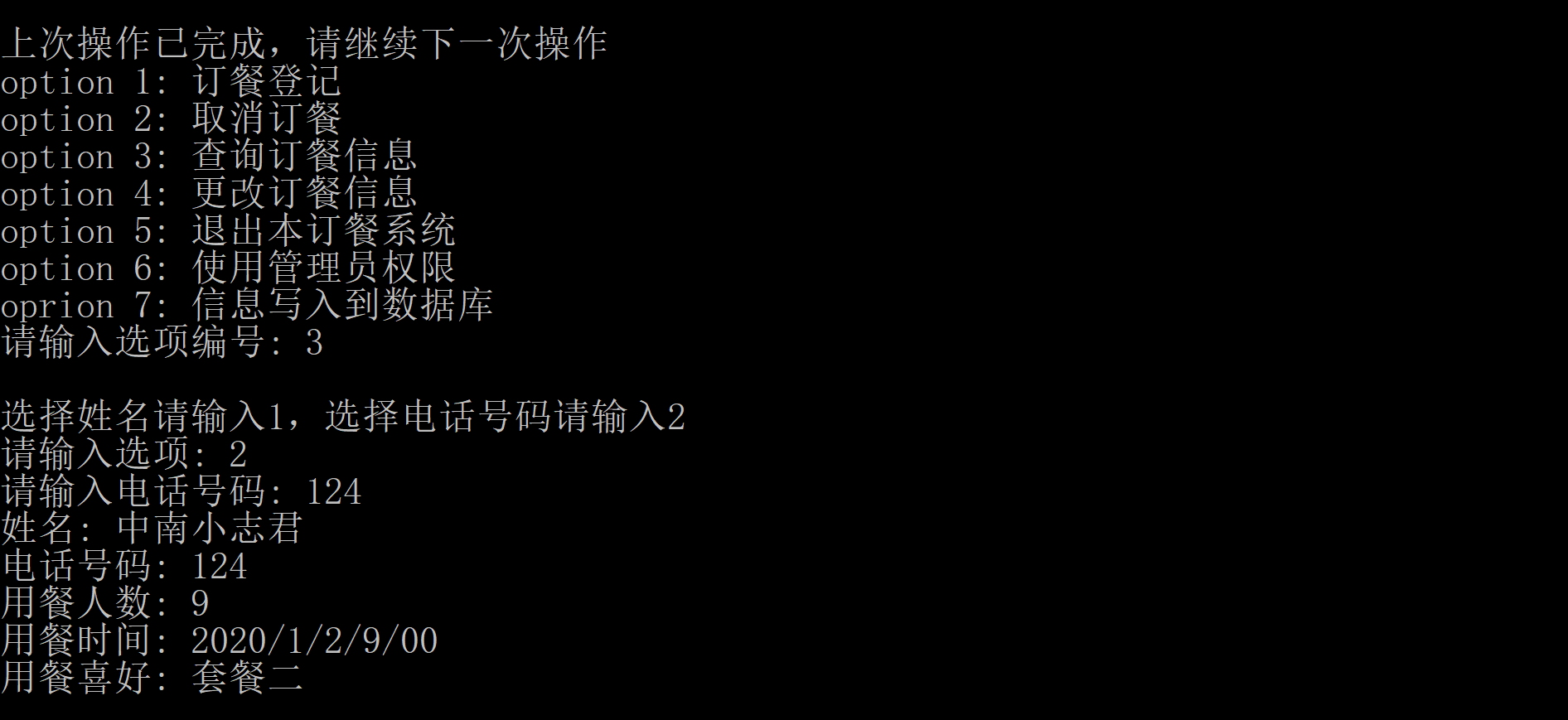
## 功能三：查询订餐信息

既可根据姓名查询，又可根据电话查询

1. 根据姓名查询（下图为用户输入选项1后，根据姓名查询“中南小志君”的订餐）



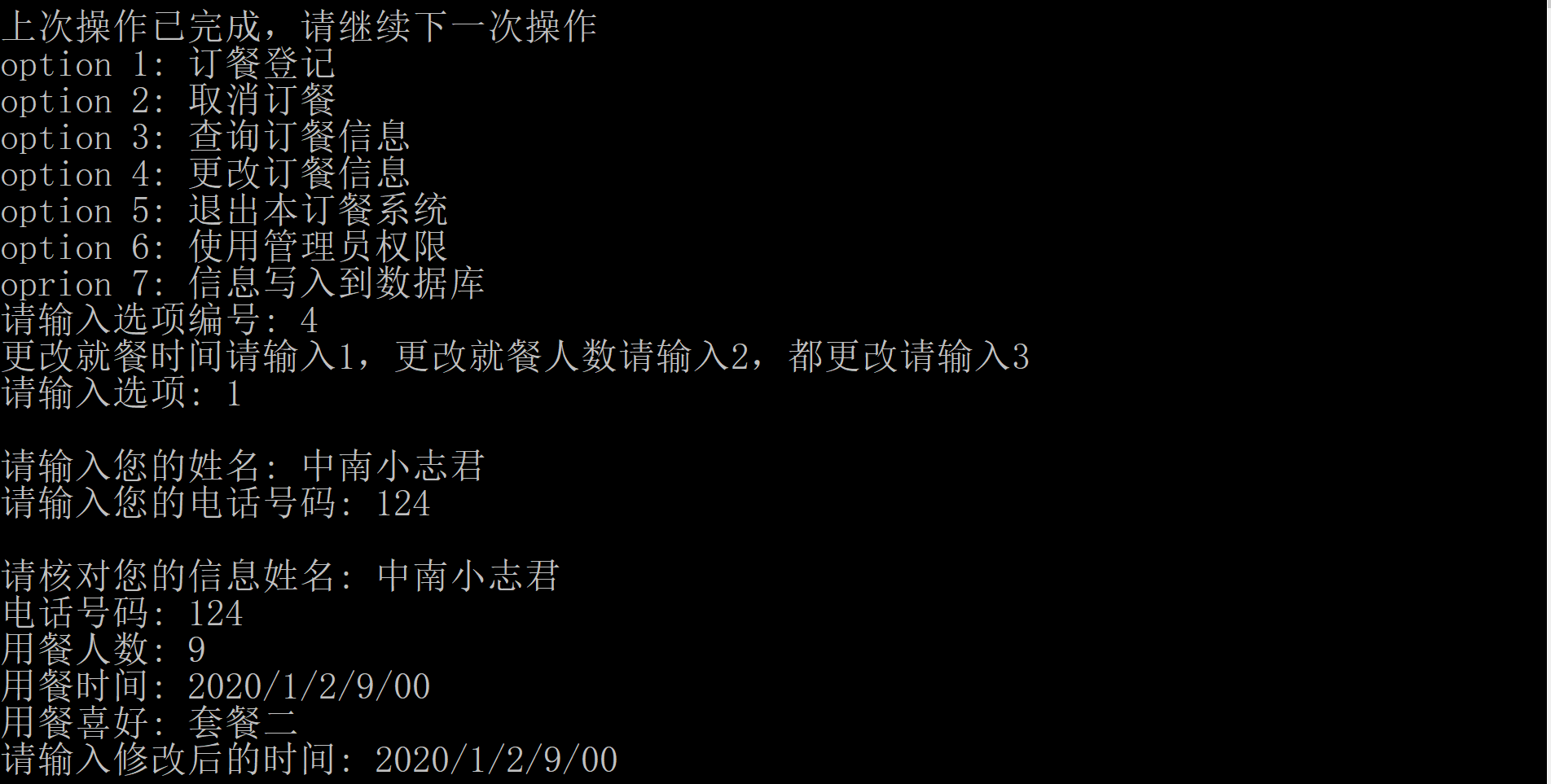
1. 根据电话查询（下图为用户输入选项二后，根据电话查询电话124的订餐）



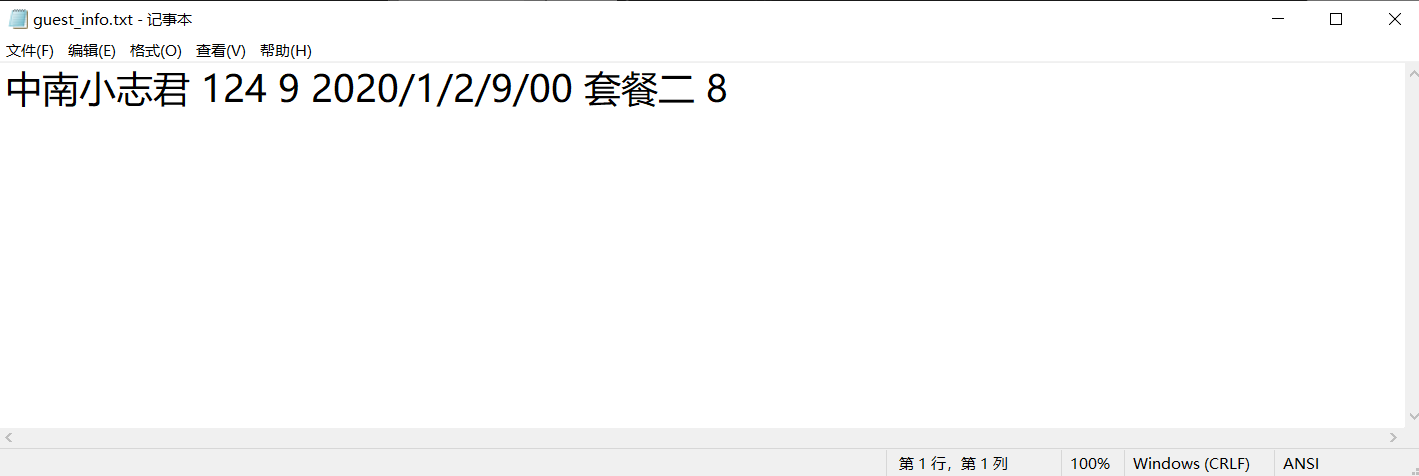
