**程序设计与实践**

**实验报告**

**实验四：自动寄存柜**

**实验时间：2022.03.30 ~ 2022.04.05**

**目录**

**实验名称………………………………………………………………P3**

**实验环境………………………………………………………………P3**

**实验目的………………………………………………………………P3**

**实验步骤………………………………………………………………P4**

**（一）菜单和n个寄存柜…..……………..………………………………P4**

**（二）寄存东西……………………………………………………………P4**

**（三）取出东西……………………………………………………………P7**

**（四）万能密码……………………………………………………………P8**

**（五）重置柜子数量………………………………………………………P9**

**（六）找回密码……………………………………………………………P10**

**实验总结………………………………………………………………P11**

**参考资料………………………………………………………………P11**

**一：****实验名称：自动寄存柜**

**二：实验环境: 一台安装有C语言编译器的电脑。**

**三：实验目的**

（1）培养和锻炼对具有一定复杂度和规模的问题的分析与求解能力。

（2）掌握程序设计的综合方法，能综合应用各种数据类型实现较复杂数据的存储。

（3）培养良好的C程序设计风格与代码规范意识。

**四：实验内容**

　　某超市门口的自动寄存柜有ｎ个寄存箱，并且有一个投币控制器，顾客想要寄存小件物品时，只要在投币控制器中投入一个１元硬币，如果此时有空闲的箱子，寄存柜就会自动打开一个空的箱子，并且打印输出一张小小的密码纸条；如果没有空闲的箱子，则提示“本柜已满”。当顾客离开超市时，用密码纸条上指定的数字密码依次输入到开箱控制器，则顾客所存包的箱子门就会自动打开，顾客取走物品后，关上门。

　　输入数据时，可先输入寄存箱总数n，再由用户选择是“投硬币”还是“输密码”。

　　如果选择“投硬币”，则只有硬币值是１时才开箱。如果有空闲的箱子，则输出箱子编号及密码（4位数字）；如果无空闲的箱子，则提示“本柜已满”。

如果选择“输密码”，若输入的密码与某一箱子密码相符，则显示打开的箱子编号，否则输出提示“密码错误”.万能密码：假定有一个密码000086，该密码用于营业结束后整理寄存柜。先在菜单中输入隐藏的选项5，然后提示请输入万能密码，密码输入后，打开所有的箱子，如有箱子中客户没有取走东西，则提示：请重点检查某编号的箱子是否有物留存。

万能密码：假定有一个密码000086，该密码用于营业结束后整理寄存柜。先在菜单中输入隐藏的选项5，然后提示请输入万能密码，密码输入后，打开所有的箱子，如有箱子中客户没有取走东西，则提示：请重点检查某编号的箱子是否有物留存。

