



# 目 录

一、课题简介 .....	3
二、系统项目设计 .....	3
三、系统实现 .....	3
1. 二叉树的建立 .....	4
2. 先序遍历 .....	4
a. 递归算法 .....	7
b. 非递归算法 .....	7
3. 中序遍历 .....	6
a. 递归算法 .....	7
b. 非递归算法 .....	7
4. 后序遍历 .....	6
a. 递归算法 .....	7
b. 非递归算法 .....	7
5. 主菜单程序 .....	4
5. 子菜单程序 .....	4
四、系统测试 .....	18
1. 二叉树的建立 .....	4
2. 先序遍历 .....	4
a. 递归算法 .....	7
b. 非递归算法 .....	7

2. 中序遍历 .....	6
a. 递归算法 .....	7
b. 非递归算法 .....	7
3. 后序遍历 .....	6
a. 递归算法 .....	7
b. 非递归算法 .....	7
4. 主菜单程序 .....	4
5. 子菜单程序 .....	4
五、小结 .....	22
六、参考文献 .....	23

### 一. 课题简介:

通过这个课题设计主要掌握三种遍历方法，包括前序遍历，中序遍历和后序遍历，以及后续遍历的非递归算法。

## 二. 项目设计:

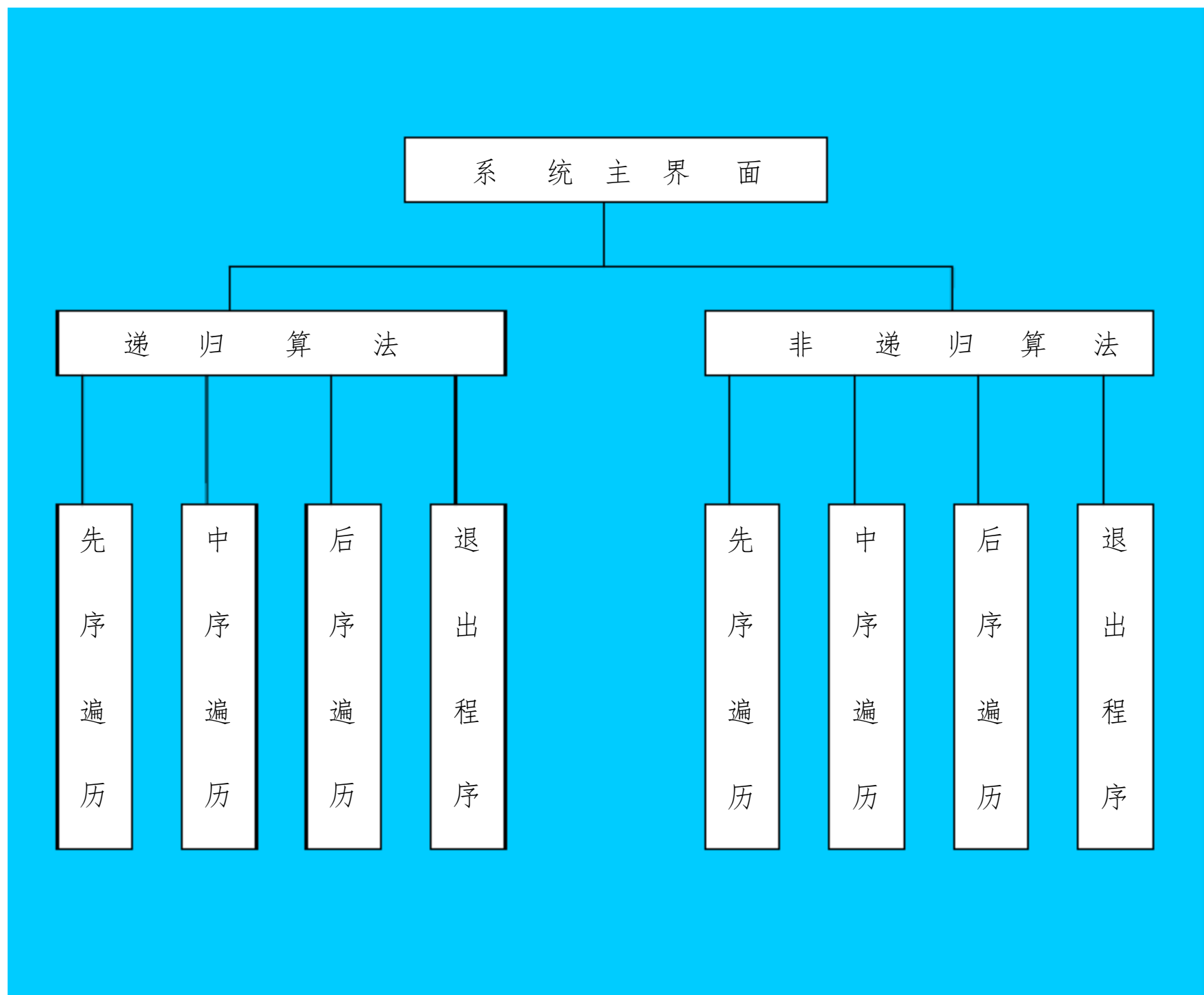


图 1: 系统功能模块图

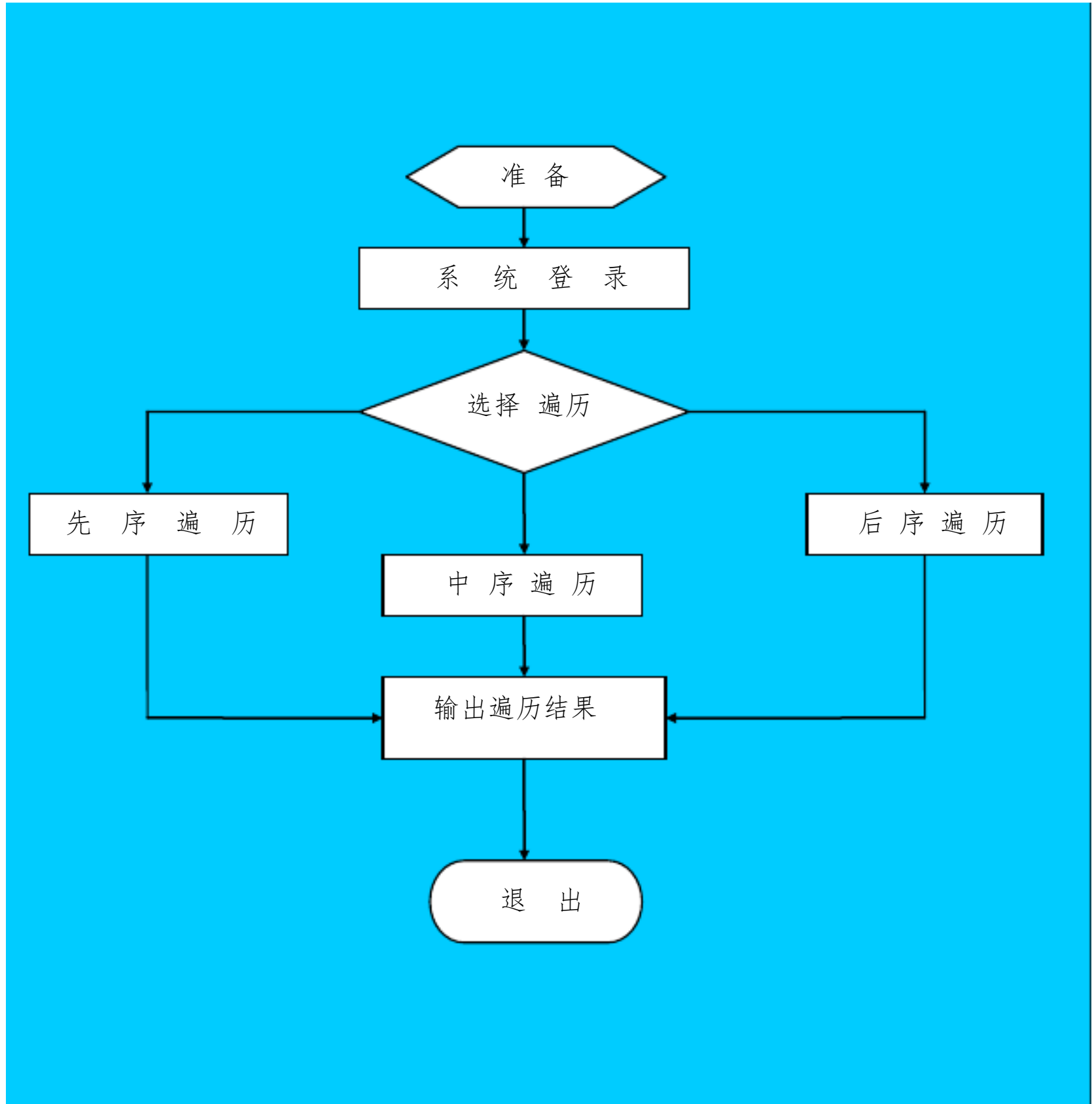


图 2: 系统存盘功能流程图

### 三 系统实现

系统核心代码:

#### 1. 二叉树的建立:

二叉树的遍历算法需要先建立二叉树, 二叉树的建立需要建立栈和数组

栈和数组的建立:

```
typedef struct node                                /*结点定义*/
{
    char    data;
    struct node * lchild, * rchild;
} BinTreeNode;
```

```
typedef struct{ /*栈的定义
    BinTreeNode * ptr;
    int tag;
} StackNode;
```

二叉树的建立:

```
BinTreeNode * CreateBinTree (BinTreeNode * Tree )
/*按先序序列建立二叉树, 输入并建立一棵二叉树 Tree*/
{
    char  c;

    if(c=='&')  Tree = NULL;
    else
    {
        Tree=(BinTreeNode * )malloc(sizeof(BinTreeNode));
        Tree->data=c;
        Tree->lchild= CreateBinTree(Tree->lchild);
        Tree->rchild= CreateBinTree(Tree->rchild);
    }

    return(Tree);
}
```

按先序序列建立二叉树, 输入先序序列:

按先序序列建立二叉树，输入先序序列：  
ab&&c&&

## 2. 先序遍历：

### a. 递归算法：

先序遍历的递归算法：

```
/*二叉树的先序遍历*/  
void PreOrder ( BinTreeNode *T )  
{  
    if( T != NULL )  
    {  
        PreOrder ( T->lchild );  
        PreOrder ( T->rchild );  
    }  
}
```