## 功能四：更改订餐信息

可更改订餐时间，或更改用餐人数，或两者均更改

1. 更改订餐时间（下图为用户输入选项1后，更改“中南小志君”的订餐时间）

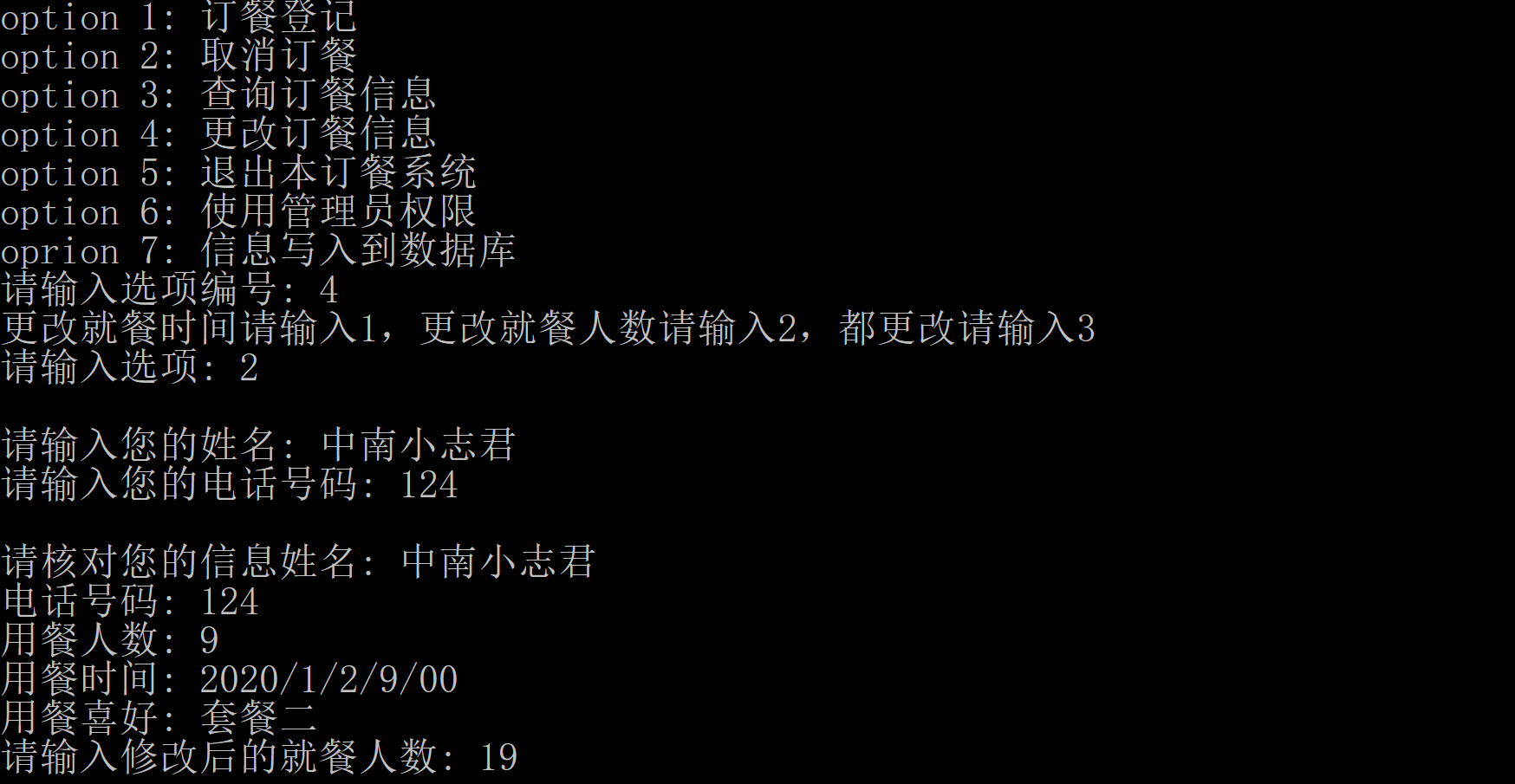


在TXT文件中表示如下



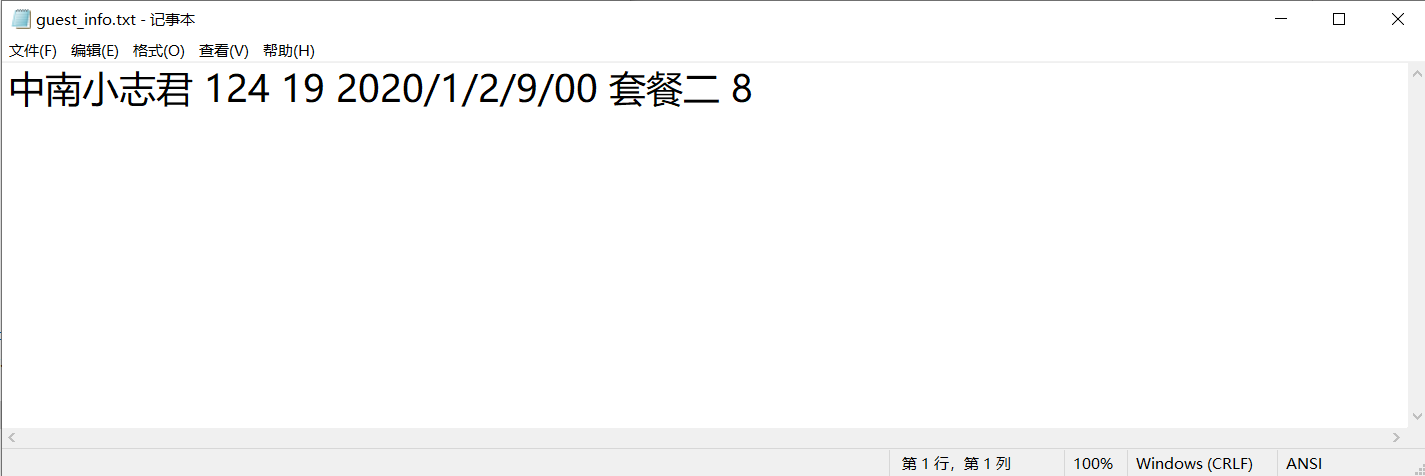
可以看到，中南小志君的订餐时间由2020/1/2/9/00修改为2019/12/31/10/00