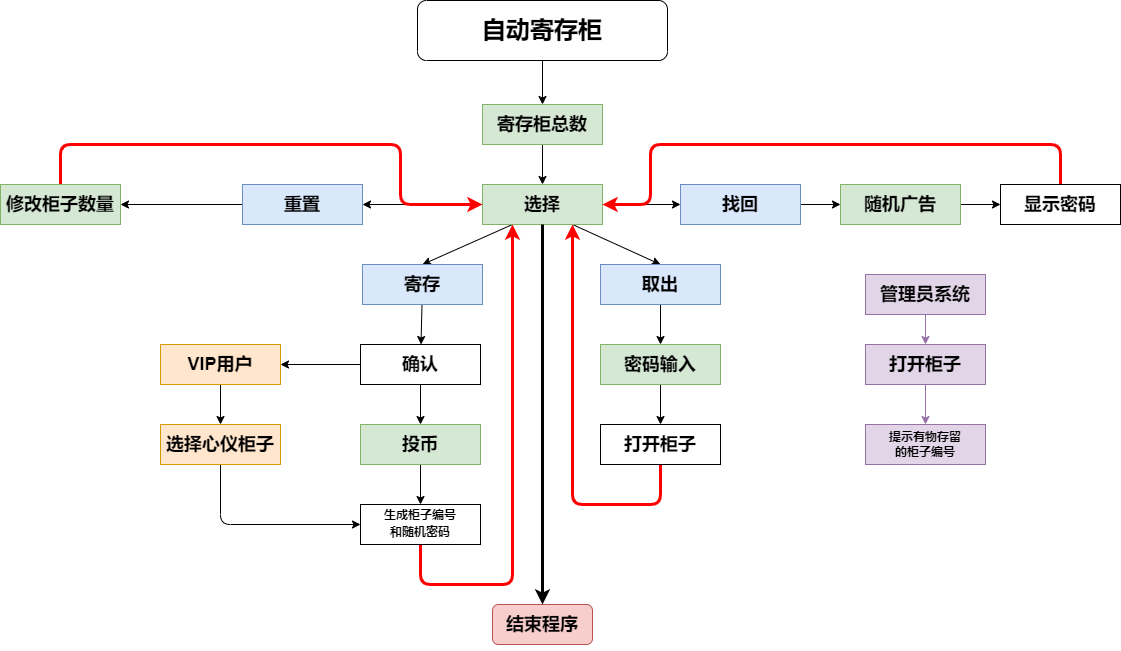


图1 本自动寄存柜正常使用的简略流程图

**五：实验步骤**

**本次实验的基本要求是完成自动寄存柜的寄存、取出和万能密码等三大功能。在此基础上增加重置柜子的数量、找回密码、VIP寄存服务等功能。**

**注意：本程序在正常应该数字输入的地方，均对程序有一定的保护措施，如果输入与数字无关的非法字符，会直接退出本程序。**

1. 菜单和n个寄存柜

菜单页面如图2.1和图2.2所示，在本页面中寄存柜会显示为【空】和【满】，未使用的会标注【未启用】。当n个柜子数大于15个的时候会有提示。其中代码部分请参考文件夹中P4.c文件的menu()函数。

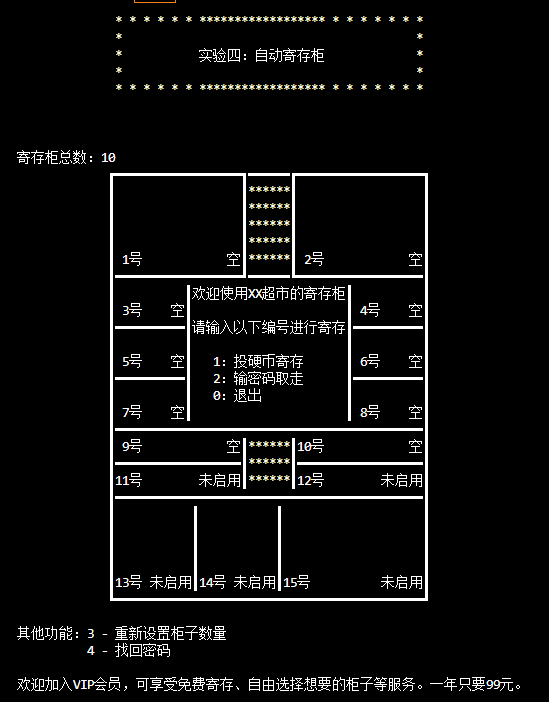
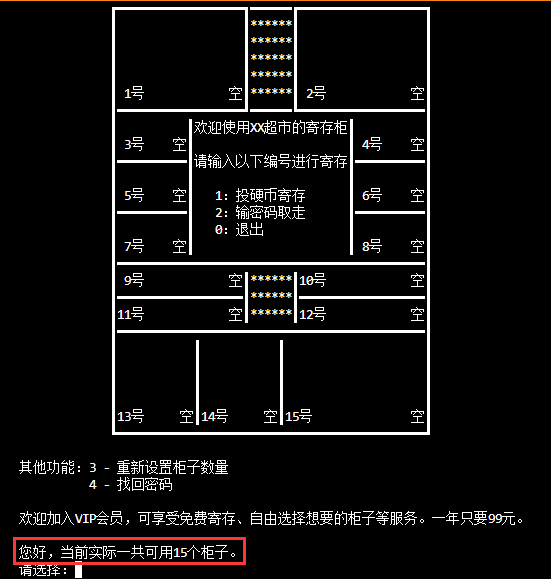
 

