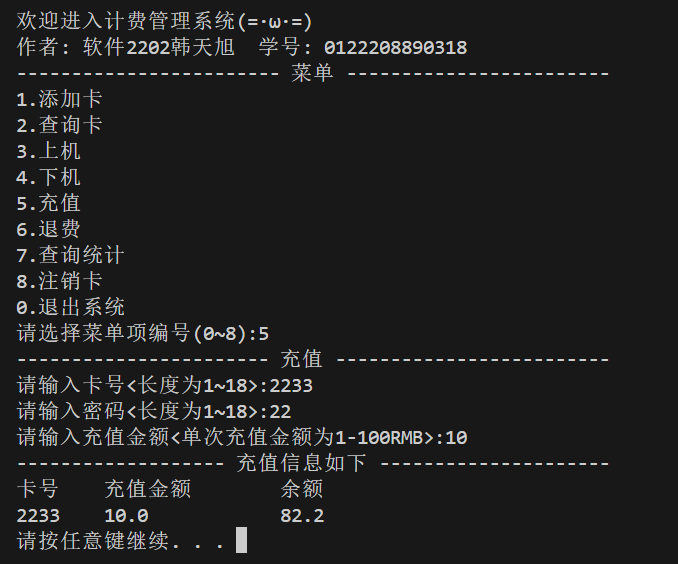
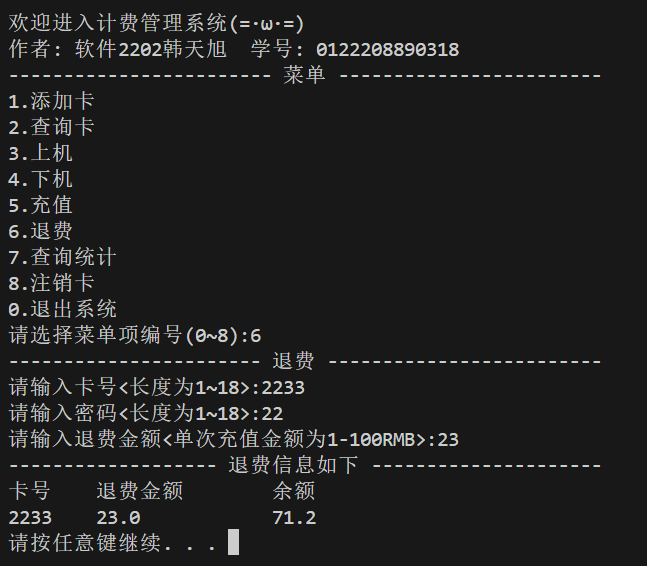
测试用例:

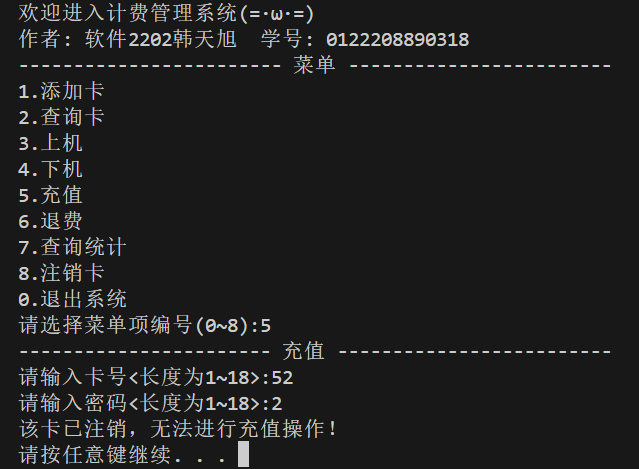
充值，卡号正常/在上机：



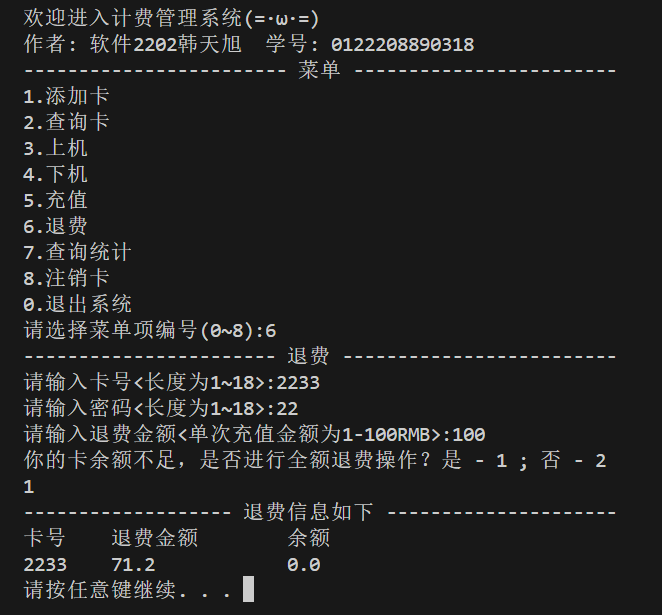
退费，卡号正常/在上机/余额大于0:



