

# Gate of Babylon

# 网店管理系统

用户手册

版本 1.0.1

发行单位: Cardinal

发行时间：2021.11.8

とある  
管理系統の  
Gate of Babylon

## 1. 引言

### 1.1 编写目的

介绍本产品的相关参数，配置要求，使用方法，便于为用户提供更优质的服务。

### 1.2 背景

该软件是网络电商信息管理系统，能够进行用户与商店交互，管理员与商店的交互，用户与管理员的交互。

## 2. 用途

### 2.1 功能

功能一：用户登录，用户用已有账号登录，登陆后有 11 个功能可供选择（注销登录、查看商品、商品搜索、添加商品至购物车、删除购物车中的商品、查看购物车、结账、浏览记录、查看订单、留言反馈、查看消息）。

功能二：用户注册，输入用户名和密码后即可创建新账号。

功能三：管理员登录，管理员登录已有账号，登陆后有 8 个功能可供选择（注销登录、查询商品、增加商品、删除商品、修改商品数量和价格、售货清单、已售清单、查看留言）。

功能四：管理员注册，输入管理员密匙后即可注册新账号。

### 2.2 性能

#### 2.2.1 精度

本产品商品价格只能有两位小数并且不能超过 99999 元。商品名称和品牌不能超过六个汉字，最好不要加入带有英文名称的商品，这会导致搜索结果的不精确。

#### 2.2.2 时间特性

本产品在与用户交互时设计了 3 至 5 秒的响应延迟，属于正常现象，请放心使用。

## 3. 运行环境

Windows 7 操作系统及以上

处理器：Intel Core i3-4340 及以上

输出环境：基于控制台的输出

硬盘所需空间：2 MB

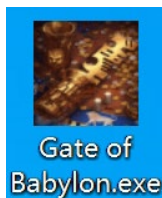
附注事项：按照程序引导进行输入，若输入不合规范可能造成无法挽回的后果。

## 4. 使用过程

请仔细阅读本节内容以获得更好的体验。

### 4.1 安装与运行

软件已打包，下载即可运行，双击“Gate of Babylon”图标运行。如果打开程序遇到问题，请联系 [201220192@smail.nju.edu.com](mailto:201220192@smail.nju.edu.com) 获取帮助。