图2.1 自动寄存柜的菜单页面 图2.2 自动寄存柜总数大于等于15时的提示

二、寄存东西

输入选择‘1’，进入寄存功能。首先输入任意字符作为寄存的确认，再进行投币（仅限1元硬币），最后打印**按照寄存顺序**生成柜子的编号和它的随机密码。

其中在确认过程中输入VIP用户的密码（VIP2022或vip2022），可以无需投币和自由选择自己需要的柜子编号进行寄存。

为了保证寄存的安全性或防止密码溢出等情况，使用随机数生成密码。

①普通用户的正常寄存与寄存后柜子的页面

程序执行截图：

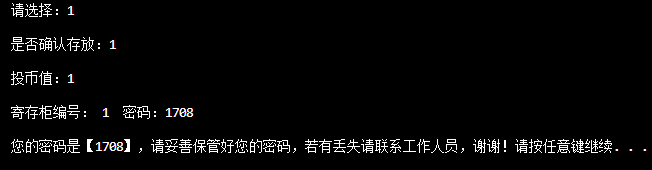
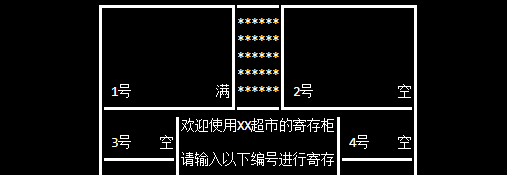
 

图3.1 自动寄存柜的正常寄存

核心代码：

srand(time(NULL)); //埋下时间种子

d=rand()%(9999-1000+1)+1000; /\*保证密码当前的唯一性，使用d生成随机4位密码\*/

for(i=0;i<=much;i++){ //确认柜子情况

if(t[i]==2) F++;

}

if(F==much){

printf("\n非常抱歉，当前柜子已满。\n\n");

break;

}

printf("\n投币值：");

scanf("%d",&coin);

if(coin!=1){ //只能投1元硬币判断

printf("\n投币失败，仅支持1元硬币。\n");

}else{

for(i=1;i<=much;i++){ /\*东西寄存\*/

if(t[i]==1){ //for数组循环，找到第一个还是空的柜子

Sleep(1000); //当输入操作小于1秒会导致随机数失效，放入Sleep函数暂停1秒

printf("\n寄存柜编号：%2d 密码：%d\n",i,d); //告知用户当前寄存情况

a[i]=d; //记录密码

t[i]=2; //内部状态设置为2-满

if(i<=15) memset(state[i], '\0', sizeof(state[i])); //先申请新的内存

if(i<=15) strcpy(state[i],"满"); //再更改外部状态

printf("\n您的密码是【%d】，请妥善保管好您的密码，若有丢失请联系工作人员，谢谢！",a[i]); //再次提示用户

break; //跳出循环

}else{

continue; //非空继续循环，防止被覆盖

}

}

break;