充值,已注销:



退费,余额不足:



源代码：

main.c

#include <stdio.h>

#include "service.h"

#include "card\_service.h"

#include "billing\_service.h"

#include "record\_service.h"

#include "menu.h"

#include "tools.h"

int main()

{

    Cards c;

    Billings b;

    Charges m;

    initSystem(&c,&b,&m);

    char ipt[10];

    int choice = 9;

    do

    {

        system("cls");

        showMenu();

        do

        {

            scanf("%s", ipt);

        } while (ipt==NULL);

        choice = ipt[0] - '0';

        if (0 <= choice && choice <= 8)

        {

            switch (choice)

            {

                case 1:

                    ReadCard(&c);

                    break;

                case 2:

                    SearchCard(&c);

                    break;

                case 3:

                    LogIn(&c,&b);

                    break;

                case 4:

                    LogOff(&c,&b);

                    break;

                case 5:

                    InsertMoney(&c,&m);

                    break;

                case 6:

                    GetMoney(&c,&m);

                    break;

                case 7:

                    ShowStatistics();

                    break;

                case 8:

                    SignOff(&c);

                    break;

                case 0:

                    Exit(&c,&b,&m);

                    break;

                default:

                    showDefault();

                    break;

            }

        }

        else

        {

            choice=9;

            showDefault();

        }

    } while (choice);

    return 0;

}

menu.c

#include "menu.h"

void showMenu()

{

    printf("欢迎进入计费管理系统(=·ω·=) \n\

作者: 软件2202韩天旭  学号: 0122208890318 \n\

------------------------ 菜单 ------------------------ \n\

1.添加卡 \n\

2.查询卡 \n\

3.上机 \n\

4.下机 \n\

5.充值 \n\

6.退费 \n\

7.查询统计 \n\

8.注销卡 \n\

0.退出系统 \n\

请选择菜单项编号(0~8):");

}

menu.h

#pragma once

#include <stdio.h>

void showMenu();

service.c

#include <stdio.h>

#include "service.h"

#include "tools.h"

void ShowStatistics()

{

    printf("--------------------- 查询统计 -----------------------\n");

    system("pause");

}

void SignOff(Cards \*c)

{

    printf("----------------------- 注销卡 -----------------------\n");

    char id[MAXLEN];

    InputID(id);

    Card\* cd;

    if((cd=getCard(c,id))==NULL)

    {

        printf("没有找到对应的用户！\n");

        system("pause");

        return ;

    }

    char pwd[MAXLEN];

    InputPWD(pwd);

    if(strcmp(cd->aPwd,pwd)==0)

    {

        printf("注销操作不可恢复,请确认是否执行？");

        printf("是 - 1 ; 否 - 2\n");

        int c=0;

        while(c!=1 && c!=2)

        {

            char ch[4];

            scanf("%s",ch);

            c = ch[0]-'0';

            if(c==1 || c==2) break;

            printf("请重新输入！\n");

        }

        if(c==2) return ;

        if(c==1 && cd->fBalance>0)

        {

            cd->nStatus = 2;

            printf("------------------- 注销信息如下 ---------------------\n");

            printf("卡号\t退款金额\n");

            printf("%s\t%.1lf\n",cd->aId,cd->fBalance);

        }

        else if(c==1)

        {

            cd->nStatus = 2;

            printf("------------------- 注销信息如下 ---------------------\n");

            printf("卡号\n");

            printf("%s\n",cd->aId);

        }

    }

    else

    {

        printf("密码错误!\n");

    }

    system("pause");

}

void Exit(Cards \*c,Billings \*b,Charges \*m)

{

    printf("正在保存数据ing。。。\n");

    save(c,b,m);

    printf("感谢你使用计费管理系统\n");

    system("pause");

}

void showDefault()

{

    printf("输入有误，请重新输入\n");

    system("pause");