1. 更改用餐人数（下图为用户输入选项二后，更改“中南小志”的用餐人数



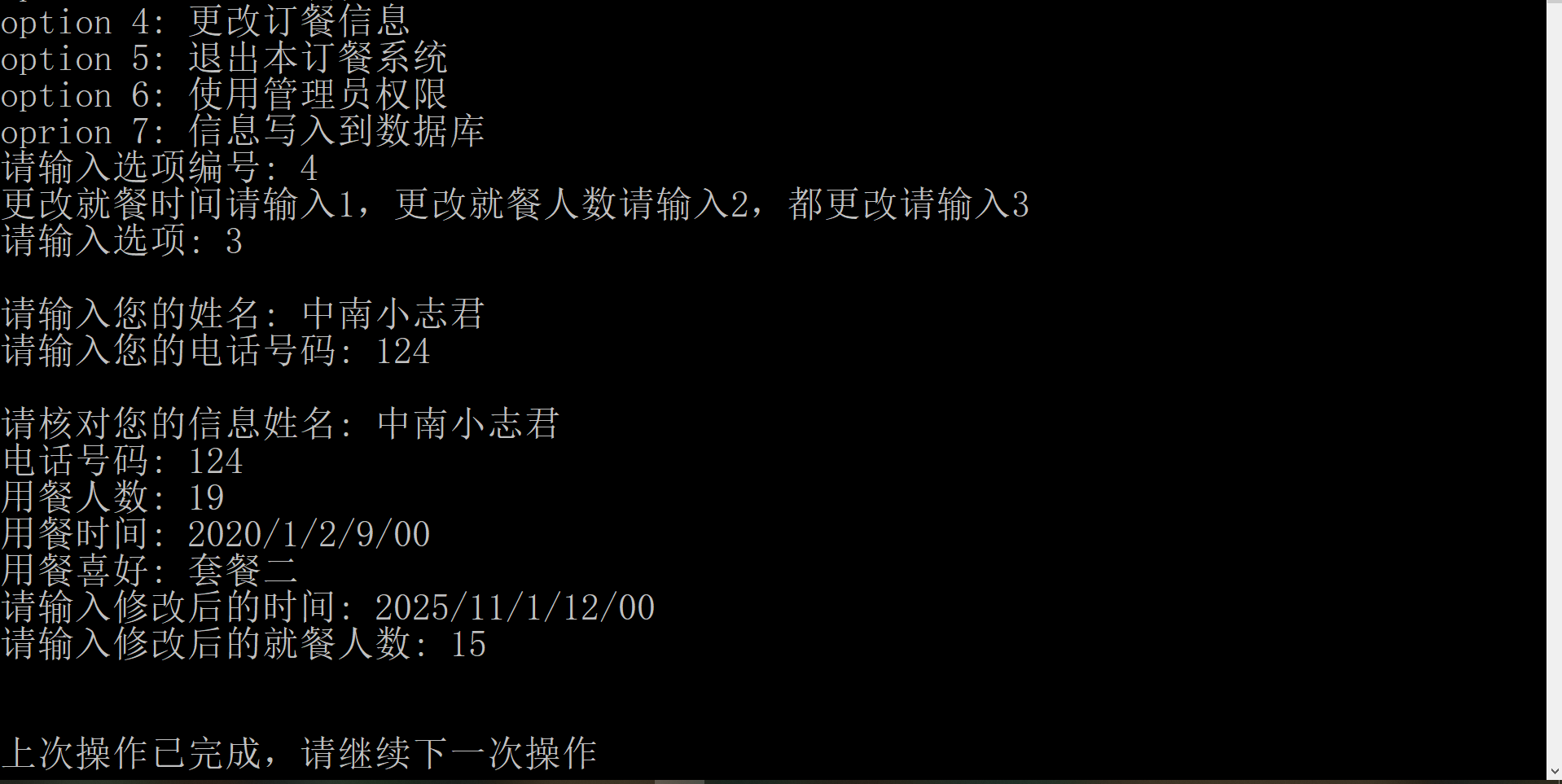
在TXT文件中表示如下

可以看到，中南小志君的用餐人数已从9变为19

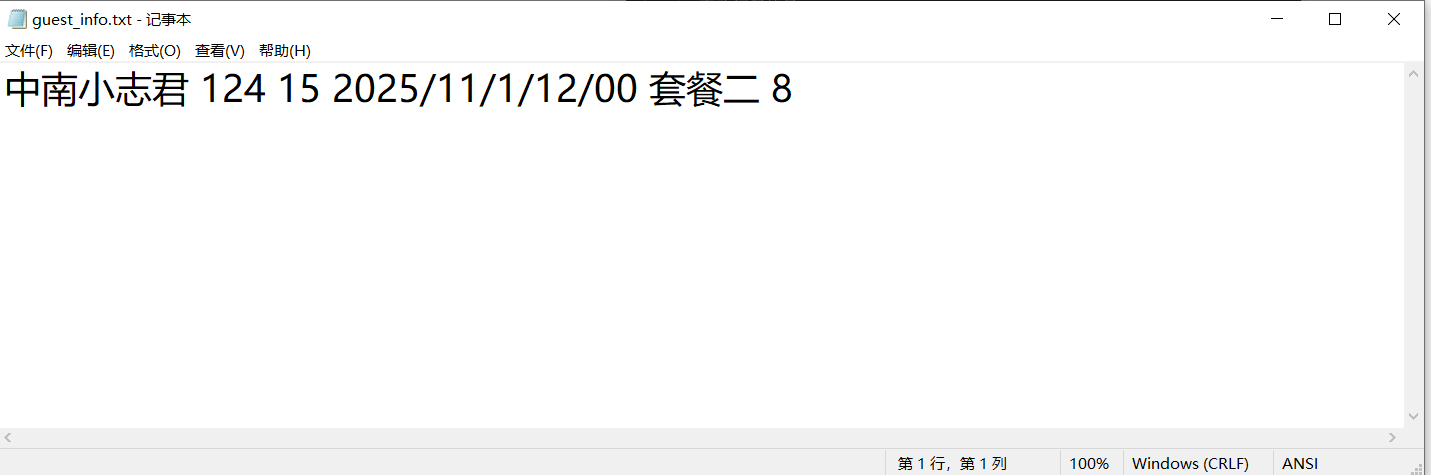


1. 更改订餐时间和用餐人数

下图为用户输入选项三后，修改”中南小志君“的订餐时间和用餐人数



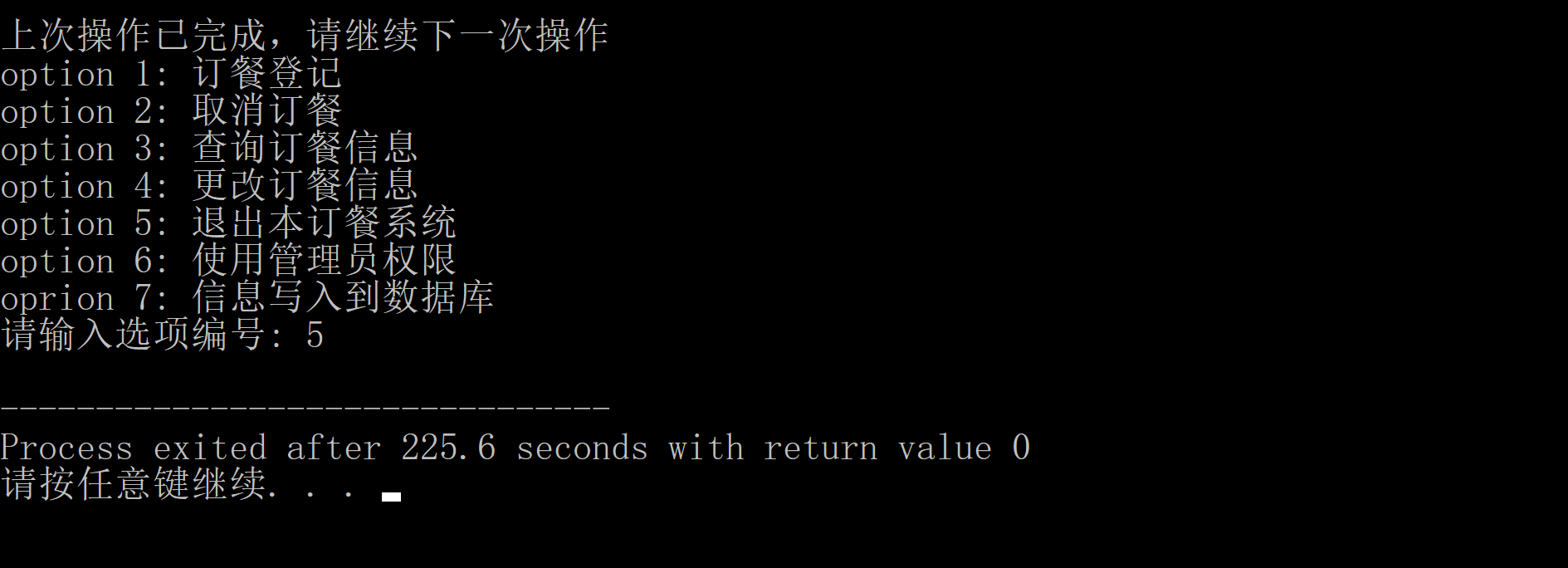
在TXT文件中表示如下



可以看到，中南小志君的订餐时间从2019/12/32/10/00更改为2025//11/1/12/00，用餐人数从20更改为15

## 功能五：退出本订餐系统

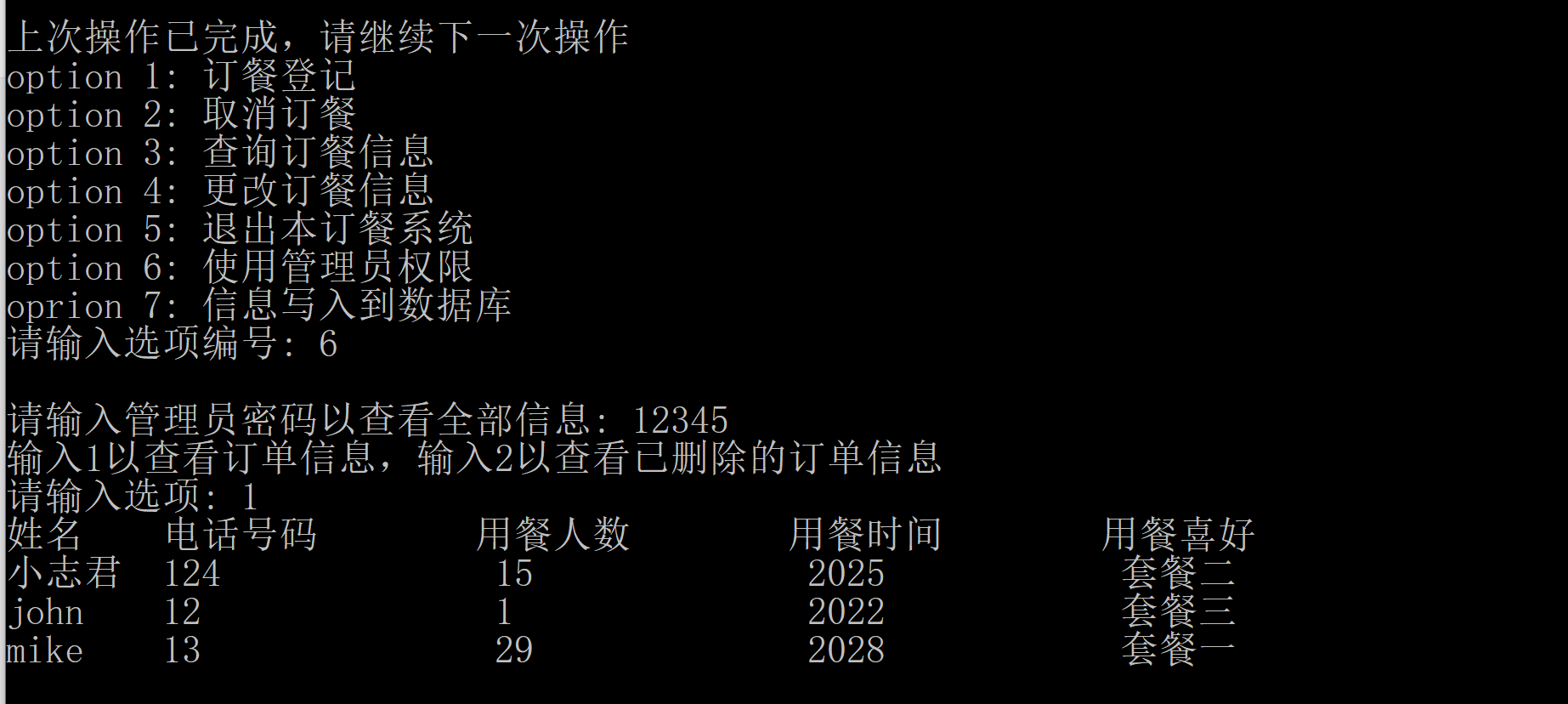
用户在输入选项5后，退出订餐系统，此时按任何键终止此程序



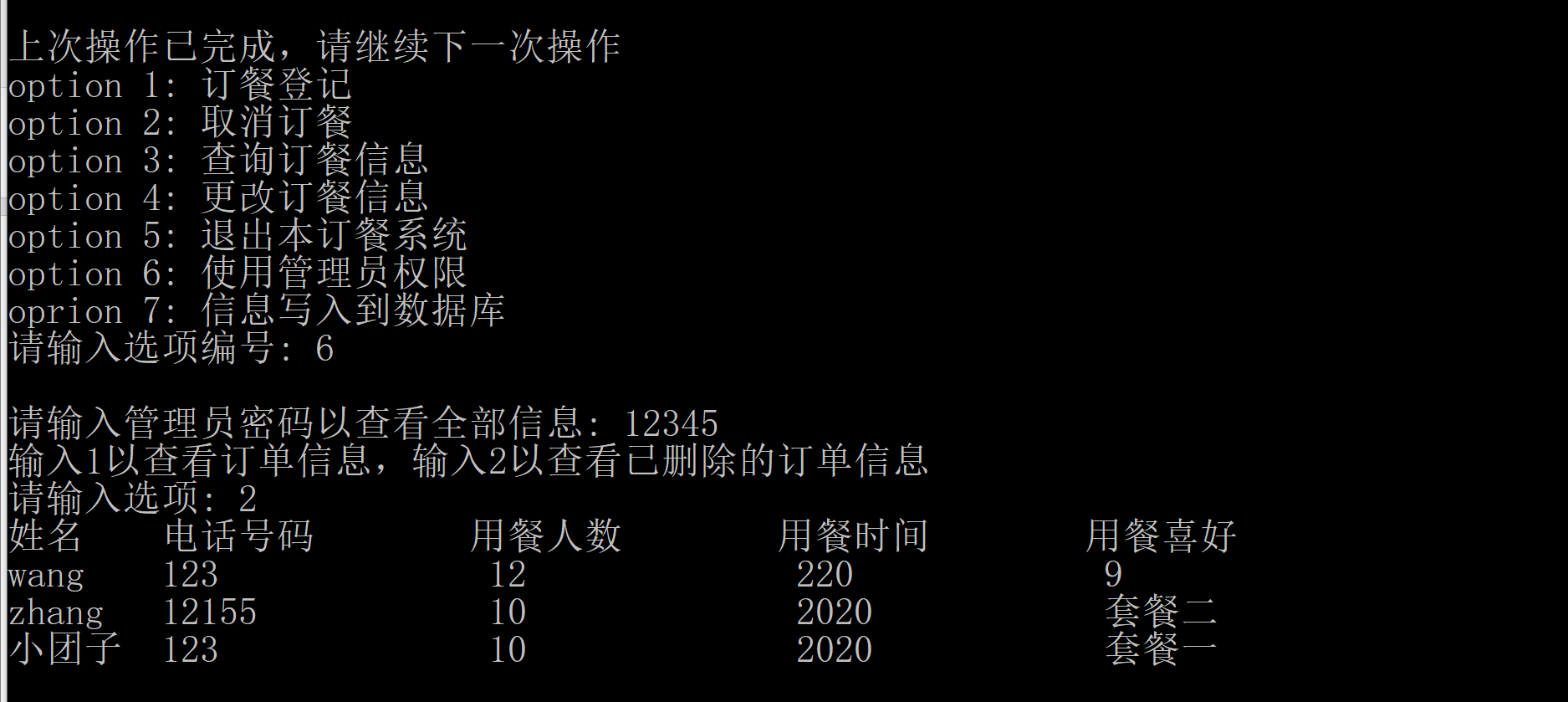
## 功能六：使用管理员权限

用户在输入选项6后，提示输入管理员密码（这里提前设置的管理员密码为12345），输入密码后进入管理员界面，可根据选项查看信息。

选线1查看所有订单信息（为明显表示，已另外加入信息）

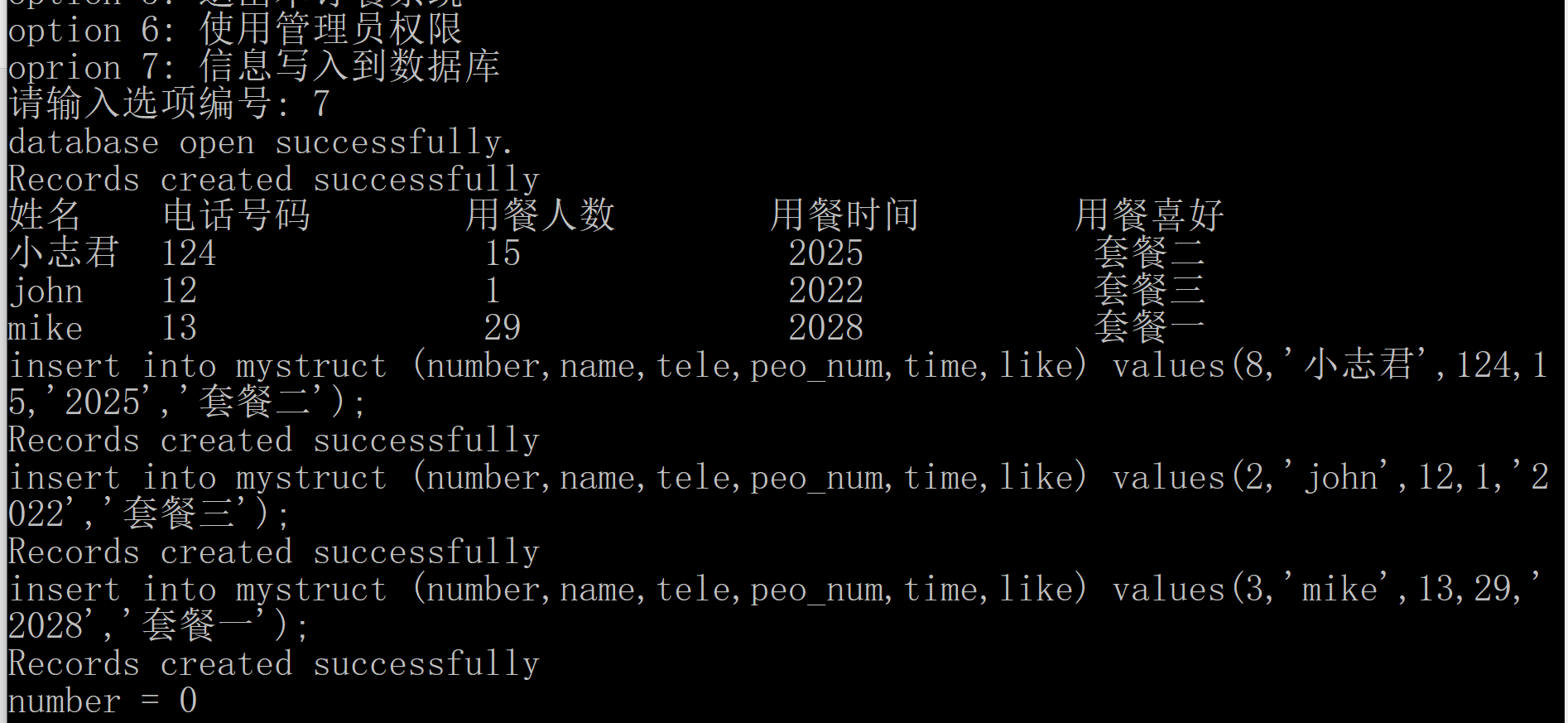


选项2产看所有被删除信息（为明显表示，已另外加入并删除信息）

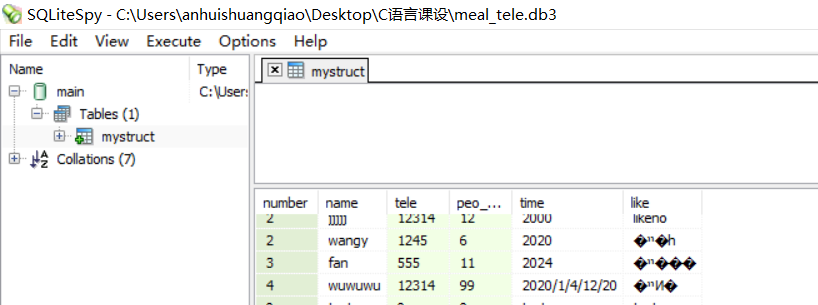


## 功能七：信息写入到数据库

输入选项7后，显示存储提示过程。



在数据库中表示如下



（功能还待优化）

# 源代码

注：sqlite3部分由于过于冗长，并未加入，读者可自行登录官网下载。

## main.c

//main.c

#include "meal\_tele\_fun.h"

int main(int argc, char \*argv[]) {

Info guest[101];//餐厅最多接受100份订餐，1-100