②投币并非为1元

程序执行截图：

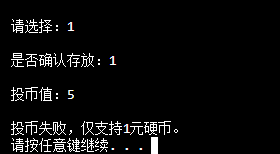


图3.2 寄存投币值非法

③柜子存满

程序执行截图：

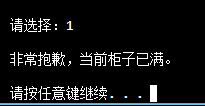


图3.3 寄存柜已满

**④VIP用户寄存**

程序执行截图：

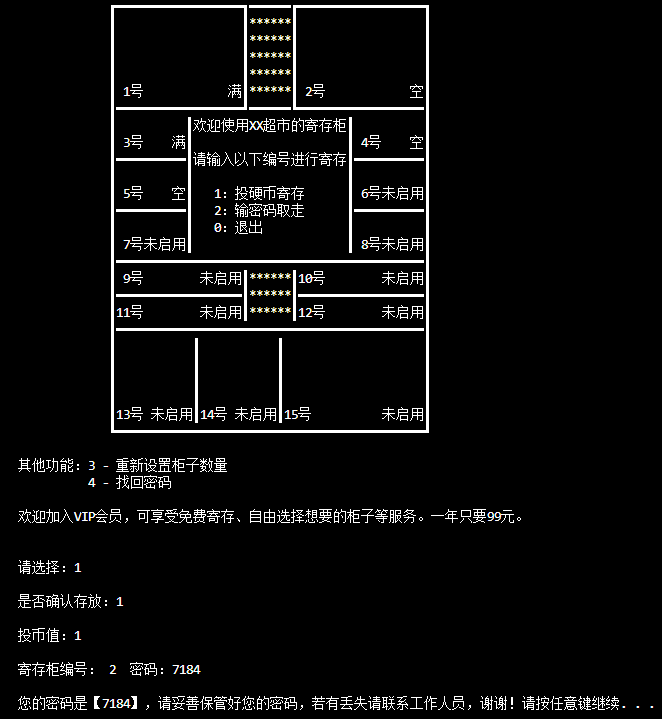
 

图3.3 VIP用户自由选择编号寄存 图3.4 VIP用户不影响正常寄存图①（2号柜子）

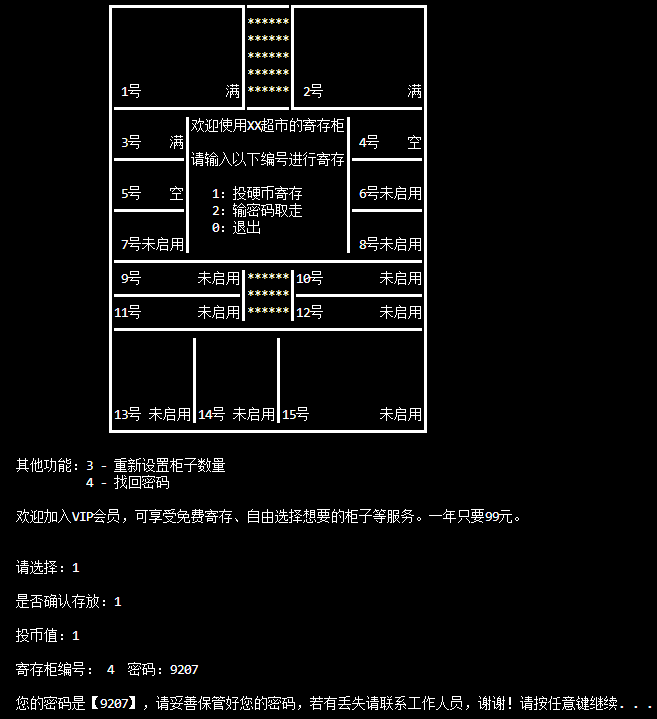
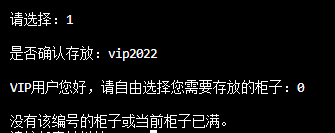
 

图3.5 VIP用户不影响正常寄存图②（2号柜子→4号柜子） 图3.6 VIP用户非法寄存

核心代码：

printf("\n是否确认存放：");

getchar(); //取回车符

gets(vip); /\*VIP用户寄存，无需硬币\*/

if(strcmp(vip,"VIP2022")==0||strcmp(vip,"vip2022")==0){ //确认密码

printf("\nVIP用户您好，请自由选择您需要存放的柜子：");

scanf("%d",&box);

if(box<=0||box>much||t[box]==2){ //判断柜子非法条件

printf("\n没有该编号的柜子或当前柜子已满。\n");

system("pause");

system("CLS");

break;