二叉树的先序遍历：选择菜单

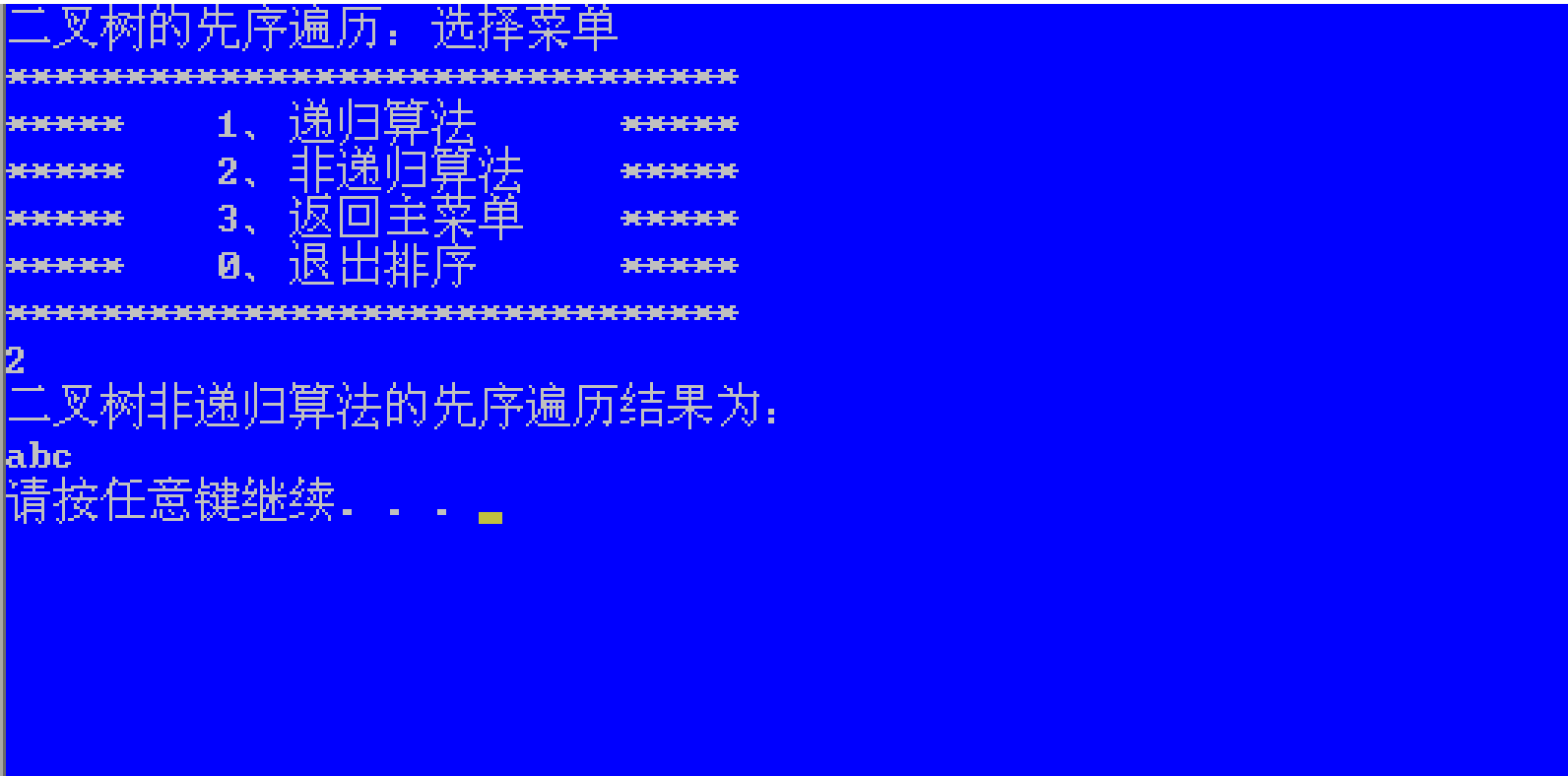
```
*****  
***** 1、递归算法 *****  
***** 2、非递归算法 *****  
***** 3、返回主菜单 *****  
***** 0、退出排序 *****  
*****
```

```
1  
二叉树递归算法的先序遍历结果为：  
abc  
请按任意键继续. . .
```

## b. 非递归算法:

先序遍历的非递归算法:

```
/*二叉树的先序遍历的非递归算法*/
void PreOrderTwo ( BinTreeNode *T)
{
    BinTreeNode *p,*S[Max];
    int top=-1;
    p=T;          /*初始化*/
    do
    {
        while ( p != NULL )
        {
            top++;S[top]=p;
            p=p->lchild;
        }
        if( top >-1 ) /*栈非空*/
        {
            p=S[top]; top--; /*取栈顶元素，出栈*/
            p = p->rchild;
        }
    }while (( p != NULL ) ||(top>-1));
}
```



```
二叉树的先序遍历：选择菜单
*****
***** 1、递归算法 *****
***** 2、非递归算法 *****
***** 3、返回主菜单 *****
***** 0、退出排序 *****
*****
2
二叉树非递归算法的先序遍历结果为：
abc
请按任意键继续. . .
```



## 2. 中序遍历:

检查该用户是否可以使用该系统, 如果没有该用户则重新输入输入, 若用户密码输入错误, 三次错误时, 退出登录系统, 并且该用户被冻结。

```
void common_user()//    普通用户登录
{
    char ch;
    char pass[20];
    char uname[20];
    int i=0;
Lab:
    if(i==3)    exit(1);
                请输入用户名:
    cin>>uname;
    for(user *p= head; p!= NULL;p= p->next)
    {
        if(!strcmp(uname,p->getmingzi()))
        {
            if(p->getstate() >=3)
            {
                该账户已经被冻结!
                i++;
                goto Lab;
            }
            else break;
        }
    }
    if(!p)
    {
        该用户不存在, 请重新输入!
```

```

        goto Lab;
    }
    int x = 0;

    while((ch=getch())!= -1 && ch!= '\n')
    {
        pass[x++]= ch;
        putchar('*');
    }

    if(!strcmp(p->getmimal(), pass))
        while(1)
        {

            int a;

            cout<<endl<<endl<<endl;

