北京邮电大学 计算机学院

《面向对象程序设计与实践》

实验报告

姓名 王睿嘉

学号2015211906

班级2015211307

目录

[银行门户 3](#_Toc502083184)

[一、 需求分析 3](#_Toc502083185)

[二、 功能说明 3](#_Toc502083186)

[三、 软件设计 5](#_Toc502083187)

[1. 高层数据结构设计 5](#_Toc502083188)

[2. 系统模块划分 7](#_Toc502083189)

[3. 高层算法设计 8](#_Toc502083190)

[四、 性能测试 9](#_Toc502083191)

[电商平台 16](#_Toc502083192)

[一、 需求分析 16](#_Toc502083193)

银行门户

1. 需求分析

该门户是一个单独的程序，至少支持以下功能：

1. 注册银行卡，且银行可选；
2. 修改银行卡密码；
3. 存款取款。

其中，银行卡至少要有卡号，密码，所属银行名称，持卡人身份证号，卡内金额等内容。

附加要求：

1. 把所有银行名写入文件（或数据库）。注册银行卡时，只能选择已存在的银行进行注册；
2. 做好错误场景的处理。例如，读银行文件错误、输入数据不合法等。
3. 功能说明

|  |  |  |  |
| --- | --- | --- | --- |
| 模块功能 | 输入 | 处理方式 | 输出 |
| 注册模块 | 注册银行选择  姓名  身份证号  预留手机号  密码  确认密码 | 判断银行选择是否正确；  判断身份证号是否为18位，允许字母出现；  判断手机号是否为11位数字；  判断密码长度是否为6位，且仅由数字组成；  确认密码与原密码是否相同 | 若银行选择有误，输出提示信息，并要求重新选择；若身份证号不符合规定，输出提示信息，并要求重新输入；若预留手机号不符合规定，输出提示信息，并要求重新输入；  若密码不符合规定，输出提示信息，并要求重新输入；若确认密码与原密码不符，输出提示信息，并要求重新设置密码；若信息均正确，注册成功，并输出相应提示信息 |
| 存款模块 | 卡号  密码  存款金额 | 判断账户是否处于冻结状态；  判断银行卡是否存在；  判断密码是否正确；  判断存款金额是否为正的小数或整数 | 若账户处于冻结状态，则无法进行存款操作，输出相应提示信息；若银行卡不存在，输出提示信息，并要求重新输入；若密码不正确，输出提示信息，并要求重新输入；若密码错误次数超过三次，账户被冻结，输出相应提示信息；若存款金额不符合规定，输出提示信息，并要求重新输入；若信息均正确，存款成功，并输出相应提示信息 |
| 取款模块 | 取款金额 | 判断账户是否处于冻结状态；  判断取款金额是否为正的小数或整数，且是否超过每日最大额度（5000元）和账户余额的较小值 | 若账户处于冻结状态，则无法进行取款操作，输出相应提示信息；若取款金额不符合规定，输出提示信息，并要求重新输入；若信息均正确，取款成功，并输出相应提示信息 |
| 修改密码模块 | 新密码  确认密码 | 判断账户是否处于冻结状态；  判断新密码长度是否为6位，且仅由数字组成；  判断确认密码与新密码是否相同 | 若账户处于冻结状态，则无法进行修改密码操作，输出相应提示信息；若新密码不符合规定，输出提示信息，并要求重新输入；若确认密码与新密码不符，输出提示信息，并要求重新设置密码；若信息均正确，修改密码成功，并输出相应信息 |
| 修改预留手机模块 | 新手机号 | 判断账户是否处于冻结状态；  判断新手机号是否为11位数字 | 若账户处于冻结状态，则无法进行修改预留手机操作，输出相应提示信息；若新手机号不符合规定，输出提示信息，并要求重新输入；若信息均正确，修改预留手机成功，并输出相应信息 |
| 查询交易记录模块 | 无 | 判断账户是否处于冻结状态 | 若账户处于冻结状态，则无法进行查询交易记录操作，输出相应提示信息；否则，打印每笔交易的时间及金额，存入以“+”表示，支出以“-”表示，并输出当前账户余额 |
| 挂失模块 | 无 | 使账户进入冻结状态，无法进行任何操作 | 若账户处于正常状态，则挂失成功，并输出相应提示信息 |

注：其中，存款、取款、修改密码、修改预留手机、查询交易记录、挂失模块，若为初次进入系统，需输入卡号及密码，其相应的处理及输出放至存款模块说明，其余模块不再赘述。若已进行过某一模块的操作，则无需再输入银行卡信息。另外，注册银行卡、存款或取款后，会打印当前银行卡信息，包括所属银行、持卡人姓名、持卡人身份证号、账户余额、账户状态（正常或冻结）等。

1. 软件设计
2. 高层数据结构设计

1）类

class Account{ //账户

private:

char cardID[20]; //卡号（必为19位

string password; //密码（必为6位

int state; //银行卡状态

string name; //持卡人姓名

string ID; //持卡人身份证号（必为18位

string phoneNumber; //预留手机号（必为11位

string bank; //所属银行

double balance; //卡内余额

int transactionAmount; //交易次数

double transactionRecord[30]; //交易金额

double transactionTime[30]; //交易时间

public:

static void readAnnotation(); //跳过注释信息

static void writeAnnotation(); //向文件写入注释信息

void writeAccount(); //向文件写入账户信息

void setCardID(char\*); //设置卡号

int setPassword(string); //设置密码

void setName(string); //设置持卡人姓名

int setID(string); //设置持卡人身份证号

int setPhoneNumber(string); //设置预留手机号

void setBank(string); //设置所属银行

void setState(); //冻结账户

void generateCardID(); //随机生成卡号

int record(double); //记录交易金额

void recordTime(double); //记录交易时间

void readAccount(); //从文件读取账户信息

void printAccountInfo(); //打印账户信息

void printTransactionRecord(); //打印交易记录

void saveMoney(); //存款

void drawMoney(); //取款

char\* getcardID(); //获得卡号

string getPassword(); //获得密码

int getState(); //获得账户状态

Account();

~Account();

};

class Bank{ //银行

private:

string name; //名称

public:

void read(); //从文件读取信息

string getName(); //获得银行名称

Bank();

~Bank();

};

2）常量

#define MAX\_ACCOUNT\_AMOUNT 100 //最大账户数量

1. 全局变量

ifstream ifp; //输入文件流

ofstream ofp; //输出文件流

double systemTime; //系统时间

Account accounts[MAX\_ACCOUNT\_AMOUNT]; //存储账户信息

1. txt文件

File/ Account.txt 记录账户信息

每个账户的信息从上至下依次为：

卡号

密码

银行卡状态

持卡人姓名

持卡人身份证号

预留手机号

所属银行

卡内余额

交易次数

交易金额

交易时间

例如：

6733187098973668072

624158

0

小可爱

410303199709010023

15601282822

中国邮政储蓄银行

3000.1

2

8000.00 -4999.90

1.13 1.27

File/Bank.txt 记录银行信息

即，可注册的银行种类：

中国工商银行

中国农业银行

中国银行

中国建设银行

中国招商银行

中国邮政储蓄银行

中国银联

File/System Time.txt 记录系统时间

登陆系统后启动计时。该文件会保存时间，以便再次启动可以继续计时。

例如：

9.01

1. 源码文件

bankPortal.h 头文件：声明所有需要跨源文件调用的类、常量、函数等

bankPortal.cpp 定义main()和所有全局有效的函数

void readSystemTime();void incSystemTime();void writeSystemTime();

Bank.cpp 定义bank类的所有成员函数

Account.cpp 定义Account类的所有成员函数

1. 系统模块划分

各模块函数说明：

|  |  |  |
| --- | --- | --- |
| 模块 | 函数名称 | 功能 |
| 主模块 | int main() | 程序入口 |
| 注册模块 | void Bank.read() | 从文件读取银行种类 |
| string Bank.getName() | 获得银行名称 |
| void Account.setBank(string) | 设置所属银行 |
| void Account.setName(string) | 设置持卡人姓名 |
| int Account.setID(string) | 设置持卡人身份证号 |
| int Account.setPhoneNumber(string) | 设置预留手机号 |
| int Account.setPassword(string) | 设置密码 |
| void Account.generateCardID() | 随机生成卡号 |
| void Account.printAccountInfo() | 打印账户信息 |
| 存款模块 | char\* Account.getcardID() | 获得卡号 |
| string Account.getPassword() | 获得密码 |
| void Account.setState() | 冻结账户 |
| int Account.getState() | 获得账户状态 |
| void Account.saveMoney() | 存款 |
| int Account.record(double) | 记录交易金额 |
| void Account.recordTime(double) | 记录交易时间 |
| 取款模块 | void Account.drawMoney() | 取款 |
| int Account.record(double) | 记录交易金额 |
| void Account.recordTime(double) | 记录交易时间 |
| 修改密码模块 | int Account.setPassword(string) | 设置密码 |
| 修改预留手机模块 | int Account.setPhoneNumber(string) | 设置预留手机号 |
| 查询交易记录模块 | void Account.printTransactionRecord() | 打印交易记录 |
| 挂失模块 | void Account.setState() | 冻结账户 |