}

else{ //打印寄存情况

Sleep(1000);

printf("\n寄存柜编号：%2d 密码：%d\n",box,d);

a[box]=d;

t[box]=2;

if(box<=15) memset(state[box], '\0', sizeof(state[box]));

if(box<=15) strcpy(state[box],"满");

printf("\n您的密码是【%d】，请妥善保管好您的密码，若有丢失请联系工作人员，谢谢！",a[box]);

break;

}

}

三、取出东西

输入选择‘2’，进入取出功能。取出东西需要输入对应柜子的正确密码，否则无法打开柜子。如果柜子全部本身为空，则提醒用户全为空。

程序执行截图：

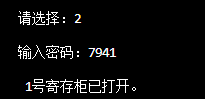
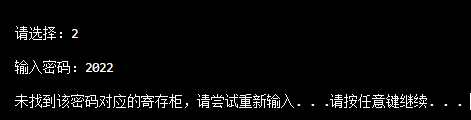
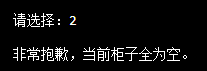
  

图4.1 正常打开柜子 图4.2 密码错误 图4.3 柜子为空

核心代码：

int L=0,flag\_p=0,password; //L作为统计柜子空的数量，用于判断是否全为空

for(i=1;i<=much;i++){

if(t[i]==1) L++;

}

if(L==much){

printf("\n非常抱歉，当前柜子全为空。\n\n");

break;

}else{

printf("\n输入密码：");

scanf("%d",&password);

}

for(i=1;i<=much;i++){ /\*取东西\*/

if(password==a[i]){ //判断输入密码和密码数组中的元素是否相等

flag\_p=0; //flag\_p用于判断有没有找到密码，每次需要初始化为0

printf("\n%2d号寄存柜已打开。\n\n",i);

t[i]=1; //内部状态设置为1-空

a[i]=0; //重新将密码设为0（无）

if(i<=15) memset(state[i], '\0', sizeof(state[i])); //先申请新的内存

if(i<=15) strcpy(state[i],"空"); //再更改外部状态

system("pause");

system("CLS");

break;

}else{

flag\_p++; //没有找到flag\_p次数增加

}

}

if(flag\_p==much){

printf("\n未找到该密码对应的寄存柜，请尝试重新输入. . .");

system("pause");

system("CLS");

}

flag\_p=0; //重新初始化flag\_p为0

break;

四、万能密码

输入隐藏选择‘5’，进入管理员功能。先输入管理员密码‘000086’，然后自动打开所有柜子，并提醒管理员，柜子内仍有物品需要检查的柜子编号。

程序执行截图：

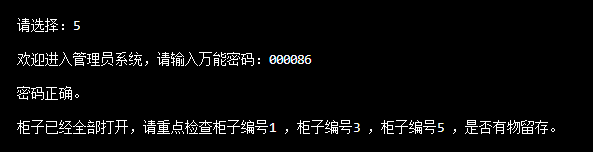
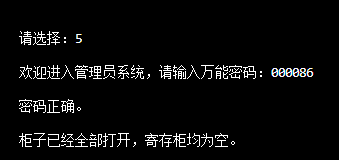
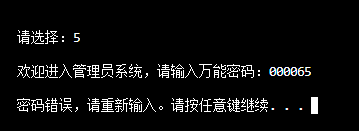
  

图5.1 正常打开柜子 图5.2 柜子为空 图5.3 密码错误

核心代码：

char admin[10]; //管理员密码

int b[1000]={0},flaga=0,O=0; //数组b用于登记非法柜子，flaga用于记录是否均为空，O是为了逗号的输出

printf("\n欢迎进入管理员系统，请输入万能密码：");

scanf("%s",admin);

if(strcmp(admin,"000086")==0){

printf("\n密码正确。\n");

printf("\n柜子已经全部打开，");

for(i=1;i<=much;i++){

if(t[i]==2){ //查找2-满的柜子

b[i]++; //数组b记录

flaga=0;

t[i]=1; //打开柜子

}else{

flaga++;

}

}

if(flaga==much){

printf("寄存柜均为空。\n\n");

