
课 程 设 计 报 告

课程名称 计算机程序设计基础（2）

班 级 无 31

学 号 2013011280

姓 名 聂浩

指导助教 无

2014 年 9 月 21 日

一、设计内容与设计要求

1. 课程设计目的

面向对象程序设计课程设计是集中实践性环节之一，是学习完《计算机程序设计基础 2》课程后进行的一次全面的综合练习。要求学生达到熟练掌握 C++ 语言的基本知识和技能；基本掌握面向对象程序设计的思想和方法；能够利用所学的基本知识和技能，解决简单的面向对象程序设计问题，从而提高动手编程解决实际问题的能力。

2. 课题题目

- 1) 学生成绩管理系统
- 2) 图书管理系统

3. 设计要求

1) 设计课题题目：每个同学都完成 2 道课题。 后面有范题，仅供同学们参考，不列入本次课程设计的课题。

2) 根据课题完成以下主要工作：

- ①完成系统需求分析：包括系统设计目的与意义；系统功能需求（系统流程图）；输入输出的要求。
- ②完成系统总体设计：包括系统功能分析；系统功能模块划分与设计（系统功能模块图）。
- ③完成系统详细设计：数据文件；类层次图；界面设计与各功能模块实现。
- ④系统调试：调试出现的主要问题，编译语法错误及修改，重点是运行逻辑问题修改和调整。
- ⑤使用说明书及编程体会：说明如何使用你编写的程序，详细列出每一步的操作步骤。
- ⑥关键源程序（带注释）

3) 按规定格式完成课程设计报告，并在网络学堂上按时提交。

4) 不得抄袭他人程序、课程设计报告，每个人应体现自己的个性设计。

4. 程序设计的基本要求：

(1) 要求利用面向对象的方法以及 C++ 的编程思想来完成系统的设计；

(2) 要求在设计的过程中，建立清晰的类层次；

(3) 每套系统中设计的类的数目不少于 4 个，每个类中要有各自的属性（多于 3 个）和方法（函数多于 3 个）；

(4) 需要定义一个抽象类，采用继承方式派生这些类。并设计一个多重继承的派生类。

(5) 在程序设计中，引入多函数的多态性、运算符重载等机制。

(6) 自己设计测试数据，将测试数据存在文件中，通过文件来进行数据读写来测试。

5. 创新要求：

在基本要求达到后，可进行创新设计，如根据查找结果进行修改的功能。

3、进 度 安 排

小学期 第 1 周			
小学期 第 2 周			

注：1、一定要保留自己那个课题的完整任务书在课程设计报告里面。

2、“评分表”放在“附录：源程序清单”的后面。

图书管理系统

目录

图书管理系统.....	5
1. 系统需求分析	6
2. 总体设计	6
管理员模块.....	6
学生模块.....	7
图书管理系统模块图.....	8
3. 详细设计	8
4. 系统调试	11
5. 使用说明书	11
6. 编程体会	12
附录 1: 源程序	13
lib.h	13
main.cpp	24
md5.h(来自互联网).....	27
md5.cpp(来自互联网).....	29
附录 2: 评分表	38

1. 系统需求分析

图书管理系统是运用于图书馆的书籍借阅和管理系统，这里涉及两种用户，一个是图书管理员，一个是借阅者（这里认为是学生）。系统的界面应该设计的简单明确、美观大方。主要功能一目了然。除此之外，该系统应该做到以下几点：

1. 在用户上应该进行分权限管理，整个系统具有两套界面——只有图书管理员可以进行借阅、增减图书、增加会员的操作；而学生只有查询本人借书情况的权力。整个认证系统使用密码进行加密。
2. 管理员可以随意增减图书。
3. 管理员可以增加学生。
4. 管理员可以查看目前所有图书及借还情况。
5. 应该提醒学生目前的应还书籍。
6. 学生可以通过搜索查询需要的书籍。
7. 系统以 CMD 窗口下的纯文本方式工作，在登陆后进入相应多级菜单完成各种操作。
8. 由于涉及敏感信息，密码部分存储前应该进行加密。