}

service.h

#pragma once

#include <stdio.h>

#include <time.h>

#include <string.h>

#include <stdlib.h>

#define MAXLEN 18

#define SIZE 100

typedef struct card

{

    char aId[MAXLEN];

    char aPwd[MAXLEN];

    int nStatus;

    time\_t tStart;

    time\_t tEnd;

    float fTotalUse;

    time\_t tLast;

    int nUseCount;

    float fBalance;

    int nDel;

    struct card\* next;

    struct card\* next1;

} Card;

typedef struct cards

{

    Card \*head,\*tail;

}Cards;

typedef struct Billing

{

    char aCardName[18];

    time\_t tStart;

    time\_t tEnd;

    float fAmount;

    int nStatus;

    int nDel;

    struct Billing\* next;

}Billing;

typedef struct Billings

{

    Billing \*head,\*tail;

}Billings;

typedef struct LoginInfo

{

    char aCardName[18];

    time\_t tLogin;

    float fBalance;

}LoginInfo;

typedef struct SettleInfo

{

    char aCardName[18];

    time\_t tStart;

    time\_t tEnd;

    float fAmount;

    float fBalance;

}SettleInfo;

typedef struct Charge

{

    char aCardName[18]; // 卡号

    time\_t tTime; // 充值退费的时间

    int nStatus; // 状态：0-表示充值；1-表示退费

    float fMoney; // 充值退费金额

    int nDel; // 删除标识，0-未删除,1-删除

    struct Charge \*next;

}Charge;

typedef struct Charges

{

    Charge \*head,\*tail;

}Charges;

void ShowStatistics();

void Exit(Cards \*c,Billings \*b,Charges \*m);

void showDefault();

record\_service.c

#include "record\_service.h"

#include "tools.h"

void InsertMoney(Cards\* c,Charges\* m)

{

    printf("----------------------- 充值 -------------------------\n");

    char id[MAXLEN];

    InputID(id);

    Card\* cd;

    if((cd=getCard(c,id))==NULL)

    {

        printf("没有找到对应的用户！\n");

        system("pause");

        return ;

    }

    char pwd[MAXLEN];

    InputPWD(pwd);

    if(strcmp(cd->aPwd,pwd)==0)

    {

        if(cd->nStatus == 2)

        {

            printf("该卡已注销，无法进行充值操作！\n");

            system("pause");

            return ;

        }

        float balance=0;

        while(1)

        {

            printf("请输入充值金额<单次充值金额为1-100RMB>:");

            scanf("%f",&balance);

            if(balance<1 || balance>100)

                printf("数据非法，请重新输入!\n");

            else

                break;

        }

        Charge\* im = CreateCharge(id,0,balance);

        InsertCharge(m,im);

        time(&(cd->tEnd));

        cd->fBalance+=balance;

        printf("------------------- 充值信息如下 ---------------------\n");

        printf("卡号\t充值金额\t余额\n");

        printf("%s\t%.1lf\t\t%.1lf\n",cd->aId,balance,cd->fBalance);

    }

    else

    {

        printf("密码错误!\n");

    }

    system("pause");

}

void GetMoney(Cards\* c,Charges\* m)

{

    printf("----------------------- 退费 -------------------------\n");

    char id[MAXLEN];

    InputID(id);

    Card\* cd;

    if((cd=getCard(c,id))==NULL)

    {

        printf("没有找到对应的用户！\n");

        system("pause");

        return ;

    }

    char pwd[MAXLEN];

    InputPWD(pwd);

    if(strcmp(cd->aPwd,pwd)==0)

    {

        if(cd->fBalance <= 0)

        {

            printf("你的卡余额不足，无法进行退费操作!\n");

            system("pause");

            return ;

        }

        float balance=0;

        while(1)

        {

            printf("请输入退费金额<单次充值金额为1-100RMB>:");

            scanf("%f",&balance);

            if(balance<1 || balance>100)

                printf("数据非法，请重新输入!\n");

            else

                break;

        }

        if(cd->fBalance - balance <= 0)

        {

            printf("你的卡余额不足，是否进行全额退费操作？");

            printf("是 - 1 ; 否 - 2\n");

            int c=0;

            while(c!=1 && c!=2)

            {

                char ch[4];

                scanf("%s",ch);

                c = ch[0]-'0';

                if(c==1 || c==2) break;

                printf("请重新输入！\n");

            }

            if(c==1) balance = cd->fBalance;

            if(c==2) return ;

        }

        Charge\* im = CreateCharge(id,1,balance);