}else{

printf("请重点检查");

for(i=1;i<=much;i++){

if(b[i]==1){

printf("柜子编号%-2d",i);

O=1;

}

if(O==1) printf("，");

O=0;

}

printf("是否有物留存。\n\n");

}

}else{

printf("\n密码错误，请重新输入。");

system("pause");

system("CLS");

break;

}

break;

五、重置柜子数量

由于可能柜子不够用，需要扩建；或者柜子有故障，减少能够使用的数量，因此开发了此功能。

程序执行截图（其中寄存和取出功能经测试能够正常使用，不再列出）：

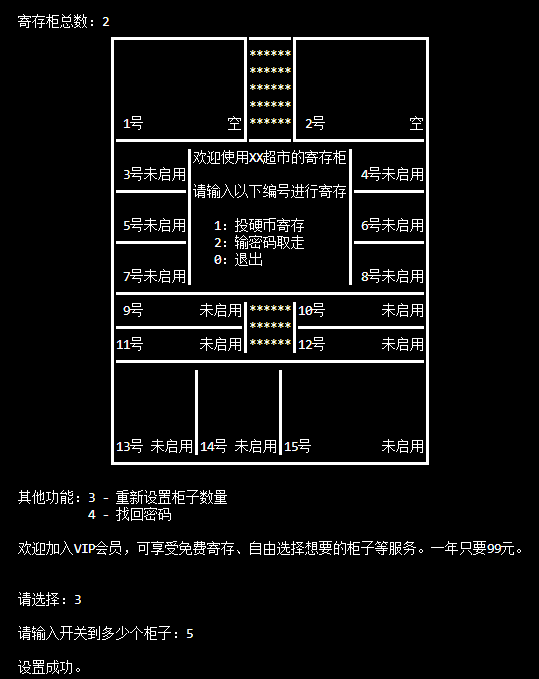
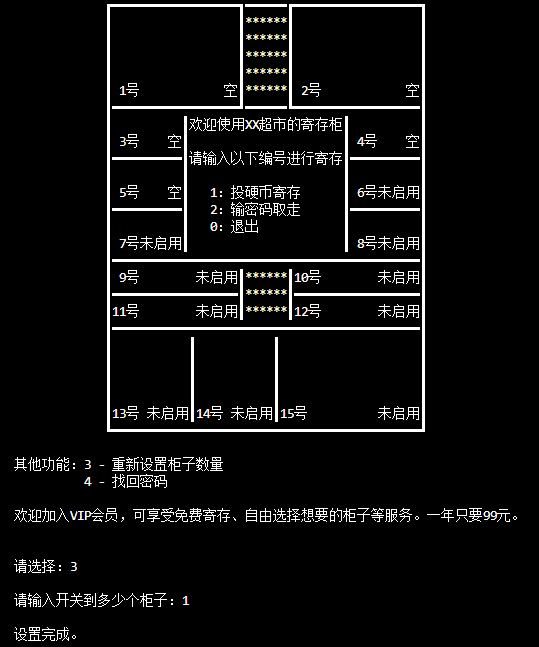
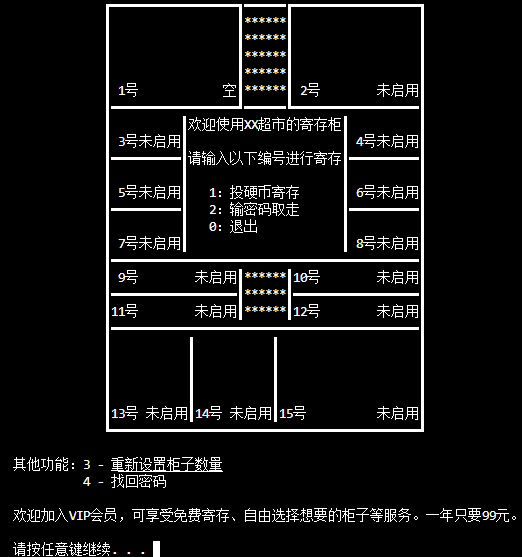
  