2. 总体设计

图书管理系统包含两个模块，分别是学生模块和图书管理员模块。登陆后自动进入（有 3 次尝试机会）。整个系统菜单详细，用户可以在程序提示下非常方便的使用。

借书信息应该包含三个内容，分别是所借书名，借阅人信息，借还书时间。

管理员模块

使用编号 0 登陆。初始密码为空。管理员模块包括 10 个功能，分别是：

- 1) 显示所有书籍。按书号显示所有书籍的借还情况。
- 2) 显示所有学生。

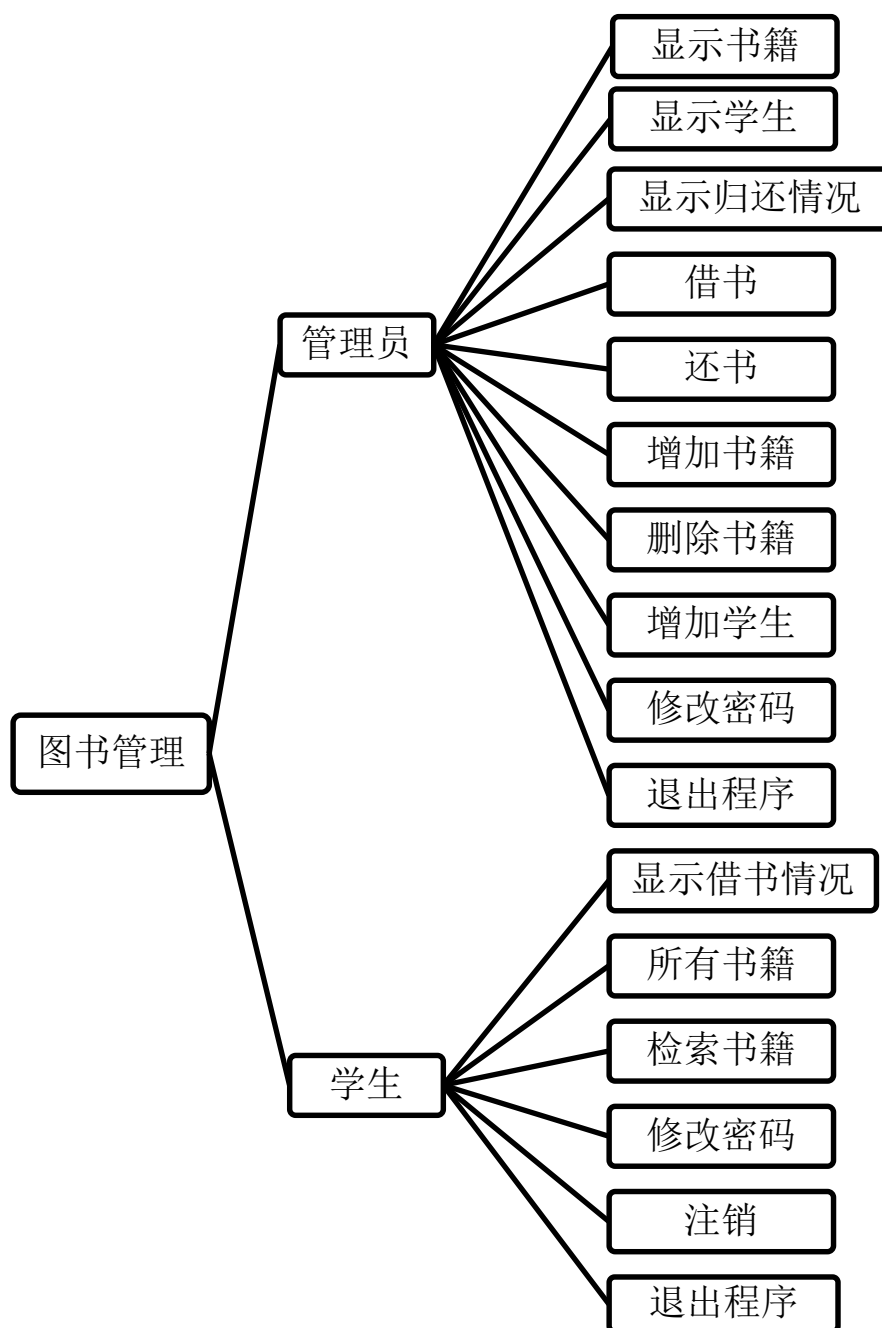
-
- 3) 查询还书信息。将未还书籍优先列出。
 - 4) 借书。通过输入书号、学号、借出天数进行借书
 - 5) 还书。
 - 6) 增加书籍。
 - 7) 删除书籍
 - 8) 增加学生
 - 9) 修改密码。修改管理员密码
 - 0) 退出程序。

学生模块

使用学号登陆后自动跳转至该页面。

- 1) 提醒当前需要归还的书籍（包含是否到期的信息）
- 2) 查询所有书籍
- 3) 按书籍名检索书籍
- 4) 修改密码
- 5) 注销。退回登陆界面
- 6) 退出

图书管理系统模块图



图一

3. 详细设计

由于学生与图书数不可能为确定数量，因此本程序利用 `vector` 容器来存储两种数据，因为每本书只能被借依次，所以没有使用二维数组，只是两个数组通过

存储彼此的信息来形形成联系。

本程序 UML 图如图二。

本程序试图使用 mvc 结构，将大部分 io 封装在 Views 类中；大部分数据结构和操作封装在 Password、STU、Book、List 中，其成员都声明为 protected 型，而 ALL 则通过继承这些类进行二次封装，通过该类 ALL 即可进行所有的数据操作，起到 model 的作用。而 main 函数则是通过对于 ALL 和 Views 相应函数的调用来实现整个程序，起到 controller 的作用。在文件操作时，采用遍历的方法，将所存储的所有信息写入文件中，程序开启时再逐条读出。