（双击打开）

### 4.2 输入格式

功能选择界面：



出现该界面时，按照提示输入 1 至 5 之一以选择一项功能。

- 1： 用户登录。
- 2： 用户注册。
- 3： 管理员登录。
- 4： 管理员注册。
- 5： 退出程序。

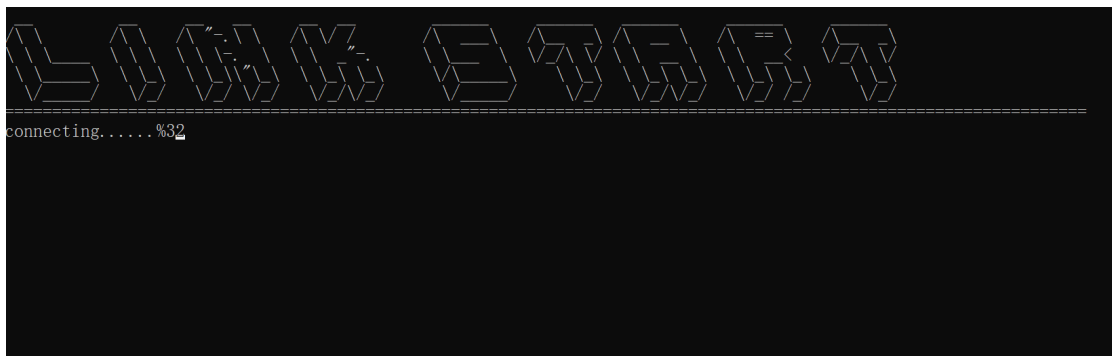
如果你是用户，那么请选择 1 或 2

4.2.1 用户登录界面：



依次输入用户名和密码，其中密码会被隐藏并用\*代替。如果账号密码错误将无法成功登录！这里给出一个用于测试的账号和密码：user1 passworduser1

输入正确的密码后将会启动连接：



出现该界面时，请耐心等待至连线完成。

连接成功后，出现用户功能选择界面：



你可以在 1 至 11 中进行选择！

#### 4.2.2 用户注册界面：



输入你想要的用户名，再连续两次输入密码即可完成注册。但是不能注册已经存在的用户名。

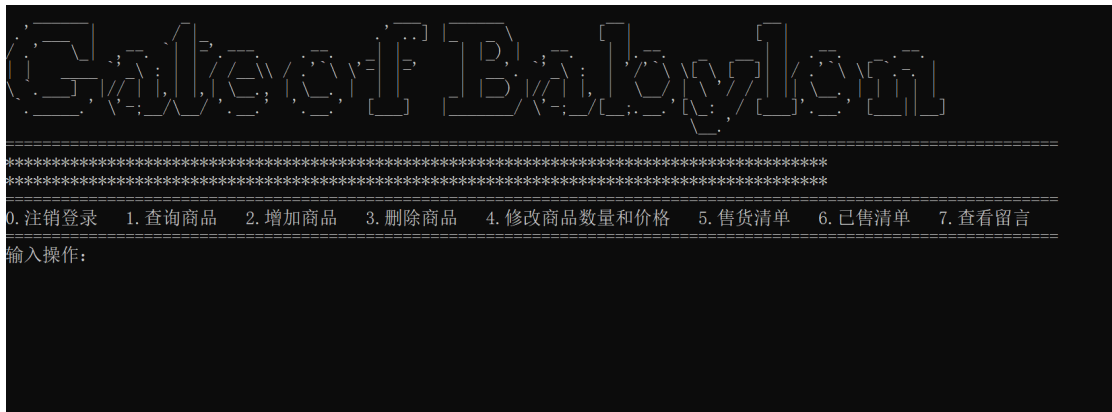


若不想注册，输入“back”或“quit”即可离开。

#### 4.2.3 管理员登录界面：



依次输入管理员 ID 和密码，其中密码会被隐藏并用\*代替。如果账号密码错误将无法成功登录！这里给出一个用于测试的账号和密码：admin1 12345  
登录成功后，出现管理员功能选择界面：



你可以在 0 至 7 中进行选择！

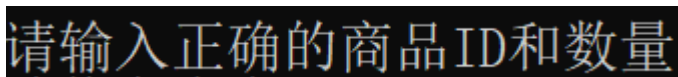
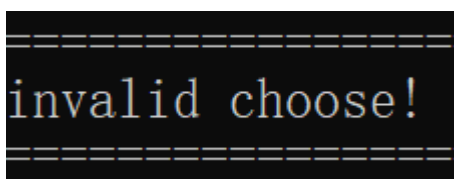
#### 4.2.4 管理员注册

注册过程同用户注册，但需要优先输入管理员密匙，若需要管理员密匙，请联系 [201220192@smail.nju.edu.com](mailto:201220192@smail.nju.edu.com) 获取帮助。

#### 4.3 输出

输出内容包括：商品信息，功能提示，操作询问，错误信息，警告提示，软件信息等。

以下展示部分报错信息：



出现这些信息这意味着你的输入不合法。

ID	名称	品牌	价格	数量
F0000	芒果干	三只松鼠	12	100
F0001	酸奶	光明	51.5	10
F0002	酸奶	蒙牛	59.4	6
F0003	酸奶	伊利	56	15
F0004	可乐	可口可乐	10	30
F0005	可乐	百事可乐	12	20
F0006	华夫饼	华美	29.9	9
F0007	每日坚果	沃隆	100	16
F0008	Enuma_Elish	Babylon	9999	1

```

*****
请输入查询内容：可乐
*****
ID          名称          品牌          价格          数量
F0004       可乐          可口可乐      10            30
F0005       可乐          百事可乐      12            20
*****

```

```

*****
ID          名称          品牌          价格          数量          操作
F0008      Enuma_Elish  Babylon      9999          1             查询商品
F0004      可乐          可口可乐      10            30            查询商品
F0005      可乐          百事可乐      12            20            查询商品
*****
=====
1. 退出    2. 清空历史记录
=====

```

[illegible]