1. 高层算法设计
   1. 注册模块

int Account.setPassword(string){

判断密码长度是否为6位，且仅由数字组成

若不是

输出提示信息，并要求重新输入

否则

将账户密码设置为输入字符串

}

int Account.setPhoneNumber(string){

判断手机号是否为11位数字

若不是

输出提示信息，并要求重新输入

否则

将预留手机号设置为输入数字串

}

* 1. 存款模块

void Account.saveMoney(){

从标准输入流读入存款金额

判断存款金额是否符合规定

若符合{

记录交易金额

记录交易时间

存款成功，输出相应提示信息

打印账户当前信息（余额等）

}

}

* 1. 取款模块

void Account.drawMoney(){

打印当前账户余额

从标准输入流读入取款金额

判断取款金额是否大于每日最大额度（5000元）或超出账户余额

若是

输出提示信息，并要求重新输入

否则{

记录交易金额

记录交易时间

取款成功，输出相应提示信息

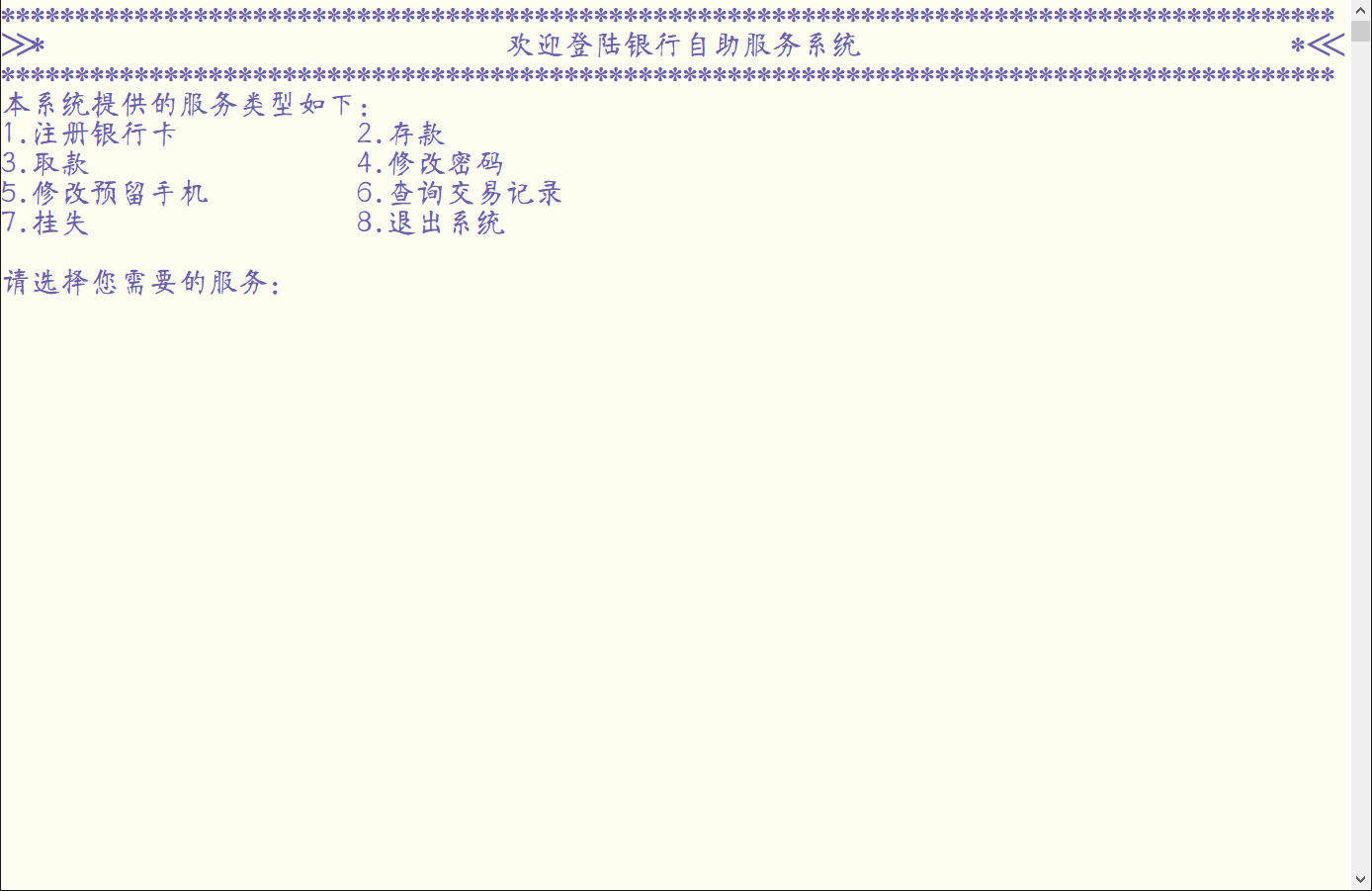
打印账户当前信息（余额等）

}

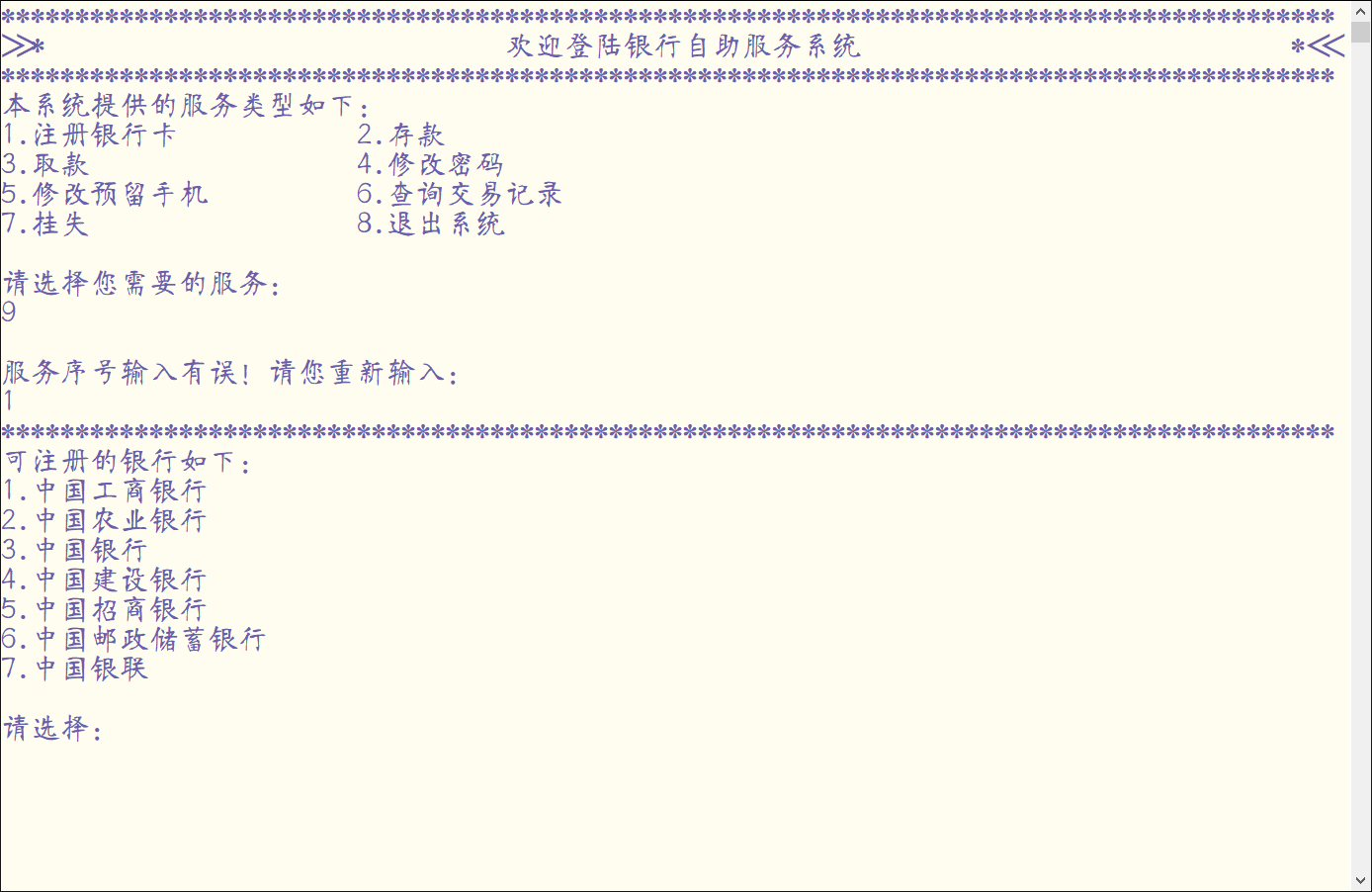
}

1. 性能测试

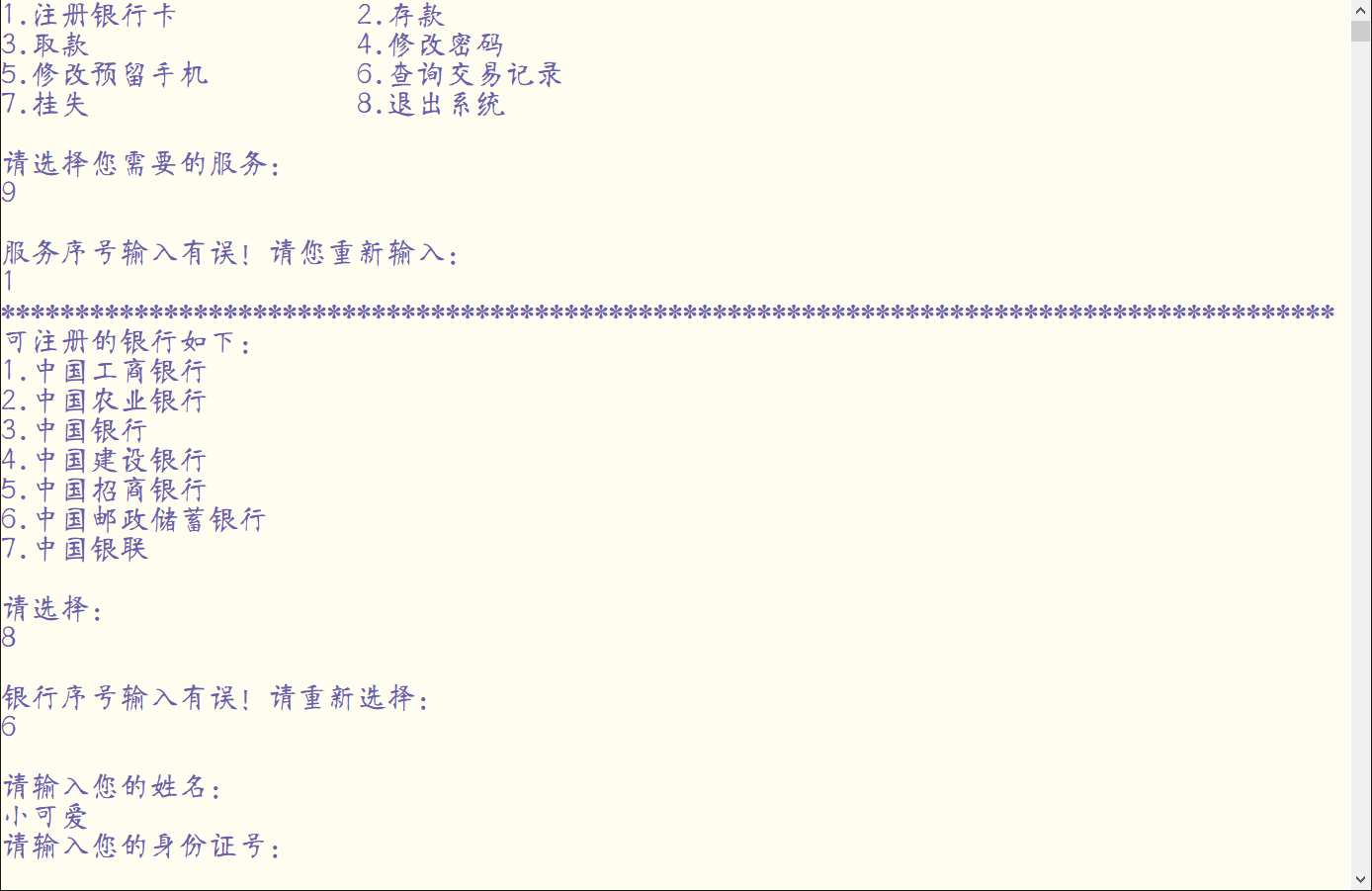
运行程序，界面如下：



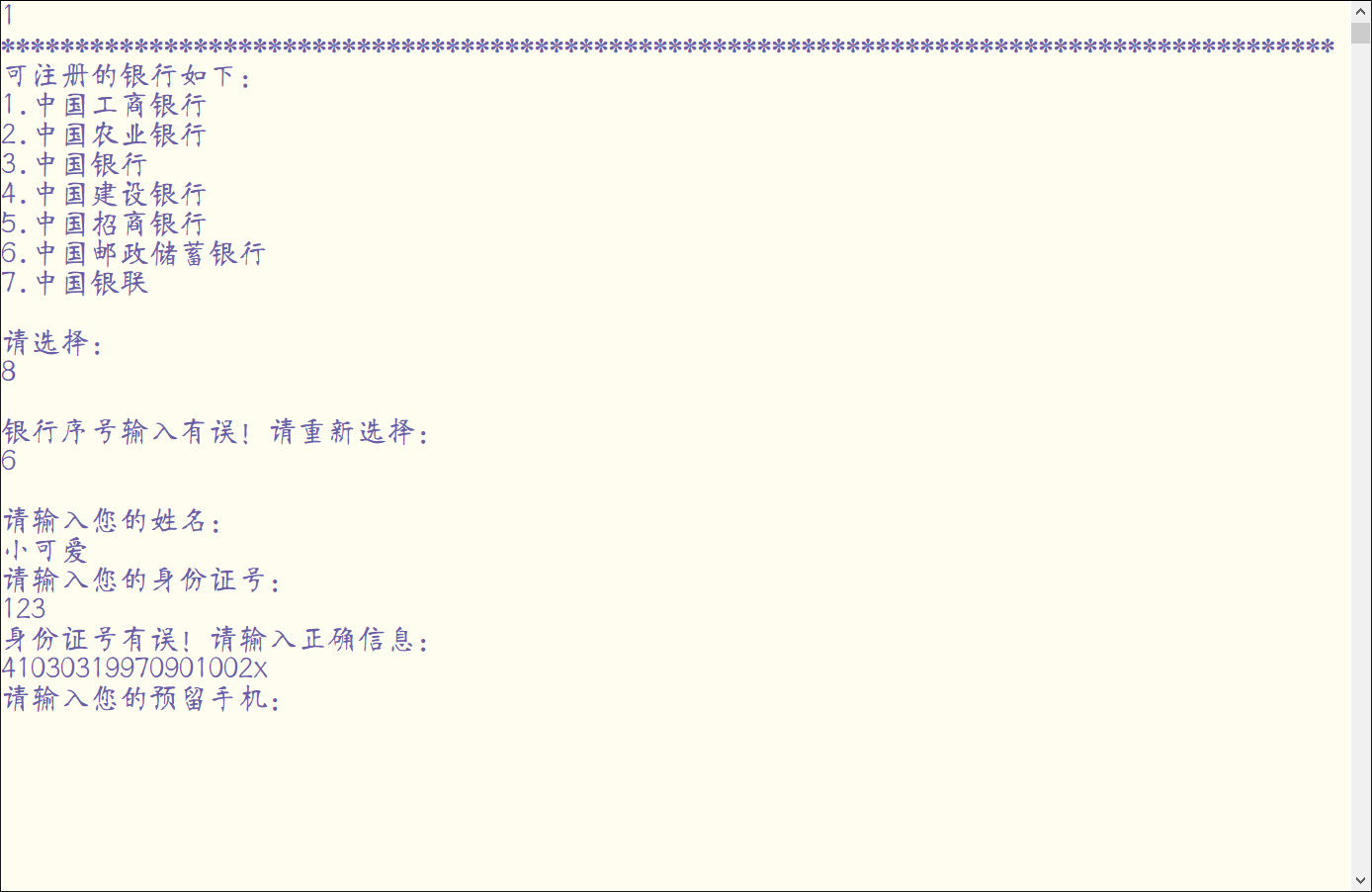
通过输入服务序号，选择相应操作。若序号输入有误，显示提示信息，并要求重新输入。接下来，逐步演示各模块功能。首先选择注册银行卡：