        InsertCharge(m,im);

        time(&(cd->tEnd));

        cd->fBalance-=balance;

        printf("------------------- 退费信息如下 ---------------------\n");

        printf("卡号\t退费金额\t余额\n");

        printf("%s\t%.1lf\t\t%.1lf\n",cd->aId,balance,cd->fBalance);

    }

    else

    {

        printf("密码错误!\n");

    }

    system("pause");

}

record\_service.h

#pragma once

#include "service.h"

void InsertMoney(Cards\* c,Charges\* m);

void GetMoney(Cards\* c,Charges\* m);

billibg\_service.c

#include "billing\_service.h"

#include "tools.h"

void LogIn(Cards\* c,Billings\* b)

{

    printf("----------------------- 上机 -------------------------\n");

    char id[MAXLEN];

    InputID(id);

    Card\* cd;

    if((cd=getCard(c,id))==NULL)

    {

        printf("没有找到对应的用户！\n");

        system("pause");

        return ;

    }

    char pwd[MAXLEN];

    InputPWD(pwd);

    if(strcmp(cd->aPwd,pwd)==0)

    {

        if(cd->fBalance<0)

        {

            printf("您的卡已欠费，请到工作人员处处理\n");

            printf("欠费金额: %.1lf\n",-(cd->fBalance));

            system("pause");

            return ;

        }

        if(cd->nStatus==1)

        {

            printf("你在上机呢!\n");

            system("pause");

            return ;

        }

        printf("上机ing...\n");

        cd->nStatus=1;

        ++cd->nUseCount;

        Billing\* newB = CreateBilling(id);

        cd->tLast = newB->tStart;

        InsertBilling(b,newB);

        LoginInfo lgif;

        strcpy(lgif.aCardName,id);

        lgif.tLogin = newB->tStart;

        lgif.fBalance = cd->fBalance;

        char time[40];

        time\_to\_string(lgif.tLogin,time,sizeof(time));

        printf("------------------- 上机信息如下 ---------------------\n");

        printf("卡号\t余额\t上机时间\n");

        printf("%s\t%.1f\t%s\n",lgif.aCardName,lgif.fBalance,time);

    }

    else

    {

        printf("密码错误!\n");

    }

    system("pause");

}

void LogOff(Cards\* c,Billings\* b)

{

    printf("----------------------- 下机 -------------------------\n");

    char id[MAXLEN];

    InputID(id);

    Card\* cd;

    if((cd=getCard(c,id))==NULL)

    {

        printf("没有找到对应的用户！\n");

        system("pause");

        return ;

    }

    char pwd[MAXLEN];

    InputPWD(pwd);

    if(strcmp(cd->aPwd,pwd)==0)

    {

        if(cd->nStatus==0)

        {

            printf("你没上机呢!\n");

            system("pause");

            return ;

        }

        Billing\* targetB = searchLoginBill(b,id);

        if(targetB == NULL)

        {

            printf("没有找到对应的消费记录!\n");

            system("pause");

            return ;

        }

        printf("下机ing...\n");

        cd->nStatus=0;

        time(&(targetB->tEnd));

        cd->tLast = targetB->tEnd;

        double price = CalcuPrice(targetB->tStart,targetB->tEnd);

        // printf("price = %.3lf\n",price);

        cd->fTotalUse+=price;

        cd->fBalance-=price;

        targetB->fAmount = price;

        targetB->nStatus = 1;

        char time1[40];

        time\_to\_string(targetB->tStart,time1,sizeof(time1));

        char time2[40];

        time\_to\_string(targetB->tEnd,time2,sizeof(time2));

        if(cd->fBalance>=0)

        {

            printf("------------------- 下机信息如下 ---------------------\n");

            printf("卡号\t消费\t余额\t上机时间\t\t下机时间\n");

            printf("%s\t%.1lf\t%.1lf\t%s\t%s\n",cd->aId,price,cd->fBalance,time1,time2);

        }

        else

        {

            printf("您的卡已欠费，请到工作人员处处理\n");

            printf("欠费金额: %.1lf\n",-(cd->fBalance));

            printf("卡号\t消费\t上机时间\t\t下机时间\n");

            printf("%s\t%.1lf\t%s\t%s\n",cd->aId,price,time1,time2);

        }

    }

    else

    {

        printf("密码错误!\n");

    }

    system("pause");

}

billing\_service.h

#pragma once

#include "service.h"

void LogIn(Cards\* c,Billings\* b);

void LogOff(Cards\* c,Billings\* b);

card\_service.c

#include "card\_service.h"