因为借还书涉及了时间问题，本程序调用了 time.h,使用了 time 函数和 localtime 函数以进行处理。

另外，由于涉及了密码操作，使用明码存储密码是非常危险与愚蠢的，因此，本程序使用了公布在互联网上的部分代买，对程序中的密码进行了 md5 加密。

菜单按照图书管理系统模块图，采用单级菜单即可。

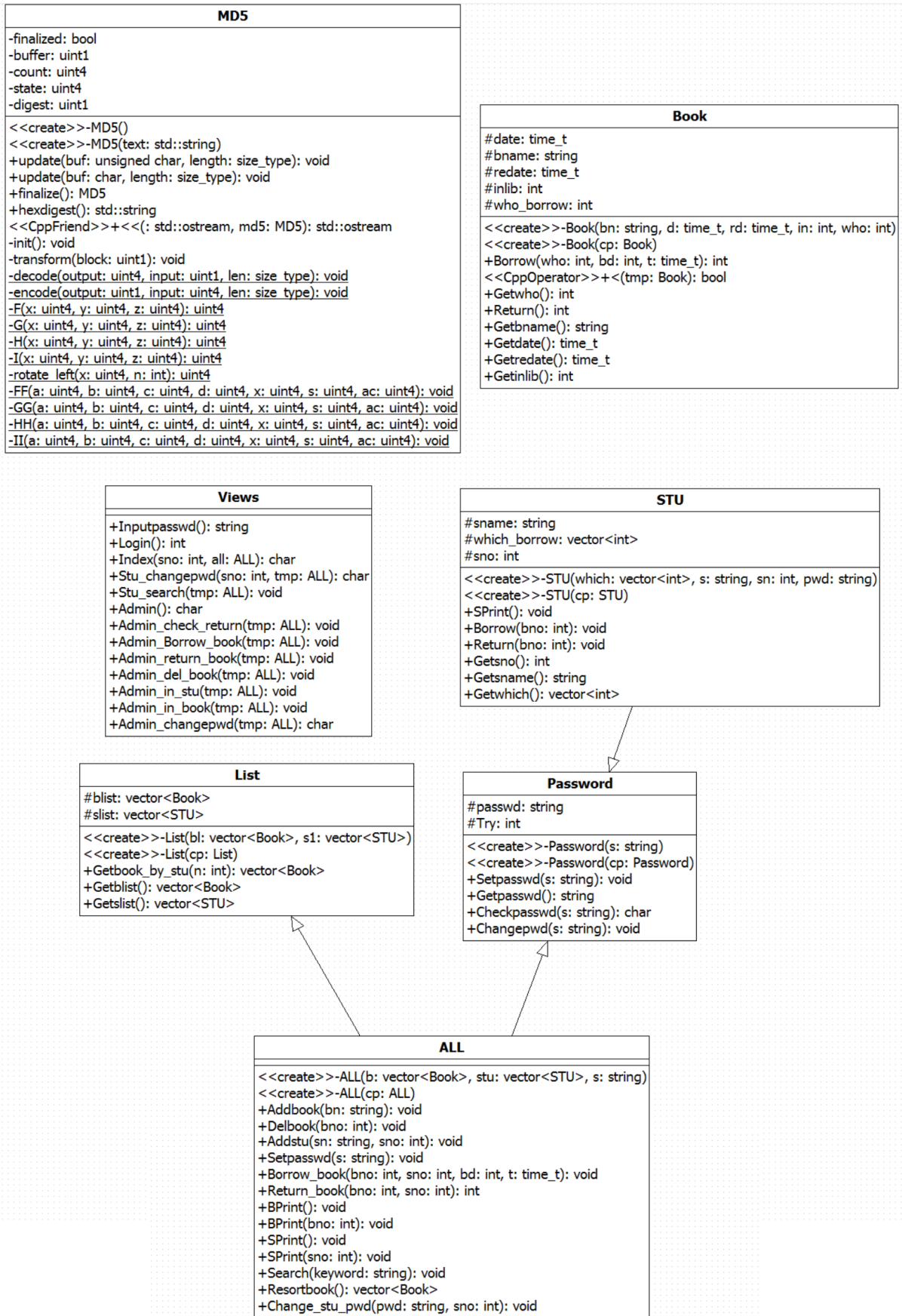


图 2

4. 系统调试

该程序调试中因为吸取了之前程序的教训，所以采取更为明晰的权限管理，继承与对象的使用也更为简洁，所以该程序的调试工作明显减少。

在调试的过程中，最困扰的一个问题是在最后将图书信息写出时，是否借阅选项会被重置为 0，在仔细检查了文件读写、Blist 的调取函数后始终没有发现问题，因此不得不逐句运行，最终发现，这一问题来自于 Book 类的复制函数。