图6.1 2个柜子扩展到5个 图6.2 5个柜子缩小到1个柜子 图6.3 设置到1个柜子完成后

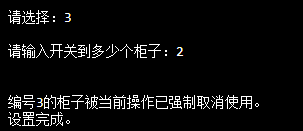


图6.4 如果有一个柜子内有东西的提示

核心代码：

int window; //重新输入柜子的数量

printf("\n请输入开关到多少个柜子：");

scanf("%d",&window);

if(window==much){ //没有更改直接跳过设置

printf("\n设置完成。");

system("pause");

system("CLS");

break;

}else{

if(window>=much){

for(i=much+1;i<window+1;i++){ /\*much->window设置\*/

strcpy(state[i],"空"); //更新外部状态

if(i==15) break;

}

for(i=much+1;i<window+1;i++){

t[i]=1; //更新内部状态

}

much=window; //更新much+1的值，不然会覆盖一个柜子

printf("\n设置成功。");

break;

}else{

for(i=window+1;i<much+1;i++){ /\*window->much设置\*/

memset(state[i], '\0', sizeof(state[i])); //先申请新的内存

strcpy(state[i],"未启用"); //然后更改外部状态

if(i==15) break;

}

for(i=window+1;i<much+1;i++){

if(t[i]==2) printf("\n\n编号%d的柜子被当前操作已强制取消使用。",i); //提示用户部分柜子被拆除

t[i]=0; //更新内部状态

}

much=window; //更新much-1的值，不然会覆盖一个柜子

printf("\n设置完成。");

break;

}

}

break;

六、找回密码

在使用中，可能会忘记密码。包括我在测试中，没办法一口气记下很多密码，导致无法正常测试性能。所有增加这个功能，但是使用中会有一定的阻碍——15秒内的小广告+图片。**其中打印广告的时候，是一个字一个字输出的。本报告无法完全展示其效果，欢迎打开程序使用试试。**

程序执行截图：

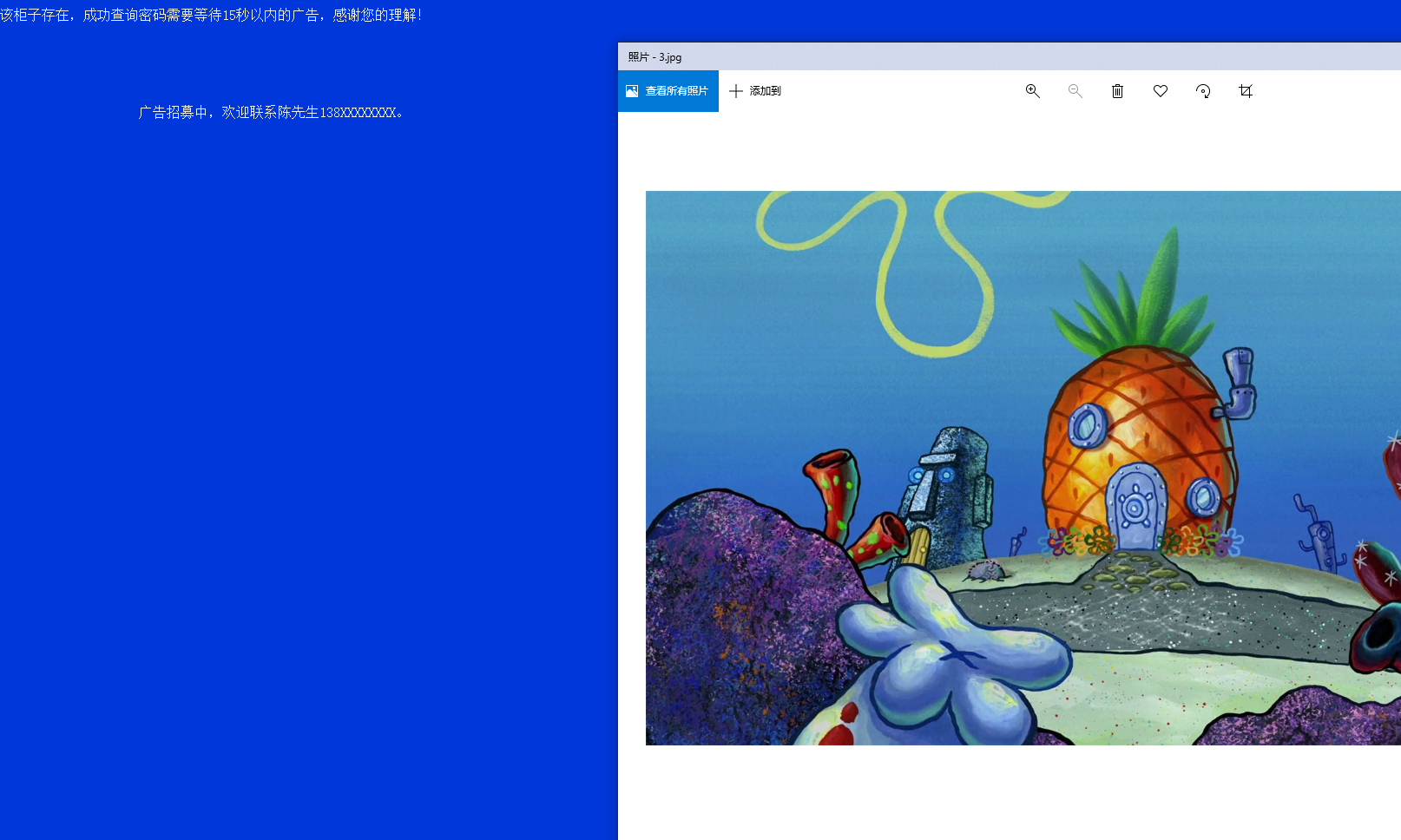
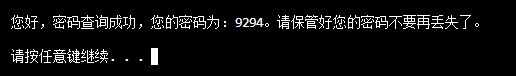
 