#include "tools.h"

void ReadCard(Cards \*c)

{

    printf("----------------------- 添加卡 -----------------------\n");

    Card\* newc = (Card\*)malloc(sizeof(Card));

    char id[MAXLEN];

    char pwd[MAXLEN];

    float balance;

    InputID(id);

    if(getCard(c,id)!=NULL)

    {

        printf("这个卡已经有啦\n");

        system("pause");

        return ;

    }

    strcpy(newc->aId,id);

    InputPWD(pwd);

    strcpy(newc->aPwd,pwd);

    while(1)

    {

        printf("请输入业务金额<数值为1~100>:");

        scanf("%f",&balance);

        if(balance<1 || balance>100)

            printf("数据非法，请重新输入!\n");

        else

            break;

    }

    newc->fBalance=balance;

    newc->nStatus=0;

    newc->fTotalUse=0;

    newc->nUseCount=0;

    newc->nDel=0;

    newc->next=NULL;

    newc->next1=NULL;

    time(&(newc->tStart));

    newc->tEnd = newc->tLast = newc->tStart;

    struct tm \* tend = localtime(&newc->tEnd);

    tend->tm\_year += 5;

    newc->tEnd = mktime(tend);

    printf("请确认：\n");

    printf("-----添加卡的信息如下-----\n");

    printf("卡号\t密码\t状态\t开卡金额\n");

    printf("%s\t%s\t%d\t%.1f\n",newc->aId,newc->aPwd,newc->nStatus,newc->fBalance);

    system("pause");

    InsertCard(c,newc);

    printf("开卡成功(=·ω·=)\n");

    system("pause");

}

void SearchCard(Cards\* c)

{

    printf("----------------------- 查询卡 -----------------------\n");

    char id[MAXLEN];

    InputID(id);

    Cards ics;

    int k=vague\_search(c,&ics,id,get\_size(id));

    if(k==0)

    {

        printf("什么也没有找到QAQ\n");

        system("pause");

        return ;

    }

    printf("-----查询到的卡的信息如下-----\n");

    Card\* ic = ics.head;

    while(ic!=ics.tail->next1)

    {

        char time[40];

        time\_to\_string(ic->tLast,time,sizeof(time));

        printf("卡号\t状态\t余额\t累计使用\t使用次数\t上次使用时间\n");

        printf("%s\t%d\t%.1f\t%.1f\t\t%d\t\t%s\n",ic->aId,ic->nStatus,ic->fBalance,ic->fTotalUse,ic->nUseCount,time);

        ic=ic->next1;

    }

    system("pause");

    Card\* ic0 = c->head;

    while(ic0!=NULL)

    {

        ic0->next1=NULL;

        ic0=ic0->next;

    }

}

card\_servicce.h

#pragma once

#include "service.h"

void ReadCard(Cards \*c);

void SearchCard(Cards\* c);

void SignOff(Cards \*c);

tools.c

#include "tools.h"

#include "card\_service.h"

#include "billing\_service.h"

#include "record\_service.h"

//card

int initCard(Cards\* c)

{

    c->tail=NULL;

    c->head=c->tail;

    FILE \*fc;

    if((fc=fopen("card\_file.txt","r"))==NULL)

    {

        return 0;

    }

    int i=0;

    char line[1024];

    char Start[40],End[40],Last[40];

    while(fgets(line,sizeof(line),fc)!=NULL)

    {

        Card\* ic = (Card\*)malloc(sizeof(Card));

        sscanf(line,"%[^##]##%[^##]##%d##%[^##]##%[^##]##%f##%[^##]##%d##%f##%d\n",ic->aId,ic->aPwd,&ic->nStatus,Start,End,&ic->fTotalUse,Last,&ic->nUseCount,&ic->fBalance,&ic->nDel);

        if(ic->nDel==1) continue;

        ic->tStart = (time\_t)string\_to\_time(Start,sizeof(Start)/sizeof(char));

        ic->tEnd = (time\_t)string\_to\_time(End,sizeof(End)/sizeof(char));

        ic->tLast = (time\_t)string\_to\_time(Last,sizeof(Last)/sizeof(char));

        ic->next1=NULL;

        InsertCard(c,ic);

        ++i;

    }

    fclose(fc);

    return i;

}

int saveCard(Cards\* c)

{

    FILE \*fp;

    Card\* ic = c->head;

    if((fp = fopen("card\_file.txt","w")) == NULL)

    {

        printf("卡片数据保存失败！\n");

        return 0;

    }

    if(ic==NULL)