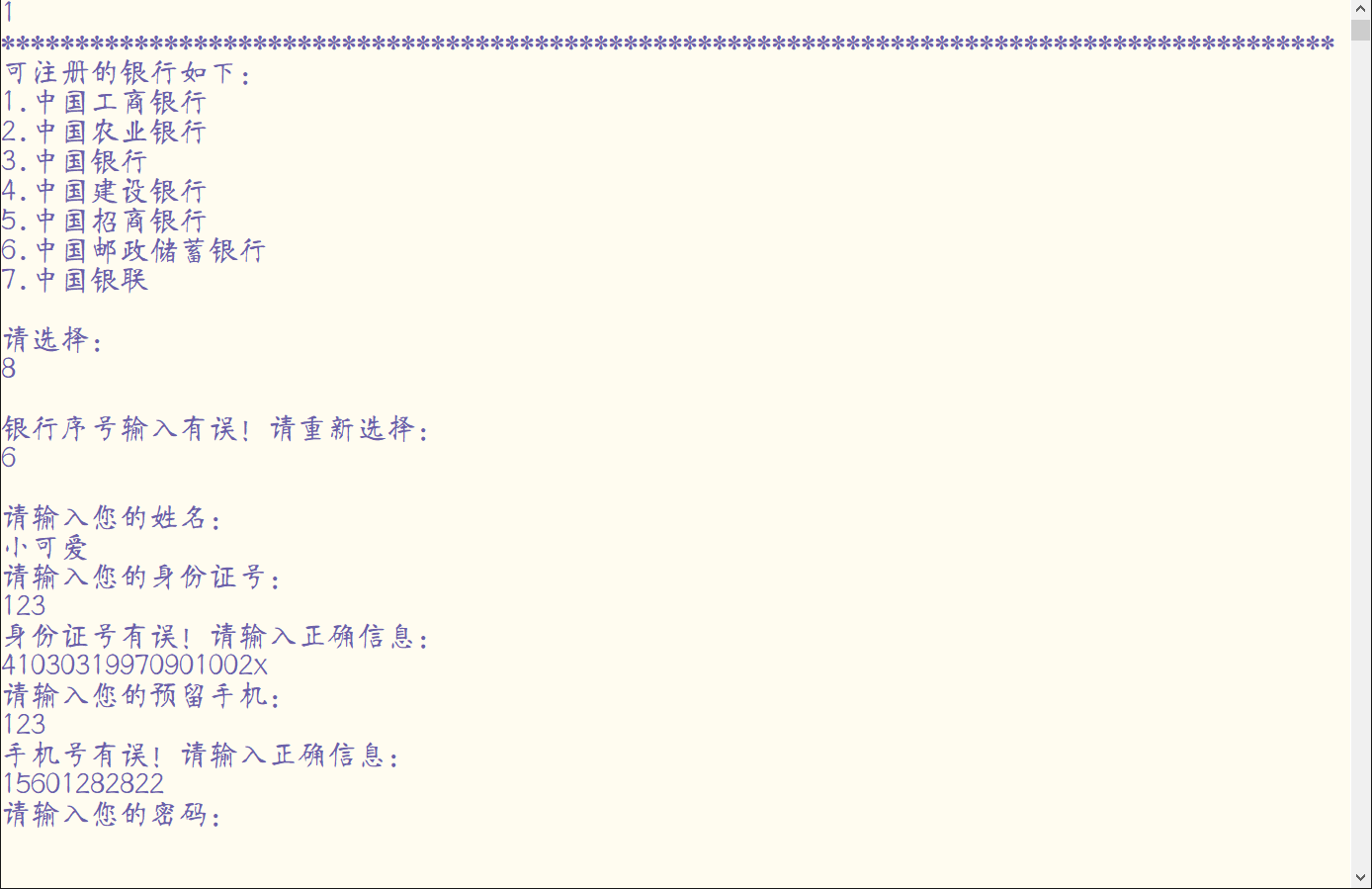
通过输入银行序号，选择所属银行。若序号输入有误，显示提示信息，并要求重新输入：



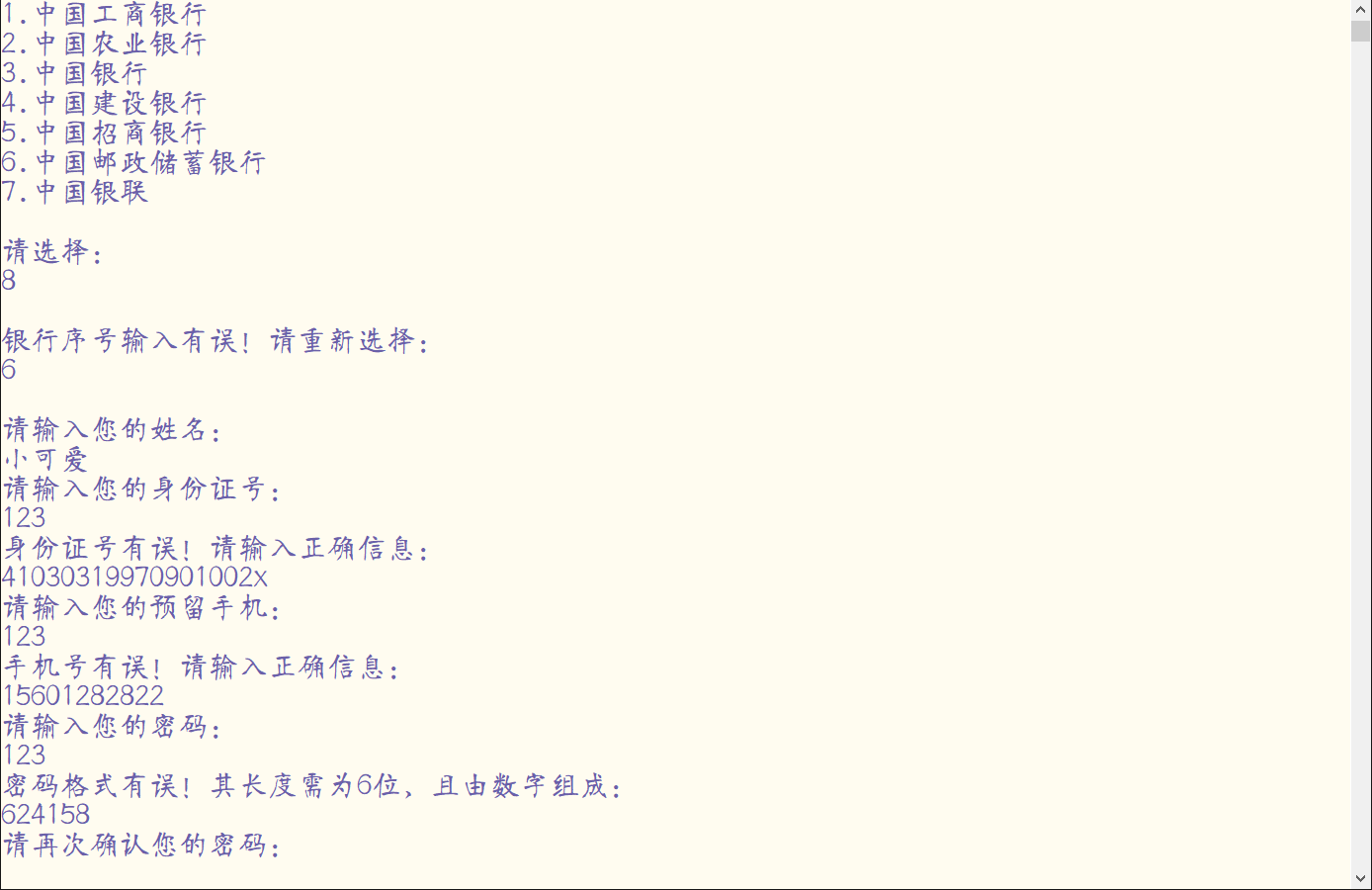
身份证号需为18位，允许字母出现。若输入有误，显示提示信息，并要求重新输入：



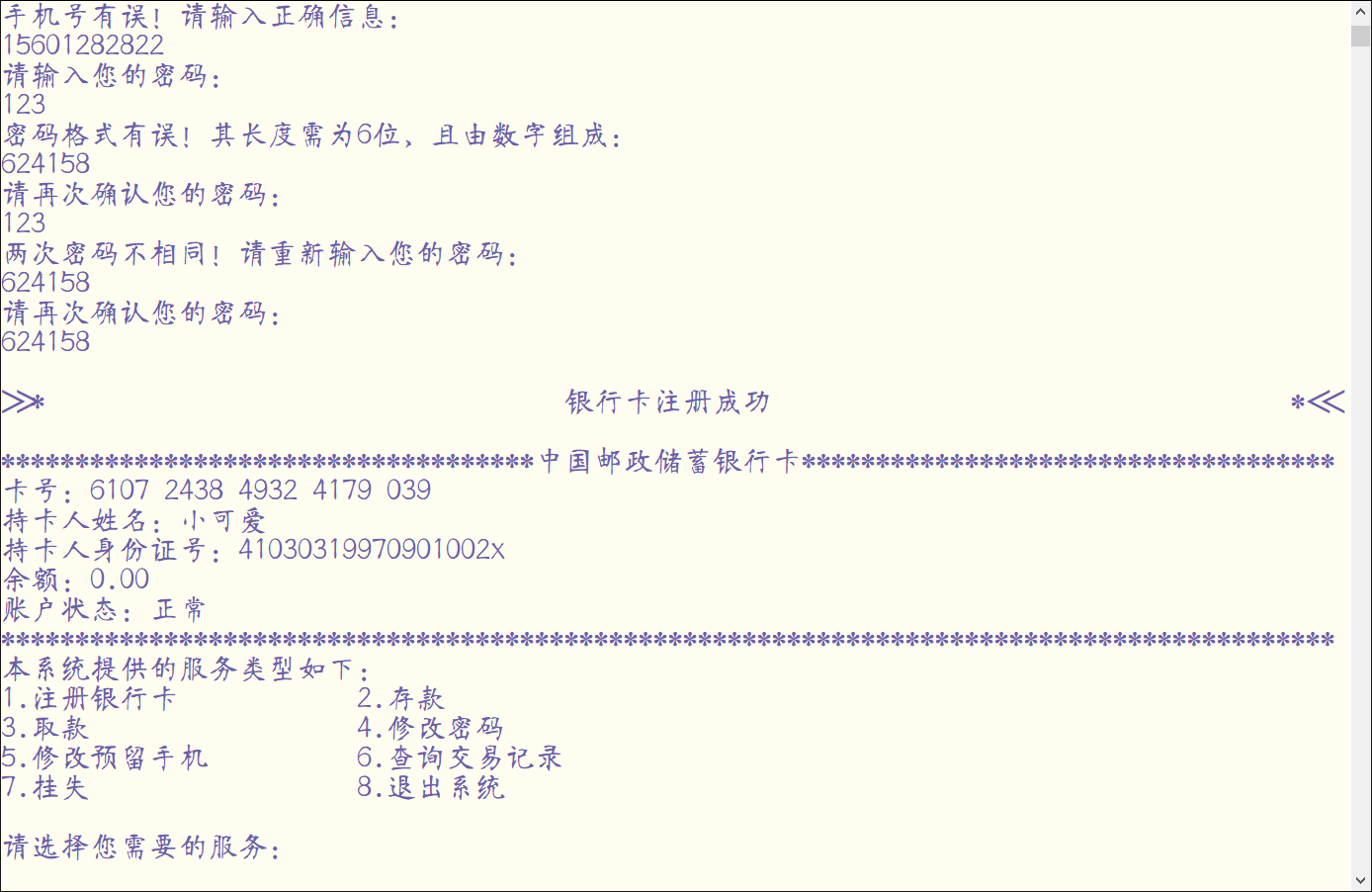
预留手机号需为11位数字。若输入有误，显示提示信息，并要求重新输入：



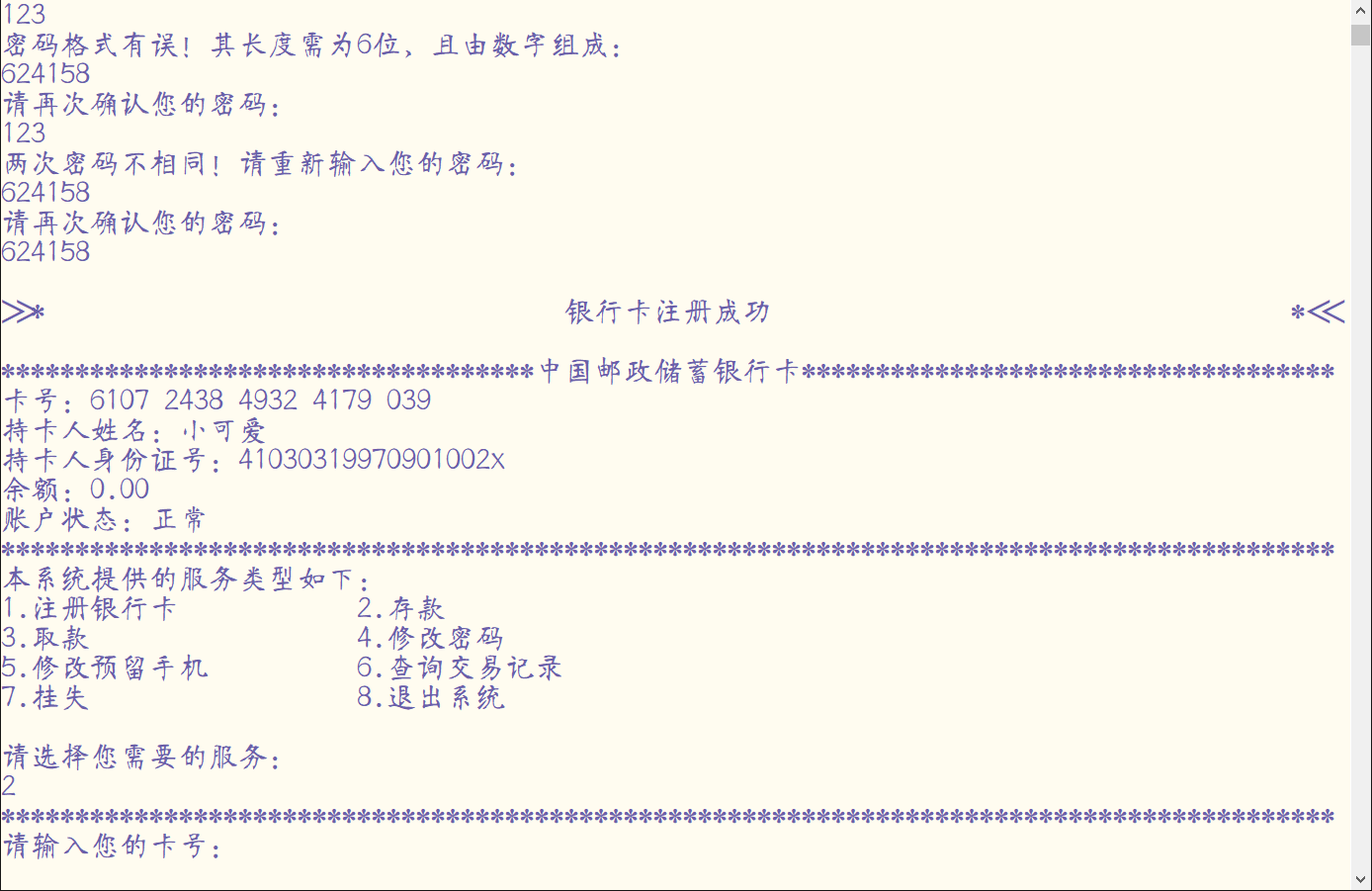