还有一个问题是将已经借阅书籍按照还书日期顺序展示的函数，这里总是无法正常排序，最后明白问题在于重载符<的问题，直接比较 redate 就可以了。

最后，vs2012 的 warning 非常烦人，而且基本没有什么用。

5. 使用说明书

本程序共有两个界面，分别对应于管理员与学生界面。

本程序的管理员账户用户账号为 0，默认无密码（直接回车）。

学生使用学号登陆，在自己没有修改的情况下，无密码（直接按回车）。

密码均可修改且无长度限制，可以使用所有的字母、阿拉伯数字与标点符号。

测试数据已经录入，但通过删除 lib.dat 可以初始化程序（包括密码），但会因此丢失所有数据。

管理员：

管理员具有修改所有数据的能力（除了学生的密码），通过菜单中的选项，管理员可以增添新书籍，删除书籍（会丢失数据）和增加学生（借阅者）。请严格按照学号 姓名的格式输入，否则可能导致程序崩溃。

书籍数量和学生数量均没有限制。（取决于运行计算的内存大小）

管理员可以进行借书和还书操作（还书只需要书籍编号），同时可以方便的查询所有图书、学生，并对已经借出的图书进行管理（会按照还书日期进行排序，已经到期的书籍会有提醒）。

学生：

学生只有查询的权利，学生可以使用两种方式查找书籍。1.查看所有书籍列表；2.通过书名搜索书籍。在学生界面的首页，会显示学生当前的借书情况，对于已到期书籍会有提醒。

6. 编程体会

之前也已经完成了大作业的第一题——学生成绩管理系统，因此从里面获得了不少有用的经验。

在本程序中，仍然使用类似 `mvc` 的结构，数据结构仍然使用了 `vector`，但这次没有引入二维的情况，只是分别加入了一定的信息来实现功能。在类的设定上进一步优化，减少了权限问题，很大程度上减少了函数的重复。

本程序和学生成绩管理系统很相似，不过，我采用了不同的数据结构，但之前的经验也教训确实教会我很多。

还有，因为在编制本程序前我进行了细致的，一些变量采用了更为合理的数据类型，在编程时减少了很多麻烦，因此，细致的规划是很有必要的。

总之，大作业能为我们平时的编程带来很多新思路与注意事项，是一个非常好的提升机会。

附录 1：源程序

lib.h

```
#include <iostream>
#include <string>
#include <sstream>
#include <stdio.h>
#include <vector>
#include <stdlib.h>
#include "md5.h"
#include <windows.h> //for windows
#include <time.h>
#include <conio.h>
using namespace std;

string Time(time_t t) {
    ostringstream s("");
    tm* tmp=localtime(&t);
    s<<tmp->tm_year+1900<<"-"<<tmp->tm_mon+1<<"-"<<tmp->tm_mday;
    return s.str();
}

class Password{
public:
    Password(string s=md5("")):passwd(s) {Try=0;}
    Password(const Password& cp):passwd(cp.passwd) {Try=0;}
    void Setpasswd(string s="") {passwd=s;}
    string Getpasswd() {return passwd;}
    char Checkpasswd(string s="") {
        Try++;
        if(Try<3) {
            if(passwd=="74be16979710d4c4e7c6647856088456")
                return 1;
            else if(s==Getpasswd())
                return 1;
            return 0;
        }
        else {
            Try=0;
            exit(0);
        }
    }
}
```

```

    }
    void Changepwd(string s="") {passwd=s;}
protected:
    string passwd;
    int Try;//限制尝试次数
};

class STU:public Password{
public:
    STU(vector<int> which,string s="",int sn=0,string
pwd=md5("")):sname(s),sno(sn),
    which_borrow(which),Password(pwd) {}
    STU(const STU& cp):sname(cp.sname),Password(cp),sno(cp.sno) {}
    void SPrint() {cout<<sno<<". "<<sname<<endl;}
    void Borrow(int bno) {
        which_borrow.push_back(bno);
    }
    void Return(int bno) {
        for(int i=0;i<which_borrow.size();i++) {
            if(which_borrow[i]==bno) {
                which_borrow.erase(which_borrow.begin()+i);
                break;
            }
        }
    }
    int Getsno() {return sno;}
    string Getsname() {return sname;}
    vector<int> Getwhich() {return which_borrow;}
protected:
    string sname;
    vector<int> which_borrow;
    int sno;
};

class Book
{
public:
    Book(string bn="",time_t d=0,time_t rd=1,int in=1,int
who=-1):bname(bn),date(d),redate(rd),inlib(in),who_borrow(who) {}
    Book(const Book
&cp):bname(cp.Getbname()),date(cp.Getdate()),redate(cp.Getredate()),i
nlib(cp.Getinlib()),who_borrow(cp.Getwho()) {

    }