    {

        return 0;

    }

    int i=0;

    while(ic!=NULL)

    {

        ic->next1=NULL;

        char sStart[40];

        char sEnd[40];

        char sLast[40];

        time\_to\_string(ic->tStart,sStart,sizeof(sStart)/sizeof(char));

        time\_to\_string(ic->tEnd,sEnd,sizeof(sEnd)/sizeof(char));

        time\_to\_string(ic->tLast,sLast,sizeof(sLast)/sizeof(char));

        fprintf(fp,"%s##%s##%d##%s##%s##%.1f##%s##%d##%.1f##%d\n",ic->aId,ic->aPwd,ic->nStatus,sStart,sEnd,ic->fTotalUse,sLast,ic->nUseCount,ic->fBalance,ic->nDel);

        Card \*p = ic;

        ic=ic->next;

        free(p);

        ++i;

    }

    fclose(fp);

    return i;

}

void cardcpy(Card\* b,Card\* a)

{

    strcpy(b->aId,a->aId);

    strcpy(b->aPwd,a->aPwd);

    b->fBalance = a->fBalance;

    b->fTotalUse = a->fTotalUse;

    b->nDel = a->nDel;

    b->nStatus = a->nStatus;

    b->nUseCount = a->nUseCount;

    b->tEnd = a->tEnd;

    b->tLast = a->tLast;

    b->tStart = a->tStart;

}

int vague\_search(Cards \*c,Cards\* ret,char\* txt,int txt\_size)

{

    Card\* p=c->head;

    int sign=1;

    ret->tail=NULL;

    ret->head=ret->tail;

    while(p!=c->tail->next)

    {

        int idx=index\_KMP(txt,p->aId,get\_size(p->aId),txt\_size);

        // printf("%d\n",idx);

        if(idx!=-1)

        {

            InsertCard1(ret,p);

            sign=0;

        }

        p=p->next;

    }

    if(sign) return 0;

    return 1;

}

void PrintCard(Cards \*c)

{

    Card\* p=c->head;

    while(p!=NULL)

    {

        printf("%s\t%s\t%d\t%.1f\n",p->aId,p->aPwd,p->nStatus,p->fBalance);

        p=p->next;

    }

}

void InsertCard(Cards\* c,Card\* newc)

{

    if(c->head)

    {

        c->tail->next=newc;

        c->tail=c->tail->next;

        c->tail->next=NULL;

    }

    else

    {

        c->head=c->tail=newc;

        c->head->next=NULL;

    }

}

void InsertCard1(Cards\* c,Card\* newc)

{

    if(c->head)

    {

        c->tail->next1=newc;

        c->tail=c->tail->next1;

        c->tail->next1=NULL;

    }

    else

    {

        c->head=c->tail=newc;

        c->head->next1=NULL;

    }

}

// billing

Billing\* CreateBilling(char id[40])

{

    Billing\* newB = (Billing\*)malloc(sizeof(struct Billing));

    strcpy(newB->aCardName,id);

    time(&(newB->tStart));

    time(&(newB->tEnd));

    newB->fAmount = 0;

    newB->nStatus = 0;

    newB->nDel = 0;

    return newB;

}

int initBill(Billings \*b)

{

    b->tail=NULL;

    b->head = b->tail;

    FILE \*fb;

    if((fb=fopen("billing\_file.txt","r"))==NULL)

    {

        return 0;

    }

    int i=0;

    char line[1024];

    char Start[40],End[40];

    while(fgets(line,sizeof(line),fb)!=NULL)

    {

        Billing\* ib = (Billing\*)malloc(sizeof(Billing));

        sscanf(line,"%[^##]##%[^##]##%[^##]##%f##%d##%d\n",ib->aCardName,Start,End,&ib->fAmount,&ib->nStatus,&ib->nDel);

        if(ib->nDel==1) continue;

        ib->tStart = (time\_t)string\_to\_time(Start,sizeof(Start)/sizeof(char));

        ib->tEnd = (time\_t)string\_to\_time(End,sizeof(End)/sizeof(char));

        InsertBilling(b,ib);

        ++i;

    }

    fclose(fb);

    return i;

}

int saveBill(Billings\* b)

{

    FILE \*fp;

    Billing\* ib = b->head;

    if((fp = fopen("billing\_file.txt","w")) == NULL)

    {

        printf("账单数据保存失败！\n");

        return 0;

    }

    if(ib==NULL)

    {

        return 0;

    }

    int i=0;

    while(ib!=NULL)