密码需为6位，且仅由数字组成。若输入有误，显示提示信息，并要求重新输入：



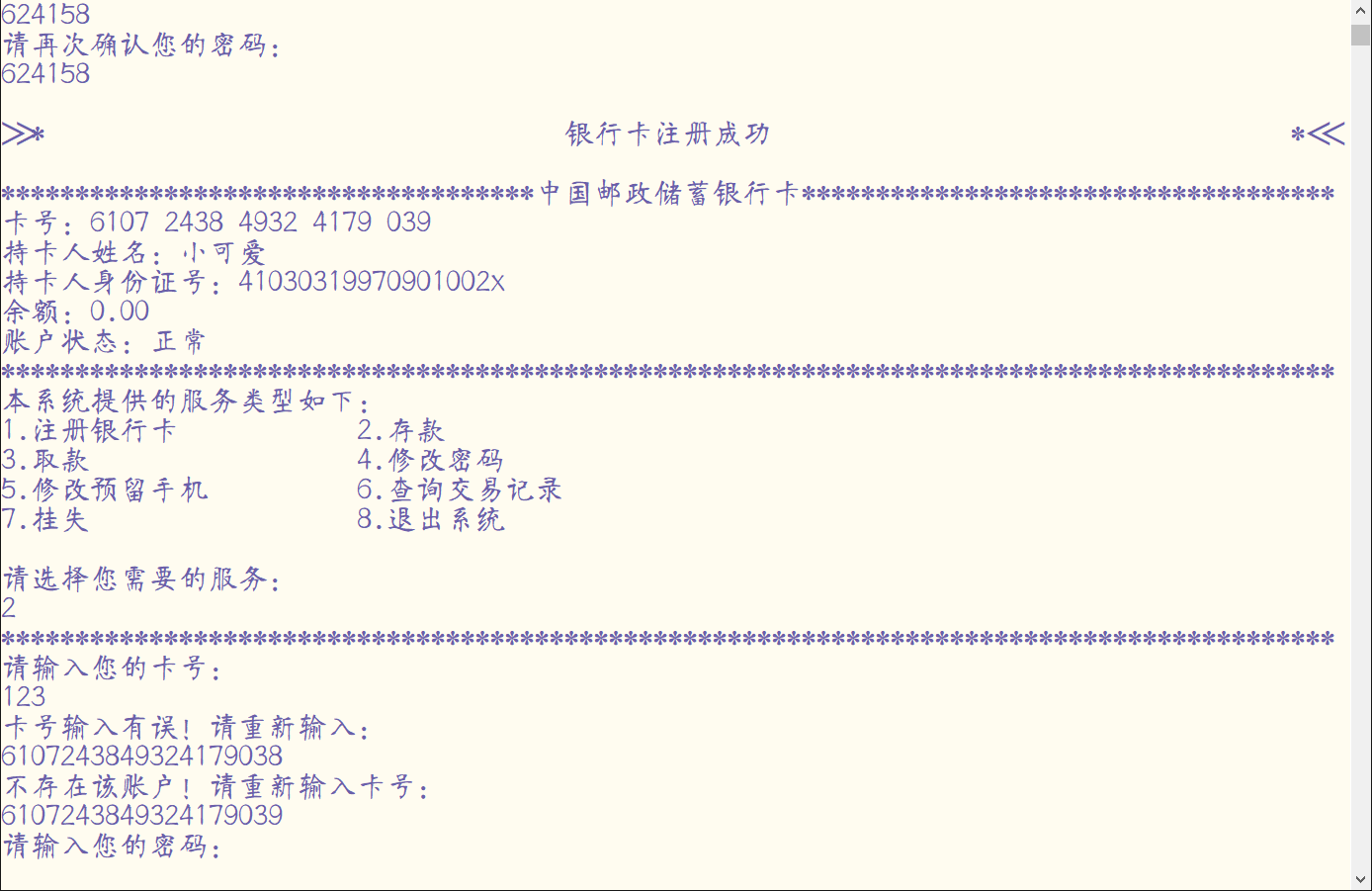
确认密码需与原密码相同，否则显示提示信息，并要求重新设置密码：



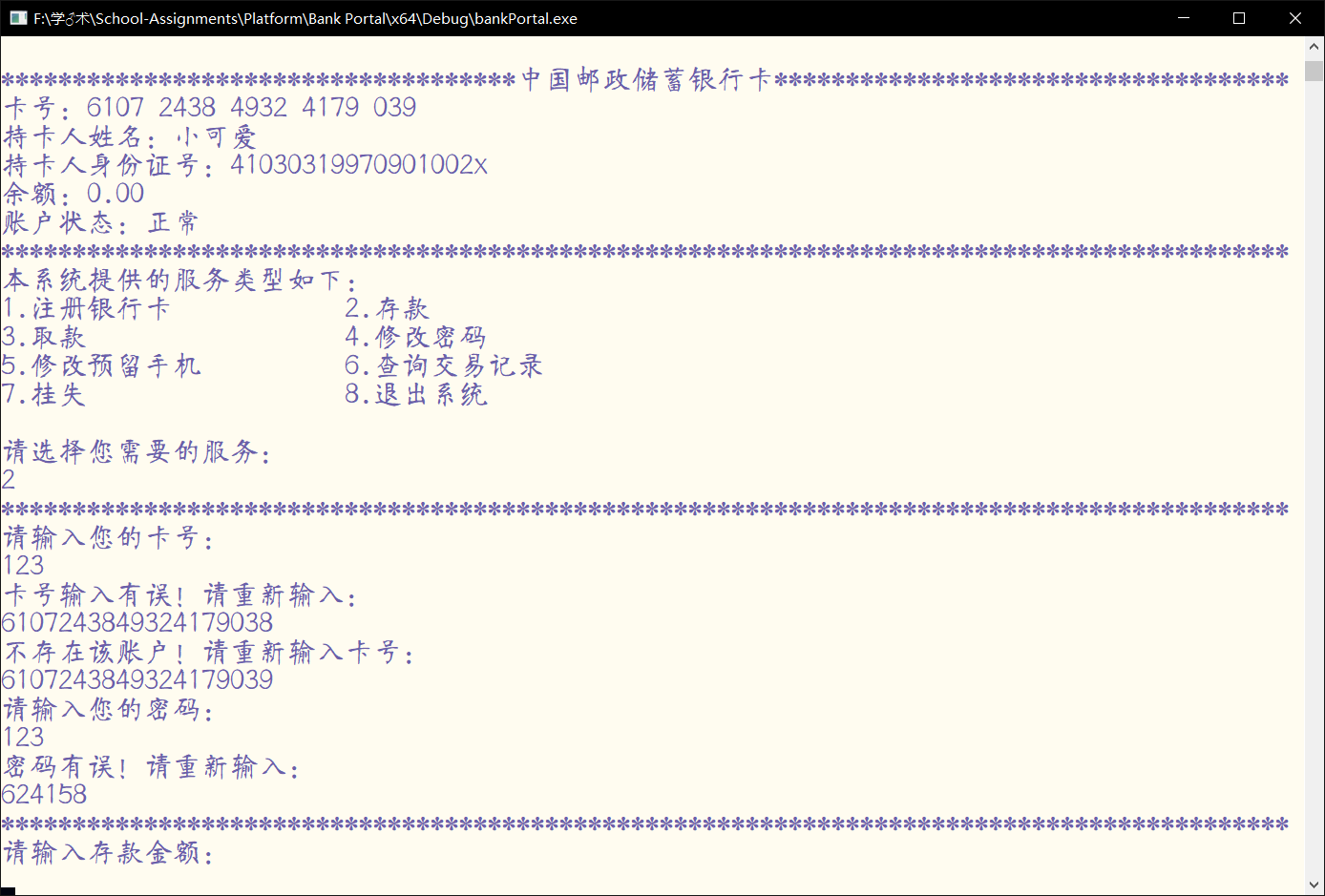
银行卡注册成功。对于存款、取款、修改密码、修改预留手机、查询交易记录、挂失操作，若为初次进入系统，均需输入卡号及密码，若已进行过某一模块的操作，则无需再输入银行卡信息。接下来以存款为例：