```

```

int Borrow(int who=0, int bd=0, time_t t=time(NULL)) {
    if(inlib==1) {
        date=t;
        redate=date+bd*86400;
        inlib=0;
        who_borrow=who;
        return 1;
    }
    else return 0;
}

bool operator<(const Book& tmp)
{return (redate<tmp.redate);}
int Getwho()const {return who_borrow;}
int Return() {
    date=0;
    inlib=1;
    if(time(NULL)>redate) {
        redate=1;
        return who_borrow;
    }
    redate=0;
    return 0;
}

string Getbname() const{return bname;}
time_t Getdate() const{return date;}
time_t Getredate() const {return redate;}
int Getinlib() const {return inlib;}

protected:
    time_t date;//借出日期
    string bname;//书名
    time_t redate;//还书日期
    int inlib;//1表示在图书馆,0表示已借出
    int who_borrow;

};

class List
{
public:
    List(vector<Book> b1, vector<STU> s1):blist(b1),slist(s1) {}
    List( List &cp):blist(cp.Getblist()),slist(cp.Getslislist()) {}
    vector<Book> Getbook_by_stu(int n) {
        vector<Book> tmp;

```

```

        for (int i=0;i<blist.size();i++) {
            if(blist[i].Getwho()==n)
                tmp.push_back(blist[i]);
        }
        return tmp;
    }
    vector<Book> Getblist() {return blist;}
    vector<STU> Getslist() {return slist;}
protected:
    vector<Book> blist;
    vector<STU> slist;
};

class ALL:public Password,public List{
public:
    ALL(vector<Book> b,vector<STU> stu,string
s=md5("")):Password(s),List(b,stu) {}
    ALL(const ALL& cp):Password(cp),List(cp.blist,cp.slist) {}
    void Addbook(string bn="") {
        Book tmp(bn);
        blist.push_back(tmp);
    }
    void Delbook(int bno) {
        blist.erase(blist.begin()+bno);
    }
    void Addstu(string sn="",int sno=0) {
        vector<int> v;
        STU tmp(v,sn,sno);
        slist.push_back(tmp);
    }
    void Setpasswd(string s="") {
        passwd=s;
    }
    void Borrow_book(int bno,int sno,int bd,time_t t=time(NULL)) {
        if(blist[bno].Borrow(sno,bd,t)) {
            slist[sno].Borrow(bno);
            cout<<"借书成功"<<endl;
        }
        else cout<<"本书不在架上!!"<<endl;
    }
    int Return_book(int bno,int sno) {
        slist[sno].Return(bno);
        return blist[bno].Return();
    }
};

```

```

void BPrint() {
    for(int i=0;i<blist.size();i++)    BPrint(i);
}
void BPrint(int bno) {
    cout<<bno<<". "<<blist[bno].Getbname()<<endl;
    if(blist[bno].Getinlib()==0)
        cout<<"\t 被借于: "<<Time(blist[bno].Getdate())<<"  "<<"于:
"<<Time(blist[bno].Getredate())<<"截止";
    if(blist[bno].Getinlib()==0&&blist[bno].Getredate()<time(NULL))
cout<<"\t\t状态: 正在催还";
    cout<<endl;
}
void SPrint() {
    for(int i=0;i<slist.size();i++) SPrint(i);
}
void SPrint(int sno) {
    slist[sno].SPrint();
    if(slist[sno].Getwhich().size()!=0) {
        for(int j=slist[sno].Getwhich().size()-1;j>-1;j--) {
            cout<<"\t";
            BPrint(slist[sno].Getwhich()[j]);
        }
    }
}
void Search(string keyword);
vector<Book> Resortbook();
void Change_stu_pwd(string pwd,int sno) {
    slist[sno].Setpasswd(pwd);
}
};
void ALL::Search(string keyword) {
    int i=0;
    for(i=0;i<blist.size();i++) {
        if(blist[i].Getbname()==keyword) {
            BPrint(i);
            break;
        }
    }
    if(blist.size()-1==i) cout<<"该书不存在";
}
vector<Book> ALL::Resortbook() {
    vector<Book> tblest=blist;
    for(int i=1;i<tblest.size();i++) {
        for(int j=0;j<(tblest.size()-i);i++) {

```

```

        if(tblist[j]<tblist[j+1])
        swap(tblist[j],tblist[j+1]);
    }
}
return tblist;
}

class Views{
public:
    string Inputpasswd() {
        char temp_c=42,length=0;
        string pwd="";
        while(true)
        {
            temp_c=getch(); //输入一个字符
            if(temp_c!=char(13)) //判断该字符是不为回车，如果是则退出
while
        {
            switch (temp_c)
            {
            case 8:
                if(length!=0)
                {
                    cout<<"\b \b";
                    pwd=pwd.substr(0,length-1);
                    length--;
                }
                else ;
                break;
            default:
                cout<<"*"; //可用用你喜欢的任意字符，如改为cout<<" ";
则无回显

                pwd+=temp_c;//连成字符串;
                length++;
                break;
            }
        }
        else break;
    }
    return md5(pwd);
}

int Login() {
    int name;
    system("cls");