4.4 部分特殊功能介绍

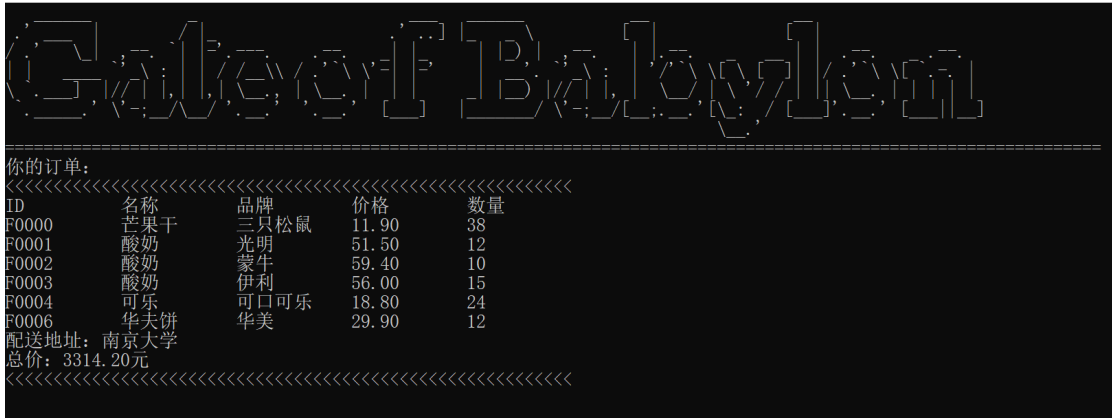
商品查询功能：

```
*****
ID      名称      品牌      价格      数量
F0000   芒果干      三只松鼠  12.00     100
F0001   酸奶          光明      51.50     10
F0002   酸奶          蒙牛      59.40      6
F0003   酸奶          伊利      56.00     15
F0004   可乐          可口可乐  10.00     30
F0005   可乐          百事可乐  12.00     20
F0006   华夫饼        华美      29.90      9
F0007   每日坚果      沃隆      100.00    16
F0008   Enuma_Elish Babylon  9999.00    1
*****
请输入查询内容：
```

出现该界面后输入你要查询的内容，你可以输入一整行，按回车进行搜索。你可以输入ID，名称，品牌，或是它们的组合，系统会将匹配度更高的内容进行输出，如果存在与你输入的内容相同的商品，程序将会立即将它们输出，若没有，将会进行模糊匹配，但这只针对中文，对于英文的输入，必须找到完全相同的商品才会输出。

```
*****
ID      名称      品牌      价格      数量
F0000   芒果干      三只松鼠  12.00     100
F0001   酸奶          光明      51.50     10
F0002   酸奶          蒙牛      59.40      6
F0003   酸奶          伊利      56.00     15
F0004   可乐          可口可乐  10.00     30
F0005   可乐          百事可乐  12.00     20
F0006   华夫饼        华美      29.90      9
F0007   每日坚果      沃隆      100.00    16
F0008   Enuma_Elish Babylon  9999.00    1
*****
请输入查询内容： 可乐
*****
ID      名称      品牌      价格      数量
F0004   可乐          可口可乐  10.00     30
F0005   可乐          百事可乐  12.00     20
*****
```