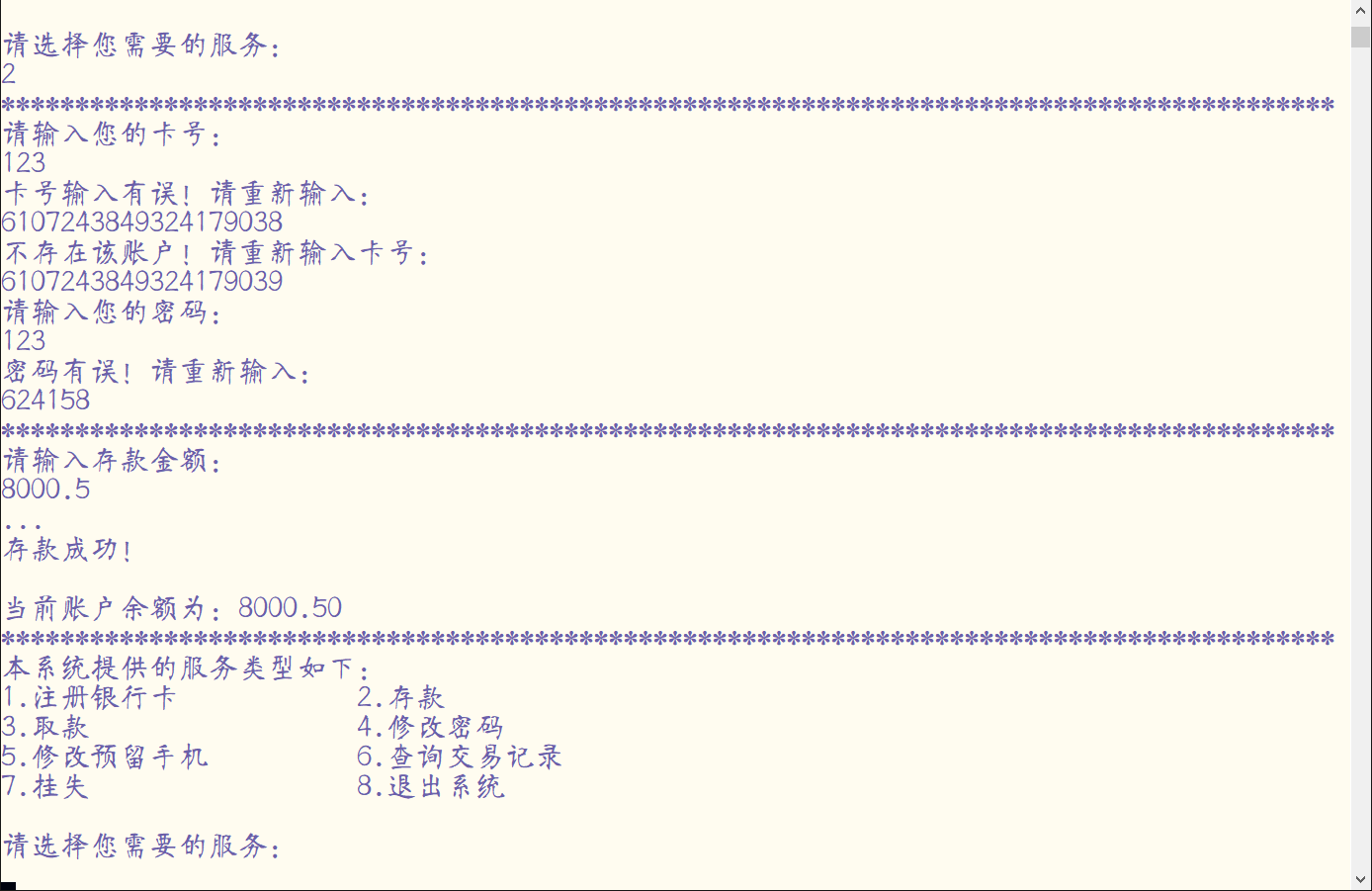
需正确输入卡号。若卡号不存在或形式不正确，显示相应提示信息，并要求重新输入：



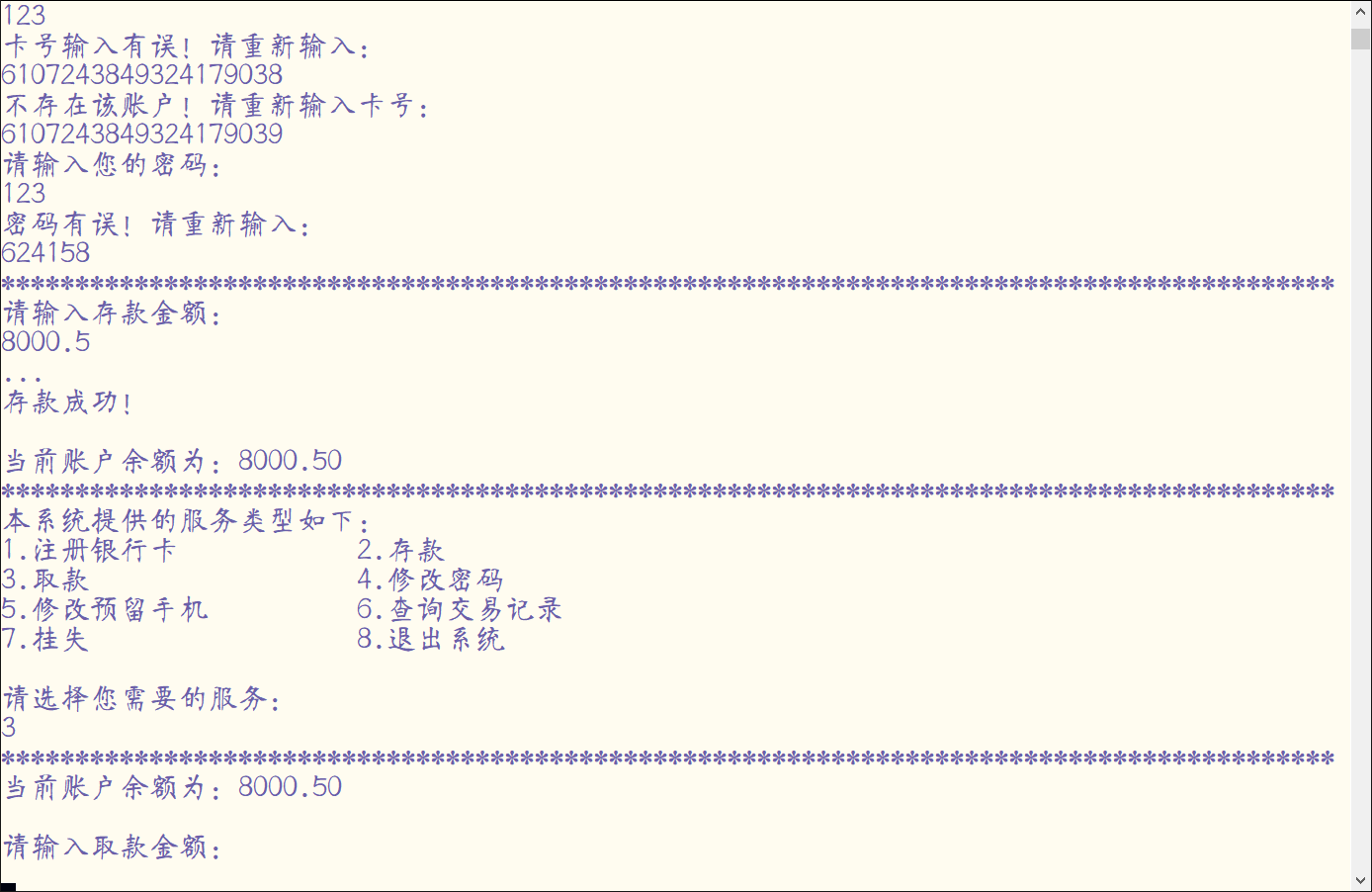
需输入正确密码。若密码不正确，显示提示信息，并要求重新输入：



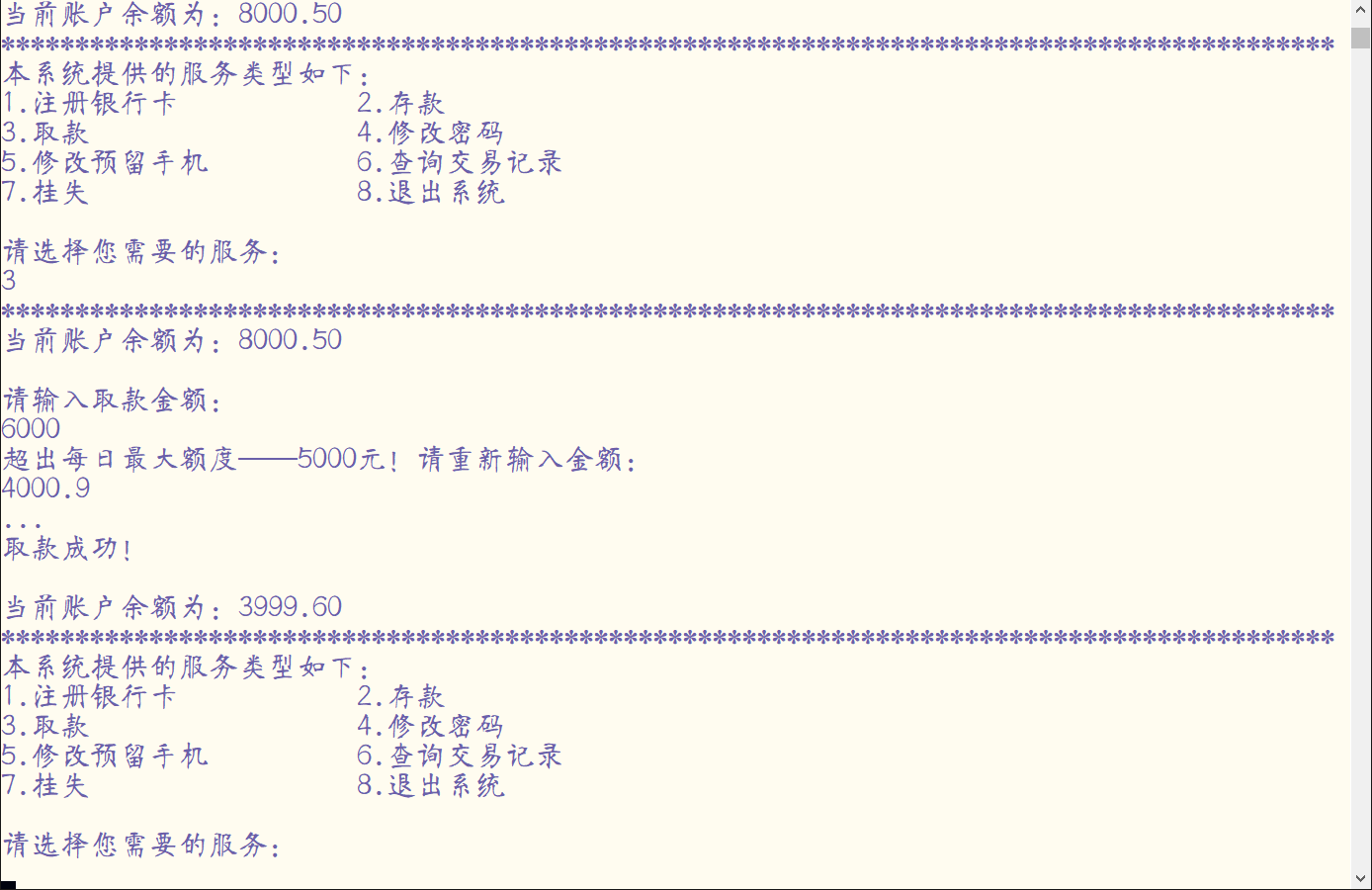