```

```

        cout<<"*****"
        <<"\n\n\n\n 请输入学号:"; //管理员帐号编码0
        cin>>name;
        return name;
    }
    //学生用户登录后的行为
    char Index(int sno, ALL & all) {
        int select=42;
        system("cls");
        //for win system("cls");

        cout<<"_____ \n
\n"
        <<"\t\t欢迎使用图书馆管理系统\n\n"

        <<"_____ \n\n\n
\n";
        all.SPrint(sno);
        cout<<"\n\n1. 查询所有书籍\n\n"
        <<"2. 查询所需书籍\n\n"
        <<"3. 修改密码\n\n"
        <<"4. 注销\n\n"
        <<"5. 退出\n\n\n\n"
        <<"请输入需要的操作: ";
        cin>>select;
        return select;
    }
    char Stu_changepwd(int sno, ALL& tmp) {
        system("cls");
        cout<<"\n请输入旧密码:";
        string s=Inputpasswd(), s1;
        char judge=tmp.Getslst()[sno].Checkpasswd(s);
        if(judge==0) {
            cout<<"\n\n错误的密码, 修改失败";
            Sleep(2000);
            return 0;
        }
        cout<<"\n\n请输入新密码:"<<endl;
        s=Inputpasswd();
        cout<<"\n请再次输入新密码:"<<endl;
        s1=Inputpasswd();
        if(s!=s1) {
            cout<<"\n\n两次输入不一样";
            Sleep(1000);

```

```

        return 0;
    }
    tmp.Change_stu_pwd(s, sno);
    cout<<"\n\n修改成功";
    Sleep(1000);
    return 1;
}
void Stu_search(ALL &tmp) {
    system("cls");
    string key;
    char select=0;

    for(;select!='n';) {
        cout<<"请输入需要查询的书籍名:";
        cin>>key;
        tmp.Search(key);
        cout<<"是否继续[y/n]";
        cin>>select;
    }
}
//管理员用户登录后的行为
char Admin() {
    int select=42;
    system("cls");
    //for win system("cls");

    cout<<"_____ \n
"

    <<"\t\t欢迎使用图书馆管理系统\n"

    <<"_____ \n\n"
    <<"1. 显示所有书籍\n"
    <<"2. 显示所有学生\n"
    <<"3. 查询还书信息\n\n"
    <<"4. 借书\n"
    <<"5. 还书\n"
    <<"6. 增加书籍\n"
    <<"7. 删除书籍\n"
    <<"8. 增加学生\n"
    <<"9. 修改密码\n"
    <<"0. 退出程序\n\n"
    <<"请输入需要的操作： ";
    cin>>select;
    return select;
}

```

```

    }
    void Admin_check_return(ALL& tmp) {
        system("cls");
        vector<Book> blist=tmp.Resortbook();
        for(int i=0;i<blist.size();i++) {
            if(blist[i].Getinlib()==0) {
                cout<<i<<". "<<blist[i].Getbname()<<endl;
                cout<<"\t 被借于: "<<Time(blist[i].Getdate())<<" "<<"
于: "<<Time(blist[i].Getredate())<<"截止";

                if(blist[i].Getredate()<time(NULL)&&blist[i].Getinlib()==0)
cout<<"\t\t状态: 正在催还";
                cout<<endl;
            }
        }
    }
    void Admin_Borrow_book(ALL &tmp) {
        system("cls");
        int user_id=-1,bno=-1,bd=7,i;
        char select=0;
        for(;select!='n';) {
            tmp.SPrint();
            cout<<"请选择借书的同学的学号:";
            cin>>user_id;

            for(i=0;(i<tmp.Getslis().size())&&(tmp.Getslis()[i].Getsno()!=u
ser_id);i++);
            if(i==tmp.Getslis().size()) {
                cout<<"用户不存在";
                Sleep(1000);
                return;
            }
            tmp.BPrint();
            cout<<"请选择所需书籍 (输入-1停止操作)";
            cin>>bno;
            if(bno==-1) break;
            cout<<"请输入本书还书期限 (天)";
            cin>>bd;
            tmp.Borrow_book(bno,i,bd);
            cout<<"是否继续[y/n]";
            cin>>select;
        }
    }
}

```

```

void Admin_return_book(ALL &tmp) {
    system("cls");
    int sno=-1, bno=-1;
    char select=0;
    vector<Book> blist=tmp.Getblist();
    for(;select!='n';) {
        for(int i=0; i<blist.size(); i++) {
            if(blist[i].Getinlib()==0) {
                cout<<i<<". "<<blist[i].Getbname()<<endl;
                cout<<"\t 被借于: "<<Time(blist[i].Getdate())<<"
"<<"于: "<<Time(blist[i].Getredate())<<"截止";

                if(blist[i].Getredate()<time(NULL)&&blist[i].Getinlib()==0)
                    cout<<"\t\t状态: 正在催还";
                cout<<endl;
            }
        }
        cout<<"请选择要还的书籍: (输入-1停止操作) ";
        cin>>bno;
        if(bno==-1) break;
        tmp.Return_book(bno, tmp.Getblist()[bno].Getwho());
        cout<<"是否继续[y/n]";
        cin>>select;
        blist=tmp.Getblist();
    }
}

void Admin_del_book(ALL &tmp) {
    system("cls");
    int bno=-1;
    char select=0;
    for(;select!='n';) {
        tmp.BPrint();
        cout<<"请选择要删除的书籍: (输入-1停止操作) ";
        cin>>bno;
        if(bno==-1) break;
        tmp.Delbook(bno);
        cout<<"是否继续[y/n]";
        cin>>select;
    }
}

void Admin_in_stu(ALL &tmp) {
    system("cls");
    string sn;