int guest\_number=0;//编号，每次订餐登记时加一 ,从文件读入

guest\_number=read(guest);

interface();//绘出界面

int option=0;

int flag=0;

while(1)//选择，在用户输入非法选项时提醒重新输入，一次操作完成后给出下一步操作提示

{

scanf("%d",&option);

do{

switch(option)

{

case 1: guest\_number++;meal\_regi(guest,guest\_number);;break;

case 2: meal\_canc(guest);break;

case 3: meal\_inqu(guest);break;

case 4: meal\_chan(guest);break;

case 5: return 0;

case 6: admin(guest);break;

case 7: exp();

default: printf("请重新输入正确的编号: ");flag=1;break;

}

if(flag==1)break;

}while(option!=1 && option!=2 && option!=3 && option!=4 && option!=5 && option!=6 && option!=7);

if(flag==1)continue;

printf("\n\n上次操作已完成，请继续下一次操作\n");

printf("option 1: 订餐登记\n");

printf("option 2: 取消订餐\n");

printf("option 3: 查询订餐信息\n");

printf("option 4: 更改订餐信息\n");

printf("option 5: 退出本订餐系统\n");

printf("option 6: 使用管理员权限\n");

printf("oprion 7: 信息写入到数据库\n");

printf("请输入选项编号: ");

}

return 0;

}

## struct\_info

//struct\_info

typedef struct mystruct

{

int number;//编号

char name[30];

long tele;

int peo\_num;//用餐人数<=20

char time[30];

char like[100];//用餐喜好，动态分配内存

}Info;

## meal\_tele\_func.h

//meal\_tele\_func.h

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "struct\_info.h"

void print\_all(Info guest[]);

void admin(Info guest[]);

int read(Info guest[]);

void interface();

void meal\_regi(Info guest[],int guest\_number);

void meal\_canc(Info guest[]);

void meal\_inqu(Info guest[]);

void meal\_chan(Info guest[]);

int exp();

## meal\_tele\_fun.c

//meal\_tele\_fun.c

#include "meal\_tele\_fun.h"

void admin\_opti(Info guest[])//管理员功能选项

{

int option=0;

printf("输入1以查看订单信息，输入2以查看已删除的订单信息");

printf("\n请输入选项: ");

scanf("%d",&option);

if(1==option)

{

print\_all(guest);

}

else if(2==option)

{