    {

        char sStart[40];

        char sEnd[40];

        time\_to\_string(ib->tStart,sStart,sizeof(sStart)/sizeof(char));

        time\_to\_string(ib->tEnd,sEnd,sizeof(sEnd)/sizeof(char));

        fprintf(fp,"%s##%s##%s##%.1f##%d##%d\n",ib->aCardName,sStart,sEnd,ib->fAmount,ib->nStatus,ib->nDel);

        Billing \*p = ib;

        ib=ib->next;

        free(p);

        ++i;

    }

    fclose(fp);

    return i;

}

Billing\* searchLoginBill(Billings\* b,char\* id)

{

    Billing\* p=b->head;

    while(p!=NULL)

    {

        if(strcmp(p->aCardName,id)==0 && p->nStatus==0) return p;

        p=p->next;

    }

    return NULL;

}

void InsertBilling(Billings\* b,Billing\* newb)

{

    if(b->head)

    {

        b->tail->next=newb;

        b->tail=b->tail->next;

        b->tail->next=NULL;

    }

    else

    {

        b->head=b->tail=newb;

        b->head->next = NULL;

    }

}

// charge

Charge\* CreateCharge(char id[40],int st,float money)

{

    Charge\* newM = (Charge\*)malloc(sizeof(struct Charge));

    strcpy(newM->aCardName,id);

    time(&(newM->tTime));

    newM->nStatus=st;

    newM->fMoney = money;

    newM->nDel=0;

}

int initCharge(Charges\* m)

{

    m->tail=NULL;

    m->head = m->tail;

    FILE \*fp;

    if((fp=fopen("record\_file.txt","r"))==NULL)

    {

        return 0;

    }

    int i=0;

    char line[1024];

    char sTime[40];

    while(fgets(line,sizeof(line),fp)!=NULL)

    {

        Charge\* im = (Charge\*)malloc(sizeof(Charge));

        sscanf(line,"%[^##]##%[^##]##%d##%f##%d\n",im->aCardName,sTime,&im->nStatus,&im->fMoney,&im->nDel);

        if(im->nDel==1) continue;

        im->tTime = (time\_t)string\_to\_time(sTime,sizeof(sTime)/sizeof(char));

        InsertCharge(m,im);

        ++i;

    }

    fclose(fp);

    return i;

}

int saveCharge(Charges \*m)

{

    FILE \*fp;

    Charge \*im = m->head;

    if((fp = fopen("record\_file.txt","w")) == NULL)

    {

        printf("充值退费数据保存失败！\n");

        return 0;

    }

    if(im==NULL)

    {

        return 0;

    }

    int i=0;

    while(im!=NULL)

    {

        char sTime[40];

        time\_to\_string(im->tTime,sTime,sizeof(sTime)/sizeof(char));

        fprintf(fp,"%s##%s##%d##%.1f##%d\n",im->aCardName,sTime,im->nStatus,im->fMoney,im->nDel);

        Charge \*p = im;

        im=im->next;

        free(p);

        ++i;

    }

    fclose(fp);

    return i;

}

void InsertCharge(Charges\* m,Charge\* newM)

{

    if(m->head)

    {

        m->tail->next=newM;

        m->tail=m->tail->next;

        m->tail->next=NULL;

    }

    else

    {

        m->head=m->tail=newM;

        m->head->next = NULL;

    }

}

// tool

void delString (char\* txt)

{

    for(int i=0;i<strlen(txt);++i) txt[i]='\0';

}

void printTime(time\_t tt)

{

    struct tm \* timeinfo;

    timeinfo = localtime(&tt);

    char strTime[40];

    strftime(strTime,sizeof(strTime),"%Y年%m月%d日 %H:%M",timeinfo);

    printf(strTime);

}

void time\_to\_string(time\_t tt,char\* strTime,int size)

{

    struct tm \* timeinfo;

    timeinfo = localtime(&tt);

    strftime(strTime,size,"%Y-%m-%d %H:%M",timeinfo);

}

long int string\_to\_time(char\* strTime,int size)

{

    // printf("%s\n",strTime);

    struct tm timeInfo = {0};

    int year,mon,mday,hour,min;

    sscanf(strTime,"%d-%d-%d %d:%d",&year,&mon,&mday,&hour,&min);

    // printf("year = %d, mon = %d, mday = %d, hour = %d, min = %d\n",year,mon,mday,hour,min);

    timeInfo.tm\_year = year - 1900;

    timeInfo.tm\_mon = mon - 1;

    timeInfo.tm\_mday = mday;

    timeInfo.tm\_hour = hour;

    timeInfo.tm\_min = min;

    return (long int)mktime(&timeInfo);