账单查看功能：



进入账单查看界面后只能通过上下键进行操作，按 ESC 键退出。

留言反馈功能：

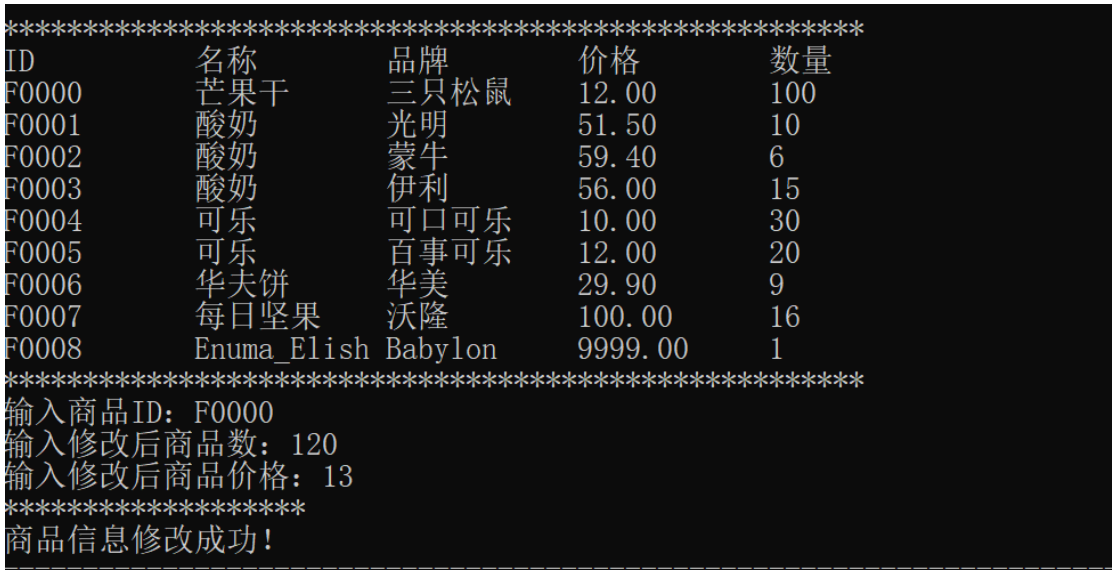
用户的反馈将直接发送至管理员进行处理。

消息查看功能：

用户可以查看管理员（商家）发来的信息，选择回复或忽略。

修改商品数量和价格功能：

依次输入商品 ID，数量，价格完成修改，若商品价格发生变动将会提醒将该商品放入购物车的客户。



管理员留言查看功能：



对于未处理的留言会被标记为红色，管理员可以选择对留言进行处理。

此时需要输入用户编号（大于 0 的整数!）



```

*****
user1的留言: 我要的货咋个还没到? 已处理
user1的留言: 收到! 已处理
user3的留言: 国科大在北京, 别送错了! 已处理
user2的留言: 要送货上门哦! 已处理
*****
1. 回复留言  2. 离开
1
请输入要处理用户的编号: 1

```

```

=====
Gilgamesh
=====
你要联系的客户是user1
ta的留言是:
*****
请输入你的回复:

```

然后输入回复（不能出现空格，回车结束）即可。

## 5.底层代码与实现

### 5.1 main() 函数

打印程序欢迎界面，选择界面，循环获取用户输入以决定各类函数调用直到用户决定退出。只进行用户登录、注册，管理员登录、注册的选择，打印结束信息。

### 5.2 商店类（Avalon.h）

```

class Treasure//商品类
{
public:
string id;//编号
string name;//商品名
string brand;//品牌
double price;//价格
int num;//数量
};
class Avalon
{
friend class Gilgamesh;
public:
Treasure treasure[100];//商品
int user_num;//用户数量
int admin_num;//管理员数量
int treasure_num;//商品种类数
void user_query(int user_idnum);//商品查询
void user_order(int user_idnum);//用户结账
void user_add(int user_idnum);//用户加车
void user_delete(int user_idnum);//用户退货
void user_display();//展示商品
void user_cart_display(int user_idnum);
//展示用户购物车 Avalon();
void renew_Avalon();//更新商店

```

```

void user_history_display(int user_idnum);
void user_order_display(int user_idnum);
void admin_to_user_display(int user_idnum);
void user_to_admin(int user_idnum);
void clear_userboard(int user_idnum);
private:
void init_Avalon();//初始化商店
void treasure_add();//管理员加商品
void treasure_delete();//管理员删除商品
void sold_display();//展示已售清单
void change_treasure_numorprice();//改变商品数量或价格
void note_display(int x);//留言展示
void note_handle();//留言处理
bool check_cart_and_id(int user_idnum,string fixid);//判断某商品是否在购物车中
};

```

重要函数说明:

成员函数中传入的参数 user\_idnum 为用户编号，以便直接确定函数所需访问的文件名

```
void init_Avalon();
```

在实例化时初始化商店，即将文件中保存的店铺信息传入。

```
Treasure treasure[100];//商品
```

```
int user_num;//用户数量
```

```
int admin_num;//管理员数量
```

```
int treasure_num;//商品种类数
```

```
void renew_Avalon();//更新商店信息。
```

由于用户与管理员对商品的增减操作直接改变类中成员:

```
Treasure treasure[100];//商品
```

```
int user_num;//用户数量
```

```
int admin_num;//管理员数量
```

```
int treasure_num;//商品种类数
```

所以该函数将改变后的信息写入文件

### 5.3 管理员类 (Gilgamesh.h)

```
class Gilgamesh//管理员类
```

```
{
```

```
public:
```

```
int idnum;//管理员编号
```

```
string id;//管理员 ID
```

```
string password;//管理员密码
```

```
Gilgamesh();
```

```
Gilgamesh(int x);
```

```
Gilgamesh(int x, string name, string pass); //构造函数
```

```
void create_admin_file(int x,string name,string pass);//新建管理员信息文件
```

```
void enter_Avalon(); //进入商店
```

```
};
```