图7 找回密码

核心代码：

int U,adv=1; //U是输入柜子编号，adv是广告编号

adv=rand()%(6-1+1)+1; //广告一共6个，取1-6的随机数

printf("\n查询密码，请输入柜子编号：");

scanf("%d",&U);

if(U==0||U>much){ //柜子编号不存在

printf("\n无该编号的柜子\n\n");

break;

}else if(a[U]==0){ //柜子编号本身为空

printf("\n您查询的柜子为空。\n\n");

break;

}else{ //忘记密码会有"小惩罚"（看广告）

system("CLS");

char adv\_1[200]={" 把心意藏在瓶子里，每天百事可乐！"};

printf("\n该柜子存在，成功查询密码需要等待15秒以内的广告，感谢您的理解！\n\n");

if(adv==1){

system("color 09");

printf("\n\n\n\n\n");//百事可乐

for(i=0;adv\_1[i]!='\0';i++){

printf("%c",adv\_1[i]);

Sleep(100);

}

printf("\n\n");

system("img\\1.jpg"); //打开图片

Sleep(8000);

}

system("color 07"); //恢复正常黑白颜色界面

system("CLS");

printf("\n您好，密码查询成功，您的密码为：%4d。请保管好您的密码不要再丢失了。\n\n",a[U]);

system("pause");

system("CLS");

break;

}

**六：实验小结**

完成情况： 完成实验要求，并新增重置柜子数量、找回密码、VIP用户寄存等功能。

需改进处： 可以尝试将柜子和密码用结构体串联起来，构成结构体数组，更方便查看对应数据。

收获： 在很多地方进行了挑战和学习，这次使用system函数打开图片、更改程序页面颜色、windows.h文件中的Sleep函数等做了广告，还有对随机数更具有随机性的方案进行了尝试。

体会： 一个寄存柜、一个程序可以通过代码实现、穿插千变万化的功能，让实验作业不再枯燥、乏味。

**七：参考资料**：C语言程序设计（第4版），何钦铭，颜晖主编。