存款金额需为正的小数或整数。若输入有误，显示提示信息，并要求重新输入：



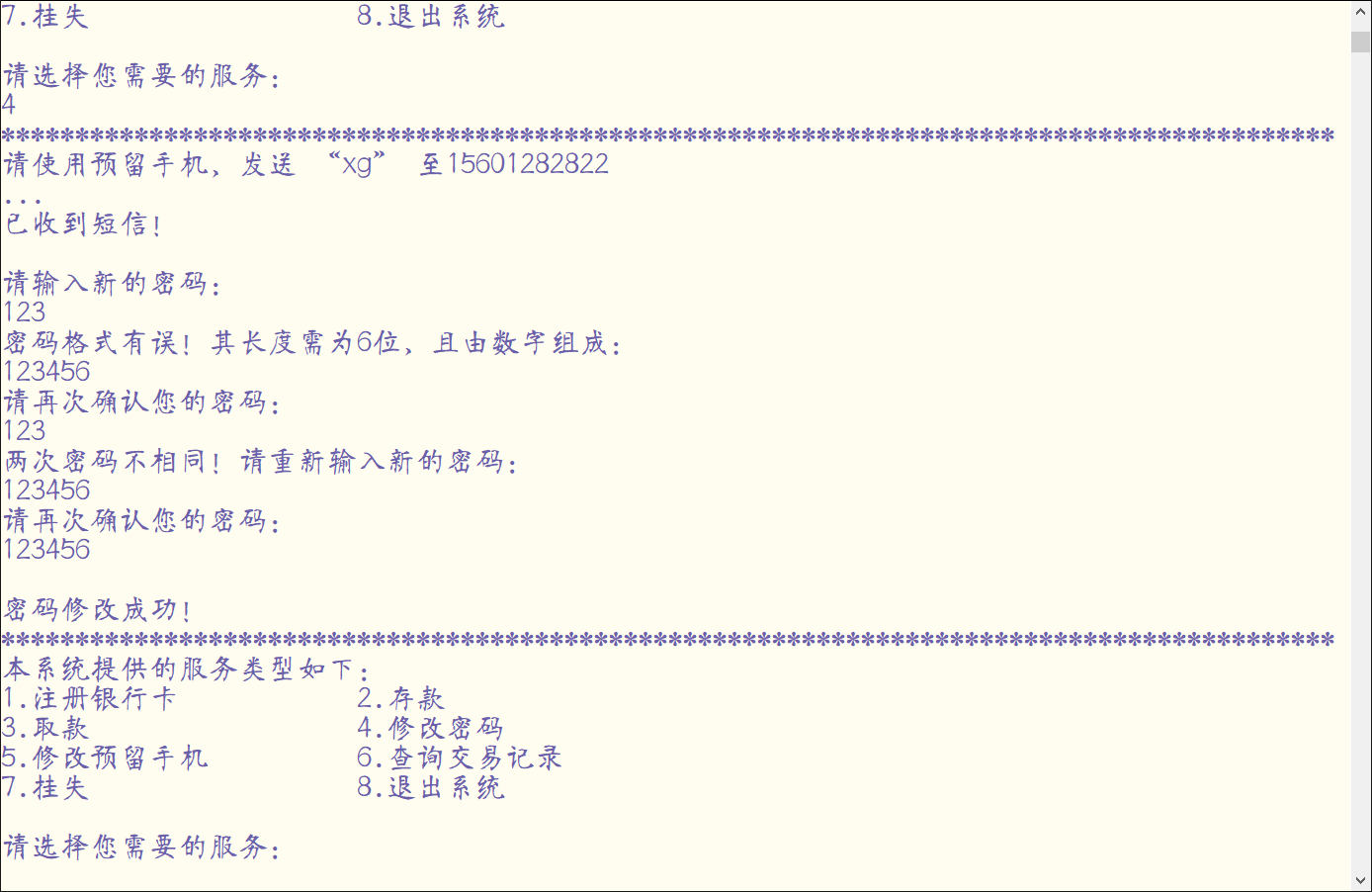
接下来选择取款：



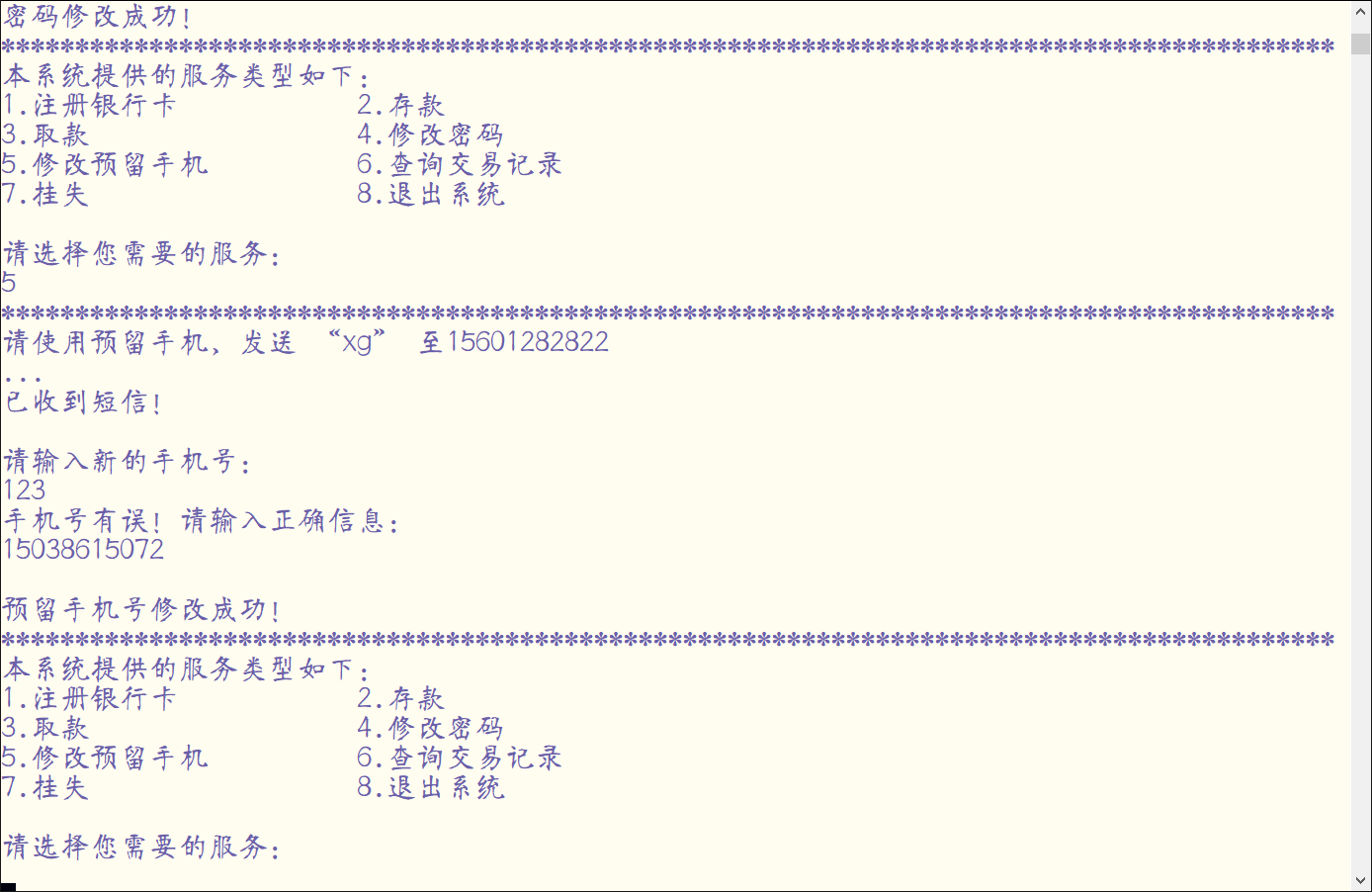
取款金额需为正的小数或整数，且不大于每日最大额度（5000元）和账户余额的较小值，否则显示提示信息，并要求重新输入：