重要函数说明：

**void create\_admin\_file(int x,string name,string pass);**

新建管理员信息文件，将管理员编号，ID，密码保存至文件中。

**void enter\_Avalon();**

访问店铺，可以根据管理员发出的命令调用店铺中的相关函数进行操作。

#### 5.4 用户类 (User.h)

```
class User//用户类
```

```
{
```

```
public:
```

```
int idnum;//用户编号
```

```
string id;//用户 ID
```

```
string password;//用户密码
```

```
User();
```

```
User(int x, string name, string pass);//构造函数
```

```
void visit_Avalon();//访问店铺
```

```
void create_user_file(int x, string name, string pass); //构造用户信息文件
```

```
void create_user_cart_file(int x);//新建购物车
```

```
void create_user_history_file(int x);//新建历史记录文件
```

```
void create_user_order_file(int x);//新建订单文件
```

```
void create_admin_to_user_file(int x);//新建信息反馈文件
```

```
};
```

重要函数说明：

**void create\_user\_file(int x, string name, string pass);**

在用户信息文件（用户.txt），新加入一行用户名，密码。

**void visit\_Avalon();**

访问店铺，可以根据用户发出的命令调用店铺中的相关函数进行操作。

#### 5.5 函数调用过程

用户操作(userop.h)

**void user\_sign\_up();//注册**

检测用户名密码是否符合要求

调用：

**create\_user\_file(x, name, pass);**

**create\_user\_cart\_file(x);**

**create\_user\_history\_file(x);**

**void user\_sign\_in();//登录**

将输入的用户名与密码和用户文件中保存的信息匹配。

匹配成功后调用 **visit\_Avalon();**

管理员操作(adminop.h)

**void admin\_sign\_up();//注册**

检测用户名密码是否符合要求

调用：

**create\_admin\_file(x,name, pass1);**

**void admin\_sign\_in();//登录**

将输入的用户名与密码和用户文件中保存的信息匹配。

匹配成功后调用 enter\_Avalon();

Main 函数打印启动信息后，根据功能选择调用：

user\_sign\_in();

void user\_sign\_up();

admin\_sign\_in();

admin\_sign\_up();

进入程序后，选择用户注册/登录。管理员登陆/注册，调用对应函数。

登陆成功后调用 visit\_Avalon/enter\_Avalon 进入店铺选择功能，调用店铺类 (Avalon) 中的各种函数修改类中成员的属性，最后调用 renew\_Avalon 将改变后的值写入文件保存。

5.6 部分代码展示

```
10
11 void Avalon::init_Avalon()
12 {
13     ifstream in_file(".\\商店\\店铺信息.txt", ios::in);
14     if (!in_file)
15         exit(-1);
16     in_file >> user_num >> admin_num >> treasure_num;
17     in_file.close();
18     ifstream in(".\\商店\\库存.txt", ios::in);
19     string head;
20     getline(in, head);
21     int i = 0;
22     while (!in.eof())
23     {
24         in >> treasure[i].id >> treasure[i].name >> treasure[i].brand >> treasure[i].price >> treasure[i].num;
25         i++;
26     }
27     in.close();
28     if (treasure_num != i-1)
29         cerr << "something wrong!" << endl;
30 }
31
32 void Avalon::renew_Avalon() { ... }
33
34 void Avalon::user_add(int user_idnum) { ... }
35
36 void Avalon::user_delete(int user_idnum) { ... }
37
38 void Avalon::user_display(int user_idnum)
39 {
40     cout << "*****" << endl;
41     cout << left << setw(12) << "ID" << setw(12) << "名称" << setw(12) << "品牌" << setw(12) << "价格" << endl;
42     for (int i = 0; i < treasure_num; i++)
43     {
44         if (treasure[i].num >= 0)
45             cout << left << setw(12) << treasure[i].id << setw(12) << treasure[i].name << setw(12) << treasure[i].brand << setw(12) << treasure[i].price << endl;
46     }
47     cout << "*****" << endl;
48 }
```

```

16 class Avalon
17 {
18     friend class Gilgamesh;
19 public:
20     Treasure treasure[100];
21     int user_num;
22     int admin_num;
23     int treasure_num;
24     void user_query(int user_idnum);
25     void user_order(int user_idnum);
26     void user_add(int user_idnum);
27     void user_delete(int user_idnum);
28     void user_display(int user_idnum);
29     void user_cart_display(int user_idnum);
30     void renew_usercart(int user_idnum);
31     Avalon()
32     {
33         init_Avalon();
34         for(int i = 1; i <= user_num; i++)
35             renew_usercart(i);
36     }
37     void renew_Avalon();
38     void user_history_display(int user_idnum);
39     void user_order_display(int user_idnum);
40     void admin_to_user_display(int user_idnum);
41     void user_to_admin(int user_idnum);
42     void clear_userboard(int user_idnum);
43 private:
44     void init_Avalon();
45     void treasure_add();
46     void admin_display();
47     void treasure_delete();
48     void change_treasure_numorprice();
49     void sold_display();
50     void note_display(int x);
51     void note_handle();
52     bool check_cart_and_id(int user_idnum, string fixid);
53 };
54