FILE\* fp=fopen("D:\\guest\_dele\_info.txt","r");//读入信息到结构体数组

if(NULL==fp)

{

printf("fopen fain in admin\_option fp");

return;

}

Info guest\_dele[101];

for(int guest\_number=1;guest\_number<=100;guest\_number++)

{

fscanf(fp,"%s",guest\_dele[guest\_number].name);

fscanf(fp,"%ld",&guest\_dele[guest\_number].tele);

fscanf(fp,"%d",&guest\_dele[guest\_number].peo\_num);

fscanf(fp,"%s",guest\_dele[guest\_number].time);

fscanf(fp,"%s",guest\_dele[guest\_number].like);

fscanf(fp,"%d",&guest\_dele[guest\_number].number);

if(guest\_dele[guest\_number].tele==0) break;//由于tele总是被初始化为0，故碰到0时，即停止读入剩下的空结构体

}

fclose(fp);

print\_all(guest\_dele);

}

else

{

printf("请输入正确的选项!\n");

admin\_opti(guest);

}

}

void print(Info guest)

{

printf("姓名: %s\n",guest.name);

printf("电话号码: %ld\n",guest.tele);

printf("用餐人数: %d\n",guest.peo\_num);

printf("用餐时间: %s\n",guest.time);

printf("用餐喜好: %s\n",guest.like);