```

```

    char select=0;
    int sno=0;
    for(;select!='n';){
        cout<<"请输入新的学生的学号与姓名:";
        cin>>sno>>sn;
        tmp.Addstu(sn, sno);
        cout<<"是否继续[y/n]";
        cin>>select;
    }
}

void Admin_in_book(ALL & tmp){
    system("cls");
    string bn;
    char select=0;
    for(;select!='n';){
        cout<<"请输入新的书籍名:";
        cin>>bn;
        tmp.Addbook(bn);
        cout<<"是否继续[y/n]";
        cin>>select;
    }
}

char Admin_changepwd(ALL& tmp){
    system("cls");
    cout<<"请输入旧密码:";
    string s=Inputpasswd(), s1;
    char judge=tmp.Checkpasswd(s);
    if(judge==0){
        cout<<"\n\n错误的密码, 修改失败";
        Sleep(2000);
        return 0;
    }
    cout<<"\n\n请输入新密码:"<<endl;
    s=Inputpasswd();
    cout<<"\n请再次输入新密码:"<<endl;
    s1=Inputpasswd();
    if(s!=s1){
        cout<<"\n\n两次输入不一样";
        Sleep(1000);
        return 0;
    }
    tmp.Setpasswd(s);
    cout<<"\n\n修改成功";
    Sleep(1000);
}

```

```
        return 1;
    }
};
```

main.cpp

```
#include<iostream>
#include<string>
#include<vector>
#include<fstream>
#include "md5.h"
#include "lib.h"
class ALL;
class Views;
int main(void) {
    string s=md5("123456"), sname="admin", pwd=md5("");
    int sno=1, bno=-1, bd=0, i=0;
    time_t date;
    vector<Book> blist;
    vector<STU> stmp;
    fstream ifile("lib.dat", ios_base::in);
    if(ifile) {
        ifile>>pwd;
    }
    ALL all(blist, stmp, pwd);
    if(ifile) {
        //导入学生
        ifile>>sno;
        while(sno!=-1) {
            ifile>>sname>>pwd;
            all.Addstu(sname, sno);
            all.Change_stu_pwd(pwd, all.Getslst().size()-1);
            ifile>>sno;
        }
        //导入书本
        int who=-2;
        string bname;
        ifile>>who;
        while(who!=-2) {
            ifile>>bname;
            all.Addbook(bname);
            ifile>>who;
        }
    }
}
```

```

    }
    //导入借书情况
    ifile>>sno;
    while(sno!=-1){
        ifile>>date>>bno>>bd;
        all.Borrow_book(bno, sno, bd, date);
        ifile>>sno;
    }
}
ifile.close();

Views view;
int select=42;
//登录模块
int user_id=42;
Begin: for(char check=0;check==0;){
    user_id=view.Login();
    if(user_id==0){
        char a=0;
        while(a!='\n'){a= getchar();}
        cout<<"请输入密码:";
        string s=view.Inputpasswd();
        check=all.Checkpasswd(s);
    }
    else{

        for(i=0;(i<all.Getslis().size())&&(all.Getslis()[i].Getsno()!=u
ser_id);i++){
            if(i!=all.Getslis().size()){
                cout<<"请输入密码:";
                string s=view.Inputpasswd();
                check=all.Getslis()[i].Checkpasswd(s);
            }
            else {
                cout<<"用户不存在";
                Sleep(1000);
                goto Begin;
            }
        }
        if(check==0){
            cout<<"\n密码输入错误\n";
            Sleep(1000);
        }
    }
}

```

```

//STU模块
if(user_id!=0) {
    for(int j=0;j==0;) {
        select=view.Index(i,all);
        switch(select) {
            case 1:system("cls");all.BPrint();cout<<"输入任意键继续:";
            getch();getchar();break;
            case 2: view.Stu_search(all);break;
            case 3: view.Stu_changepwd(i,all);break;
            case 4: goto Begin; break;
            case 5: j=1;break;
            default: cout<<"无效的选项";
                    Sleep(1000);
        }
    }
}
else{//admin模块
    for(int j=0;j==0;) {
        select=view.Admin();
        switch(select) {
            case 1: system("cls"); all.BPrint();cout<<"输入任意键继续:";getch();getchar();break;
            case 2: system("cls"); all.SPrint();cout<<"输入任意键继续:";getch();getchar();break;
            case 3: view.Admin_check_return(all);cout<<"输入任意键继续:";getch();getchar();break;
            case 4: view.Admin_Borrow_book(all);break;
            case 5: view.Admin_return_book(all);break;
            case 6: view.Admin_in_book(all);break;
            case 7: view.Admin_del_book(all);break;
            case 8: view.Admin_in_stu(all);break;
            case 9: view.Admin_changepwd(all);break;
            case 0: j=1;break;
            default: cout<<"无效的选项, 按任意键继续";
                    cin>>select;
        }
    }
}
//输出所有信息
stmp=all.Getslis();
fstream of("lib.dat",ios_base::out);
of<<all.Getpasswd()<<endl;
for(int j=0;j<stmp.size();j++)
    of<<stmp[j].Getsno()<<" "<<stmp[j].Getsname()<<"