}

void initSystem(Cards\* c,Billings\* b,Charges\* m)

{

    int cNum = initCard(c);

    int bNum = initBill(b);

    int mNum = initCharge(m);

    printf("好像连上了，\n获得了%d条卡片数据\n",cNum);

    printf("获得了%d条收费数据\n",bNum);

    printf("获得了%d条充值退费数据\n",mNum);

    system("pause");

}

void save(Cards\* c,Billings\* b,Charges\* m)

{

    int cNum = saveCard(c);

    int bNum = saveBill(b);

    int mNum = saveCharge(m);

    printf("保存了%d条卡片数据\n",cNum);

    printf("保存了%d条收费数据\n",bNum);

    printf("保存了%d条充值退费数据\n",mNum);

}

void get\_next(char \*txt,int \*next,int txt\_size)

{

    int j=0,k=-1;

    next[j]=-1;

    while(j<txt\_size-1)

    {

        if(k==-1 ||txt[j]==txt[k])

        {

            ++j;++k;

            next[j]=k;

        }

        else k=next[k];

    }

}

int index\_KMP(char\* txt,char \*id,int id\_size,int txt\_size)

{

    int next[txt\_size-1];

    get\_next(txt,next,txt\_size);

    int i=0,j=0;

    while(i<id\_size-1 && j<txt\_size-1)

    {

        if(j==-1 || id[i]==txt[j]) ++i,++j;

        else j=next[j];

    }

    if(j>=txt\_size-1) return i-txt\_size+1;

    else return -1;

}

int get\_size(char\* txt)

{

    int i=0;

    while(txt[i]!='\0') ++i;

    return i+1;

}

void InputID(char\* id)

{

    while(1)

    {

        printf("请输入卡号<长度为1~18>:");

        scanf("%s",id);

        if(strlen(id)>MAXLEN-1)

        {

            printf("卡号过长，请重新输入!\n");

            delString(id);

        }

        else

            break;

    }

}

void InputPWD(char\* pwd)

{

    while(1)

    {

        printf("请输入密码<长度为1~18>:");

        scanf("%s",pwd);

        if(strlen(pwd)>MAXLEN-1)

        {

            printf("密码过长，请重新输入!\n");

            delString(pwd);

        }

        else

            break;

    }

}

double CalcuPrice(time\_t tStart,time\_t tEnd)

{

    double deltaTime = tEnd - tStart;

    int price = 100;

    return deltaTime \* price / 3600;

}

Card\* getCard(Cards\* c,char\* id)

{

    Card\* p=c->head;

    while(p!=NULL)

    {

        if(strcmp(p->aId,id)==0) return p;

        p=p->next;

    }

    return NULL;

}

tools.h

#pragma once

#include "service.h"

// card

int initCard(Cards\* c);

int saveCard(Cards\* c);

void cardcpy(Card\* b,Card\* a);

Card\* getCard(Cards\* c,char\* id);

int vague\_search(Cards \*c,Cards\* ret,char\* txt,int txt\_size);

void PrintCard(Cards \*c);

void InsertCard(Cards\* c,Card\* newc);

void InsertCard1(Cards\* c,Card\* newc);

// bill

Billing\* CreateBilling(char id[40]);

int initBill(Billings \*b);

int saveBill(Billings\* b);

Billing\* searchLoginBill(Billings\* b,char\* id);

void InsertBilling(Billings\* b,Billing\* newb);

// charge

Charge\* CreateCharge(char id[40],int st,float money);

int initCharge(Charges\* m);

void InsertCharge(Charges\* m,Charge\* newM);

int saveCharge(Charges \*m);

void InsertCharge(Charges\* m,Charge\* newM);

// tool

void delString (char\* txt);

void printTime(time\_t tt);

void time\_to\_string(time\_t tt,char\* strTime,int size);

long int string\_to\_time(char\* strTime,int size);

void initSystem(Cards\* c,Billings\* b,Charges\* m);

void save(Cards\* c,Billings\* b,Charges\* m);

void get\_next(char \*txt,int \*next,int txt\_size);

int index\_KMP(char\* txt,char \*id,int id\_size,int txt\_size);

int get\_size(char\* txt);

void InputID(char\* id);

void InputPWD(char\* pwd);

double CalcuPrice(time\_t tStart,time\_t tEnd);