}

void print\_all(Info guest[])//输出此结构体数组所有信息

{

printf("姓名\t电话号码\t用餐人数\t用餐时间\t用餐喜好\n");

for(int k=1;k<=100;k++)

{

if(guest[k].tele==0 && guest[k].peo\_num==0 && guest[k].number==0)//遇到尚未存入信息的项就截止

{

break;

}

printf("%s\t%ld\t\t %d \t\t %s \t\t %s \n",guest[k].name,guest[k].tele,guest[k].peo\_num,guest[k].time,guest[k].like);

// print(guest[k]);

// printf("\n");

}

}

void admin(Info guest[])//管理员功能

{

printf("\n请输入管理员密码以查看全部信息: ");

int key=0;

scanf("%d",&key);

if(12345==key)

{

admin\_opti(guest);

}

else

{

printf("密码错误，请重新输入!\n");

admin(guest);

}

}

int read(Info guest[])//从txt文件中将顾客数据读入到结构体，返回最后一位顾客的编号 （遇到已删除的，删除并重读，遇到截止，停止读入）

{

int guest\_number=0;

FILE \*fp=fopen("D:\\guest\_info.txt","a+");

if(NULL==fp)

{

printf("fopen fail in read");

return 0;

}

for(guest\_number=1;guest\_number<=100;guest\_number++)

{

fscanf(fp,"%s",guest[guest\_number].name);

fscanf(fp,"%ld",&guest[guest\_number].tele);

fscanf(fp,"%d",&guest[guest\_number].peo\_num);

fscanf(fp,"%s",guest[guest\_number].time);

fscanf(fp,"%s",guest[guest\_number].like);

fscanf(fp,"%d",&guest[guest\_number].number);

if(guest[guest\_number].tele==0 && strcmp(guest[guest\_number].name,"0")==0)//遇到被删除的内容，即0 0 1，指针退格再读

{

dele\_gue(guest,guest\_number);//利用dele\_gue的删除、重新写入的功能，逐个排除已被删除的内容

read(guest);//重新从文件读入结构体

}

if(guest[guest\_number].tele==0) break;//由于tele总是被初始化为0，故碰到0时，即停止读入剩下的空结构体

}

fclose(fp);

return \*&guest\_number-1;//由于其实多读了一个，所以减一

}

void dele\_gue(Info guest[],int guest\_number)//先将要删除的数据存入新的文件，再删除结构体数组中的指定内容，再将结构体重新写入

{

FILE\* fs=fopen("D:\\guest\_dele\_info.txt","a");

if(NULL==fs)

{

fprintf(stderr,"fopen fain in dele\_gue fs");

return;

}

fprintf(fs,"%s %ld %d %s %s %d\n",guest[guest\_number].name, guest[guest\_number].tele, guest[guest\_number].peo\_num, guest[guest\_number].time, guest[guest\_number].like, guest\_number);

fclose(fs);

for(int j=0;j<30;j++)//再清除对应结构体

{

guest[guest\_number].name[j]=0;

guest[guest\_number].time[j]=0;

guest[guest\_number].like[j]=0;

}

guest[guest\_number].tele=0;

guest[guest\_number].peo\_num=0;

FILE \*fp=fopen("D:\\guest\_info.txt","w+");//将结构体重新写入

if(NULL==fp)

{

printf("fopen fail in dele\_gue");

return;

}

for(int k=1;k<=guest\_number-1;k++)//重新写入，遇到尚未存入新信息的项就截止

{

fprintf(fp,"%s %ld %d %s %s %d\n",guest[k].name, guest[k].tele, guest[k].peo\_num, guest[k].time, guest[k].like, guest[k].number);

}

for(int k=guest\_number+1;k<=100;k++)

{

if(guest[k].tele==0 && guest[k].peo\_num==0 && guest[k].number==0)

{

break;

}

fprintf(fp,"%s %ld %d %s %s %d\n",guest[k].name, guest[k].tele, guest[k].peo\_num, guest[k].time, guest[k].like, guest[k].number);

}

fclose(fp);

}

void meal\_canc\_option(Info guest[],int i)//读入meal\_canc里的option选项，根据选项进行功能切换，i为符合输入的顾客编号

{

int option=0;

scanf("%d",&option);

if(1==option)

{

dele\_gue(guest,i);

}

else if(2==option)

{

meal\_canc(guest);

}

else if(3==option)

{

system("CLS");

interface();

}

else

{

printf("请重新输入正确的选项: ");

meal\_canc\_option(guest,i);

}

}

int compare(Info guest[])//根据输入的姓名和电话号码寻找符合的顾客信息，输出信息，返回顾客编号

{

printf("\n请输入您的姓名: ");

char name[30];

scanf("%s",name);

printf("请输入您的电话号码: ");//防止同名

long tele;

scanf("%ld",&tele);

int i=0;

for(i=1;i<=100;i++)

{

if(strcmp(name,guest[i].name)==0 && tele==guest[i].tele)

{

printf("\n请核对您的信息");

print(guest[i]);

return \*(&i);

}

}

printf("对不起，系统未检测到您的信息");

return 0;

}

void interface()//初始界面

{

printf("-------------------------欢迎使用升华酒店电话订餐系统----------------------------\n");

printf("请按编号输入您的要求，我们将竭诚为您服务\n");

printf("option 1: 订餐登记\n");

printf("option 2: 取消订餐\n");

printf("option 3: 查询订餐信息\n");

printf("option 4: 更改订餐信息\n");

printf("option 5: 退出本订餐系统\n");

printf("option 6: 使用管理员权限\n");

printf("oprion 7: 信息写入到数据库\n");

printf("请输入选项编号: ");

}

void meal\_regi\_opti(Info guest[],int guest\_number)

{

printf("请按对应序号输入您的用餐选项: ");

int sele=0;

scanf("%d",&sele);

if(1==sele)

{

strcpy(guest[guest\_number].like,"套餐一");

}

else if(2==sele)

{

strcpy(guest[guest\_number].like,"套餐二");

}

else if(3==sele)

{

strcpy(guest[guest\_number].like,"套餐三");

}

else

{

printf("请重新输入正确选项");

meal\_regi\_opti(guest,guest\_number);

}

}

void meal\_regi(Info guest[],int guest\_number)//订餐登记，将信息存入结构体，再存到txt文件中去，注意地址

{

guest[guest\_number].number=guest\_number;

printf("请输入您的姓名： ");

scanf("%s",guest[guest\_number].name);

printf("请输入您的电话号码: ");

scanf("%ld",&guest[guest\_number].tele);

printf("请输入用餐人数: ");

scanf("%d",&guest[guest\_number].peo\_num);

printf("请输入用餐时间:(y/m/d/h/min) ");

scanf("%s",guest[guest\_number].time);

printf("我们提供三个套餐供选择:\n");

printf("套餐一: 西红柿炒蛋+米饭+冬瓜排骨汤\n");

printf("套餐二: 土豆红烧肉+米饭+肉丸汤\n");

printf("套餐三: 汉堡+鸡腿+鸡肉卷\n");

meal\_regi\_opti(guest,guest\_number);