```

```

8
9 void User::visit_Avalon()
10 {
11     HANDLE handle = GetStdHandle(STD_OUTPUT_HANDLE);
12     WORD word = FOREGROUND_RED;
13     WORD word2 = FOREGROUND_RED | FOREGROUND_GREEN | FOREGROUND_BLUE;
14     Avalon gate;
15     system("cls");
16     welcome_info();
17     gate.admin_to_user_display(idnum);
18     user_func_choose();
19     cout << "输入操作: " << endl;
20     string choose;
21     cin >> choose;
22     while (choose != "1") { ... }
111 }
112
113 void User::create_user_file(int x, string name, string pass)
114 {
115     ofstream out_file(".\\用户\\用户.txt", ios::app);
116     if (!out_file) exit(-1);
117     out_file << endl << name << ' ' << pass;
118     out_file.close();
119 }
120
121 void User::create_user_cart_file(int x)
122 {
123     string path = ".\\用户\\user";
124     char a[10] = { '\0' };
125     sprintf_s(a, "%d", x);
126     path += a;
127     path += ".txt";
128     ofstream out_file(path, ios::out);
129     if (!out_file) exit(-1);
130     out_file << "ID名称 品牌 价格 数量" << endl;
131     out_file.close();
132 }
133
134 void User::create_user_history_file(int x)

```

```

1  #pragma once
2  #include<iostream>
3  #include<string>
4  #include"Avalon.h"
5  using namespace std;
6  class Gilgamesh
7  {
8  public:
9      int idnum;
10     string id;
11     string password;
12     Gilgamesh() { ... }
18     Gilgamesh(int x) { ... }
24     Gilgamesh(int x, string name, string pass) { ... }
30     void create_admin_file(int x,string name,string pass);
31     void enter_Avalon();
32 };

```

```

1  #pragma once
2  #include<iostream>
3  #include<string>
4  #include"Avalon.h"
5  class User
6  {
7  public:
8      int idnum;
9      string id;
10     string password;
11     User() { ... }
17     User(int x, string name, string pass) { ... }
23     void visit_Avalon();
24     void create_user_file(int x, string name, string pass);
25     void create_user_cart_file(int x);
26     void create_user_history_file(int x);
27     void create_user_order_file(int x);
28     void create_admin_to_user_file(int x);
29 };
30
31

```

```

8 void Gilgamesh:: create_admin_file(int x,string name,string pass)
9 {
10     ofstream out_file(".\\管理员\\管理员.txt", ios::app);
11     if (!out_file) exit(-1);
12     out_file<< endl << x << ' ' << name << ' ' << pass;
13     out_file.close();
14 }
15
16 void Gilgamesh::enter_Avalon()
17 {
18     Avalon gate;
19     system("cls");
20     welcome_info();
21     gate.note_display(0);
22     admin_func_choose();
23     cout << "输入操作: " << endl;
24     string choose;
25     cin >> choose;
26     while (choose != "0")
27     {
28         system("cls");
29         welcome_info();
30         if (choose == "1")
31         {
32             system("cls");
33             welcome_info();
34             gate.admin_display();
35             gate.user_query(0);
36         }
37         else if (choose == "2")
38         {
39             system("cls");
40             welcome_info();
41             gate.admin_display();
42             gate.treasure_add();
43         }

```

本程序与手册最终解释权由南京大学本科生院赵祯烨所有。  
若有疑问，请咨询 201220192@smail.nju.edu.cn

祝使用愉快!  
2021.10.18