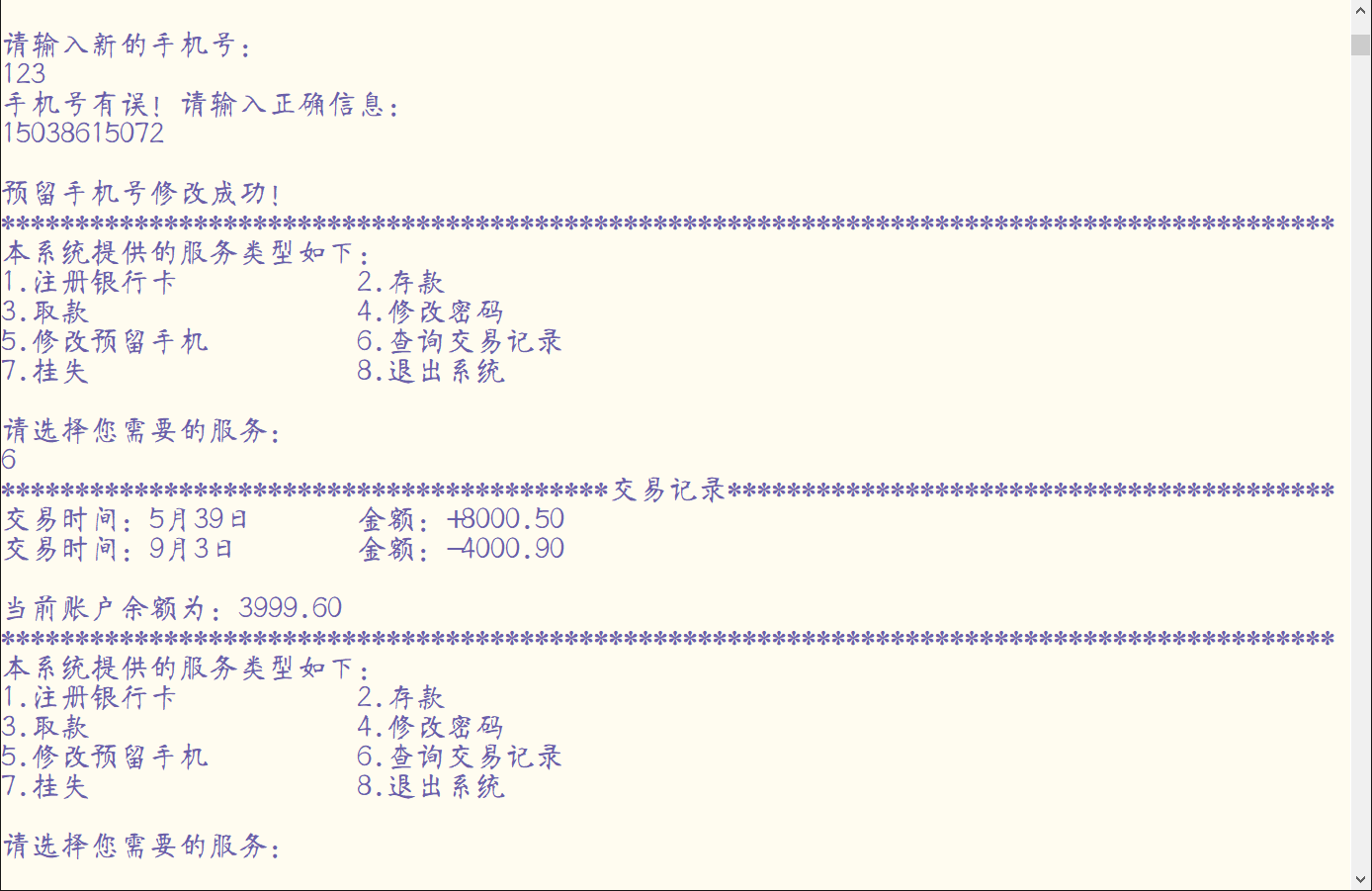
接下来选择修改密码，其过程与注册时的设置密码类似，因而简要演示：



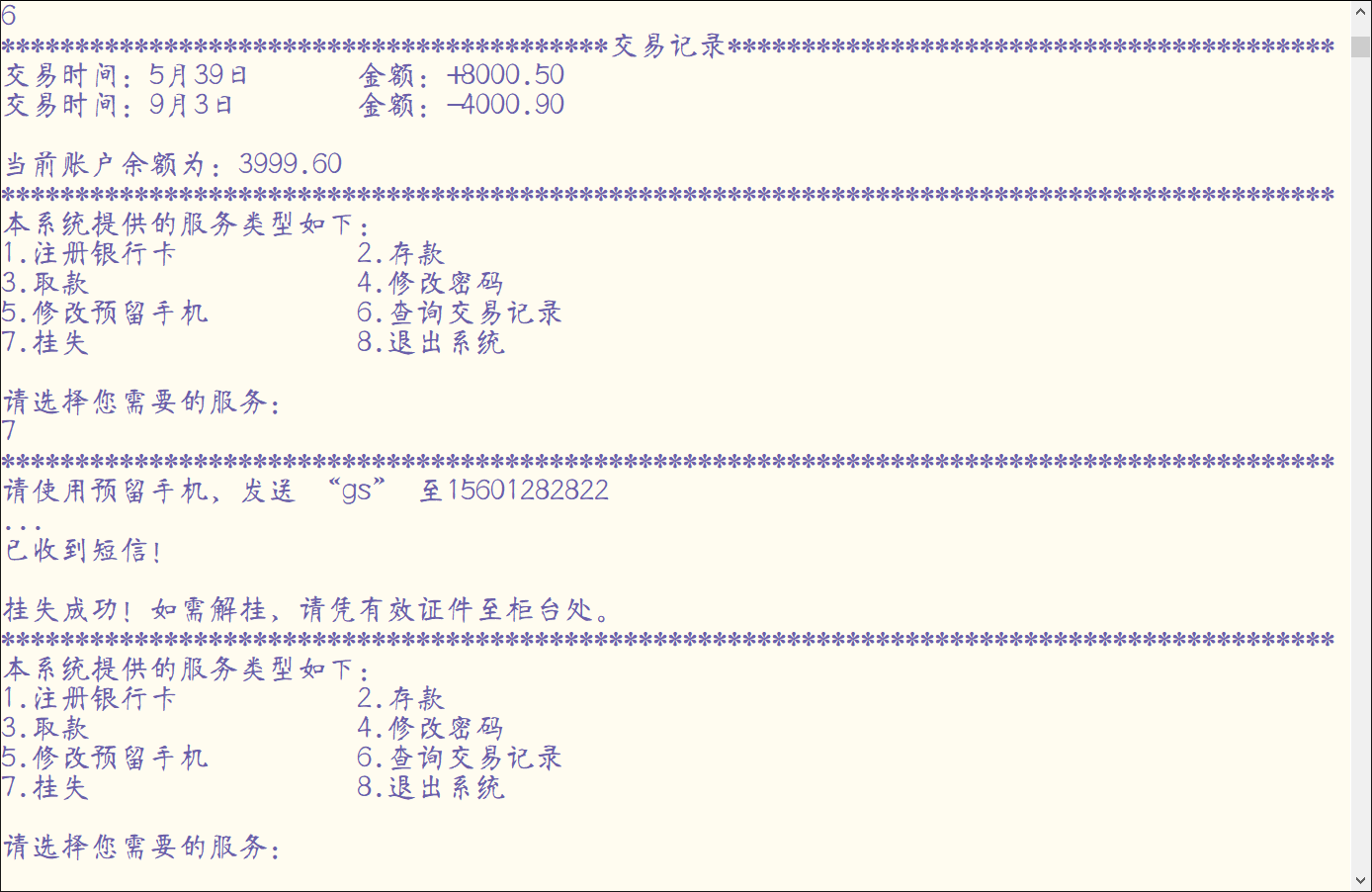
接下来选择修改预留手机，其过程与注册时的设置预留手机类似，因而简要演示：



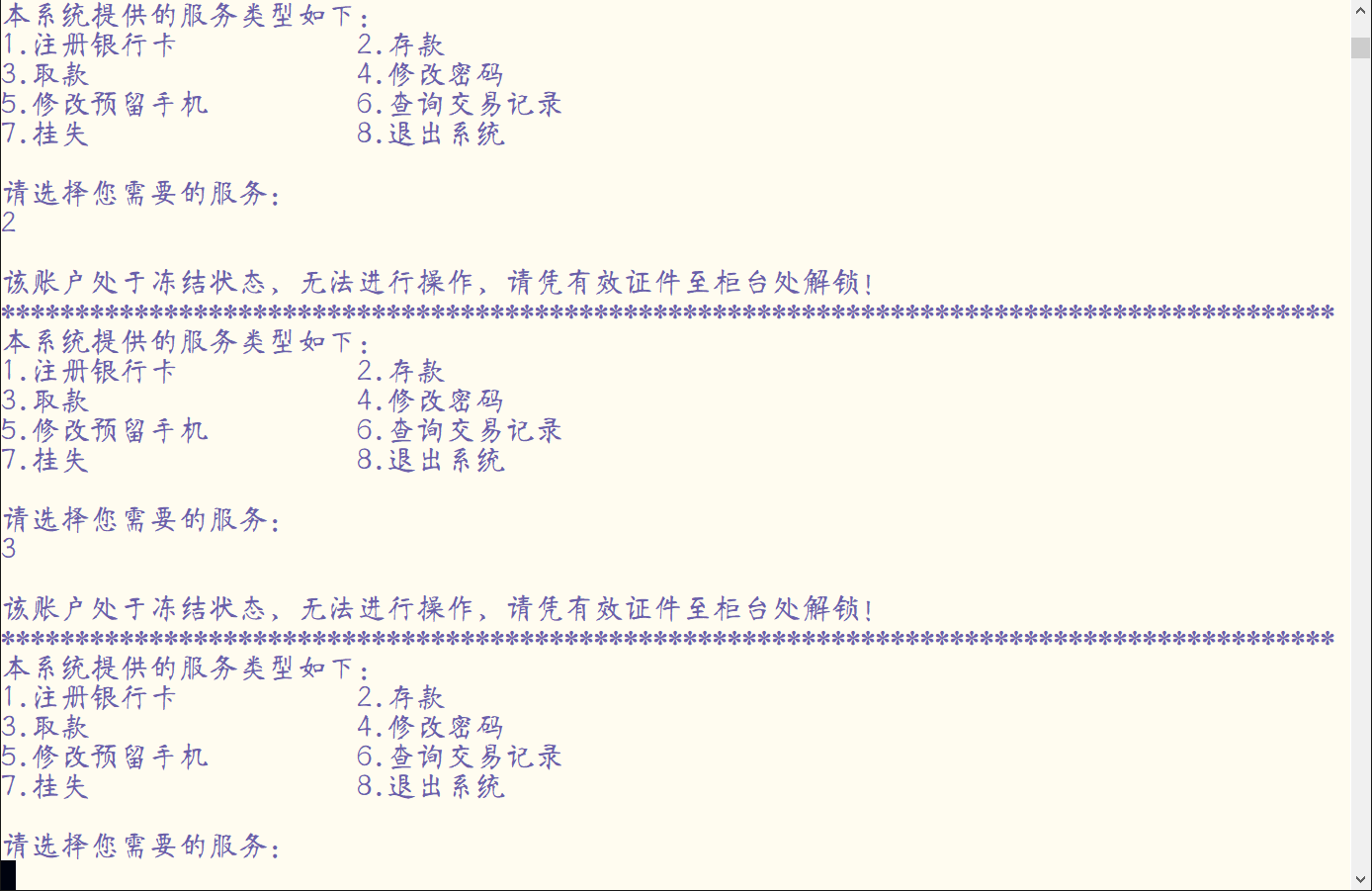
接下来选择查询交易记录：



接下来选择挂失：



一旦挂失成功，2~7操作均无法进行：



以上即为程序功能的大体演示。

电商平台

1. 需求分析

该平台是银行门户之外的一个单独的程序，要求至少支持以下功能：

1. 注册&登录：支持新用户注册平台账号，已注册用户用平台账号登录平台。（要求已注册用户的信息长久保留）；
2. 浏览平台信息；
3. 优惠活动：支持对同一品类所有产品打折的活动，支持单笔订单满X减Y的活动；
4. 购买产品：支持用户添加产品到购物车，查看实际应付的产品价格，提交订单；

附加要求：

1. 电商平台上至少有三类产品：如食物、服装、图书等，每类产品中至少有三个具体的产品，每个具体的产品请至少包含产品描述、产品原价、产品剩余量等数据。所有的产品信息需要存储在数据库或文件中，不能写在代码中，平台管理员通过直接修改数据库或文件，管理本平台上的产品，包括产品的增加和删除，修改数量以及具体产品的属性信息等。
2. 请至少设计一层继承体系（产品基类-产品子类），设计一个产品基类，然后让图书类、电子产品类和服装类等产品子类继承它，具体的产品是产品子类的实例对象。产品基类请至少具体有一个虚函数getPrice()，用于计算具体产品的价格；
3. 请通过为每个产品子类定义“品类折扣系数”来支持对同产品子类下所有产品打折扣的活动（如图书全场5折，则图书类这一产品子类的折扣系数为0.5）；
4. 请做好错误场景的处理。
5. 软件改进

由于时间较为紧迫，银行门户有点亟待实现：

其一，文件操作的改进。数据库非本学期课程，在专业选修JAVA和Python的压力下没有抽出时间进行相关的自学。C++对于文本文件的直接操作较C方便许多，但仍繁琐且效率不高。若搭载在数据库的框架下，效率会有显著提升，代码量也会相应降低，需改进；

其二，用户界面的美化。银行门户的设计用时不多，功能增添较少。本次大作业重心在电商平台的设计，基于实际和人性化的考虑，加入较多额外功能。

其三，解挂失的处理。