FILE \*fp=fopen("D:\\guest\_info.txt","a");

if(NULL==fp)

{

printf("fopen fail in meal\_regi");

return;

}

//向文件依次写入 姓名 电话号码 用餐人数 用餐时间 用餐喜好 编号（放最后，方便读取）

fprintf(fp,"%s %ld %d %s %s %d\n",guest[guest\_number].name, guest[guest\_number].tele, guest[guest\_number].peo\_num, guest[guest\_number].time, guest[guest\_number].like, guest\_number);

fclose(fp);

}

void meal\_canc(Info guest[])//取消订餐

{

int i=compare(guest);//符合输入的顾客编号

if(0==i)

{

return;

}

printf("请问是否确认取消此次订餐(确认请输入1，信息错误想重新填写请输入2，否并返回主菜单请输入3\n");

printf("请输入选项: ");

meal\_canc\_option(guest,i);

}

void meal\_inqu(Info guest[])

{

printf("\n选择姓名请输入1，选择电话号码请输入2\n");

printf("请输入选项: ");

int option=0;

scanf("%d",&option);

char name[30];

long tele;

if(1==option)

{

printf("请输入姓名: ");

scanf("%s",name);

for(int i=1;i<=100;i++)

{

if(strcmp(name,guest[i].name)==0)

{

print(guest[i]);

return;

}

}

printf("对不起，系统未检测到您的信息");

}

else if(2==option)

{

printf("请输入电话号码: ");

scanf("%ld",&tele);

for(int i=1;i<=100;i++)

{

if(tele==guest[i].tele)

{

print(guest[i]);

return;

}

}

printf("对不起，系统未检测到您的信息");

}

else

{

printf("请重新输入正确的选项!\n");

meal\_inqu(guest);

}

}

void meal\_rewr(Info guest[])//将结构体重新写入文件流

{

FILE \*fp=fopen("D:\\guest\_info.txt","w+");

if(NULL==fp)

{

printf("fopen fail in meal\_rewr");

return;

}

for(int k=1;k<=100;k++)

{

if(guest[k].tele==0 && guest[k].peo\_num==0 && guest[k].number==0)//遇到尚未存入信息的项就截止

{

break;

}

fprintf(fp,"%s %ld %d %s %s %d\n",guest[k].name, guest[k].tele, guest[k].peo\_num, guest[k].time, guest[k].like, guest[k].number);

}

fclose(fp);

}

void meal\_chan(Info guest[])

{

printf("更改就餐时间请输入1，更改就餐人数请输入2，都更改请输入3\n");

printf("请输入选项: ");

int option=0;

scanf("%d",&option);

if(1==option)

{

int i=compare(guest);

if(0==i)return;//未查询到信息时退出

printf("请输入修改后的时间: ");

char new\_time[30];

scanf("%s",new\_time);

strcpy(guest[i].time,new\_time);

meal\_rewr(guest);

}

else if(2==option)

{

int i=compare(guest);

if(0==i)return;//未查询到信息时退出

printf("请输入修改后的就餐人数: ");

int new\_num=0;

scanf("%d",&new\_num);

guest[i].peo\_num=new\_num;

meal\_rewr(guest);

}

else if(3==option)

{

int i=compare(guest);

if(0==i)return;//未查询到信息时退出

printf("请输入修改后的时间: ");

char new\_time[30];

scanf("%s",new\_time);

strcpy(guest[i].time,new\_time);

printf("请输入修改后的就餐人数: ");

int new\_num;

scanf("%d",&new\_num);

guest[i].peo\_num=new\_num;

meal\_rewr(guest);

}

else

{

printf("请重新输入正确的选项!\n");

meal\_chan(guest);

}

}

## sqlite3\_experiment.c

//sqlite3\_experiment.c

#include <stdio.h>

#include <stdlib.h>

#include "sqlite3.h"

#include "meal\_tele\_fun.h"

int callback(void \*NotUsed, int argc, char \*\*argv, char \*\*azColName){

int i;

for(i=0; i<argc; i++){

printf("%s = %s\n", azColName[i], argv[i] ? argv[i] : "NULL");

}

printf("\n");

return 0;

}

int exp()

{

sqlite3 \*db;

int rc;

char \*sql;

char \*zErrMsg;

rc = sqlite3\_open("meal\_tele.db3",&db);//rc使返回值，用来报错

if(rc)

{

fprintf(stderr,"Can not open database:%s\n",sqlite3\_errmsg(db));

return -1;

}

else

{

fprintf(stdout,"database open successfully.\n");

}

sql = "insert into mystruct (number,name,tele,peo\_num,time,like) values(0,\'test\',0,0,\'test\',\'test\');";//sql语句 ,引号转义

rc = sqlite3\_exec(db, sql, NULL, 0, &zErrMsg);

if( rc != SQLITE\_OK )

{

fprintf(stderr, "SQL error: %s\n", zErrMsg);

sqlite3\_free(zErrMsg);

}

else

{

fprintf(stdout, "Records created successfully\n");

}

Info std[101];

read(std);

print\_all(std);

char sqlstr[100]={0};

for(int i=1;i<=100;i++)

{

if(std[i].tele==0 && std[i].peo\_num==0 && std[i].number==0)//遇到尚未存入信息的项就截止

{

break;

}

memset(sqlstr,0,100);

sprintf(

sqlstr,

"insert into mystruct (number,name,tele,peo\_num,time,like) values(%d,\'%s\',%ld,%d,\'%s\',\'%s\');",

std[i].number,

std[i].name,

std[i].tele,

std[i].peo\_num,

std[i].time,

std[i].like);

puts(sqlstr);

sql = sqlstr;

rc = sqlite3\_exec(db, sql, NULL, 0, &zErrMsg);

if( rc != SQLITE\_OK )

{

fprintf(stderr, "SQL error: %s\n", zErrMsg);

sqlite3\_free(zErrMsg);

}

else

{

fprintf(stdout, "Records created successfully\n");

}

}

sql = "select \* from mystruct";

rc = sqlite3\_exec(db, sql, callback, 0, &zErrMsg);

if( rc != SQLITE\_OK )

{

fprintf(stderr, "SQL error: %s\n", zErrMsg);

sqlite3\_free(zErrMsg);

}

else

{

fprintf(stdout, "Records created successfully\n");

}

char \*\*dbResult;

int nRow, nColumn;

rc = sqlite3\_get\_table( db, sql, &dbResult, &nRow, &nColumn, &zErrMsg );

if( rc != SQLITE\_OK )

{

fprintf(stderr, "SQL error: %s\n", zErrMsg);

sqlite3\_free(zErrMsg);

}

else

{

int i,j;

for(i=0;i<nColumn;i++)

printf("%s\t\t",dbResult[i]);

putchar('\n');

for(i=0;i<nRow;i++){

for(j=0;j<nColumn;j++){

printf("%s\t\t",dbResult[nColumn+i\*nColumn+j]);

}

putchar('\n');

}

}

sqlite3\_close(db);

}