```

★欢迎使用本银行★

存 钱

取 钱

转 帐

查 询

挂 失

修改密码

保存信息

返回上级

请输入您的选择

```
cin>>a;
```

```
switch(a)
```

case 4:

{

查询

break;

}

case 7:

{

cunyonghu();

cunzhanghu();

cunguanli();

exit(1);

}

case 0: return;

}

}

else

{

密码错误!

p->setstate(p->getstate()+1); //

如果密码错误, 在以前基础上加状态一并存盘

cunyonghu();

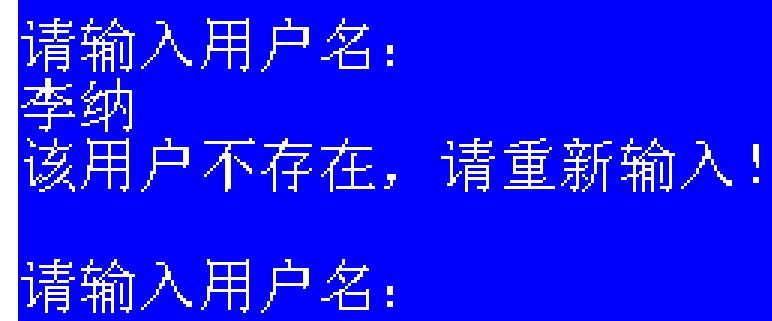
i++;

goto Lab;

}

```
void display() //
{
    p= head;
    while(p!= NULL)
    {
        用户名:
        普通密码:
        状态:
        p=p->next;
    }
}
```

没有此用户:



```
请输入用户名:
李纳
该用户不存在, 请重新输入!
请输入用户名:
```

```
请输入用户名:
123
请输入密码:
****密码错误!

请输入用户名:
123
请输入密码:
**密码错误!

请输入用户名:
123
请输入密码:
****密码错误!
Press any key to continue
```

3) 密码正确:

```
=====★欢迎使用本银行★=====
      ****      存  钱  1      ****
      ****      取  钱  2      ****
      ****      转  帐  3      ****
      ****      查  询  4      ****
      ****      挂  失  5      ****
      ****      修改密码 6      ****
      ****      保存信息 7      ****
      ****      返回上级 0      ****
      ****
请输入您的选择:
```

后序遍历:

检查该用户是否可以使用本系统，三次密码错误则退出登录系统。

```
void manage() // 管理员
{
```

```
char ch;
char pass[20];
char uname[20];
int i=0;
```

Lab:

```
if(i==3) exit(1);

cin>>uname;
for(user *p= head; p!= NULL;p= p->next)
{
    if(!strcmp(uname,p->getmingzi()))
    {
        if(p->getstate() <0 )
        {
            该账户已经被冻结!

            i++;

            goto Lab;
        }
        else break;
    }
}

if(!p)
{
    该用户不存在，请重新输入!

    i++;

    goto Lab;
}

int x = 0;

    请输入密码:

while((ch=getch())!= -1 && ch!= '\n')
```

```
{  
  
    pass[x++] = ch;  
  
    putchar('*');  
  
}
```

```
while(1)
```

```
{
```

```
    int x;
```

```
    cout<<endl<<endl<<endl;
```

```
    =====
```

```
        *****    开    户    1  
        *****    销    户    2  
        *****    查    看    3  
        *****    修改密码    4  
        *****    删除账户    5  
        *****    修改账户状态 6  
        *****    修改用户状态 7  
        *****    保存用户信息 8  
        *****    返回上级    0
```

请输入您的选择

```
cin>>x;
```

```
    switch(x)
```

```
    {
```

```
        case 1:
```

```
        {
```

```
            kaihu();
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/035132344131011122>