```

```

"<<stmp[j].Getpasswd()<<" ";
    of<<-1<<endl;
    blist=all.Getblist();
    for(int j=0;j<blist.size();j++)
        of<<blist[j].Getwho()<<" "<<blist[j].Getbname()<<" ";
    of<<-2<<endl;
    for(int j=0;j<blist.size();j++){
        if(blist[j].Getinlib()==0){
            of<<blist[j].Getwho()<<" "<<blist[j].Getdate()<<"
"<<j<<" "<<(blist[j].Getredate()-blist[j].Getdate())/ 86400<<" ";
        }
    }
    of<<-1;
    of.close();
    return 0;
}

```

md5.h(来自互联网)

```

//MD5.h
#ifndef BZF_MD5_H
#define BZF_MD5_H

#include <string>
#include <iostream>

// a small class for calculating MD5 hashes of strings or byte arrays
// it is not meant to be fast or secure
//
// usage: 1) feed it blocks of uchars with update()
//         2) finalize()
//         3) get hexdigest() string
//         or
//         MD5(std::string).hexdigest()
//
// assumes that char is 8 bit and int is 32 bit
class MD5
{
public:
    typedef unsigned int size_type; // must be 32bit

```

```

MD5();
MD5(const std::string& text);
void update(const unsigned char *buf, size_type length);
void update(const char *buf, size_type length);
MD5& finalize();
std::string hexdigest() const;
friend std::ostream& operator<<(std::ostream&, MD5 md5);

private:
    void init();
    typedef unsigned char uint1; // 8bit
    typedef unsigned int uint4; // 32bit
    enum {blocksize = 64}; // VC6 won't eat a const static int here

    void transform(const uint1 block[blocksize]);
    static void decode(uint4 output[], const uint1 input[], size_type len);
    static void encode(uint1 output[], const uint4 input[], size_type len);

    bool finalized;
    uint1 buffer[blocksize]; // bytes that didn't fit in last 64 byte chunk
    uint4 count[2]; // 64bit counter for number of bits (lo, hi)
    uint4 state[4]; // digest so far
    uint1 digest[16]; // the result

    // low level logic operations
    static inline uint4 F(uint4 x, uint4 y, uint4 z);
    static inline uint4 G(uint4 x, uint4 y, uint4 z);
    static inline uint4 H(uint4 x, uint4 y, uint4 z);
    static inline uint4 I(uint4 x, uint4 y, uint4 z);
    static inline uint4 rotate_left(uint4 x, int n);
    static inline void FF(uint4 &a, uint4 b, uint4 c, uint4 d, uint4 x, uint4 s,
uint4 ac);
    static inline void GG(uint4 &a, uint4 b, uint4 c, uint4 d, uint4 x, uint4 s,
uint4 ac);
    static inline void HH(uint4 &a, uint4 b, uint4 c, uint4 d, uint4 x, uint4 s,
uint4 ac);
    static inline void II(uint4 &a, uint4 b, uint4 c, uint4 d, uint4 x, uint4 s,
uint4 ac);
};

std::string md5(const std::string str);

#endif

```

md5.cpp(来自互联网)

```
//MD5.cpp
/* MD5
converted to C++ class by Frank Thilo (thilo@unix-ag.org)
for bzflag (http://www.bzflag.org)

based on:

md5.h and md5.c
reference implementation of RFC 1321

Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All
rights reserved.

License to copy and use this software is granted provided that it
is identified as the "RSA Data Security, Inc. MD5 Message-Digest
Algorithm" in all material mentioning or referencing this software
or this function.

License is also granted to make and use derivative works provided
that such works are identified as "derived from the RSA Data
Security, Inc. MD5 Message-Digest Algorithm" in all material
mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either
the merchantability of this software or the suitability of this
software for any particular purpose. It is provided "as is"
without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this
documentation and/or software.

*/

/* interface header */
#include "md5.h"

/* system implementation headers */
#include <stdio.h>
#include <cstring>
```

```

// Constants for MD5Transform routine.
#define S11 7
#define S12 12
#define S13 17
#define S14 22
#define S21 5
#define S22 9
#define S23 14
#define S24 20
#define S31 4
#define S32 11
#define S33 16
#define S34 23
#define S41 6
#define S42 10
#define S43 15
#define S44 21

////////////////////////////////////

// F, G, H and I are basic MD5 functions.
inline MD5::uint4 MD5::F(uint4 x, uint4 y, uint4 z) {
    return x&y | ~x&z;
}

inline MD5::uint4 MD5::G(uint4 x, uint4 y, uint4 z) {
    return x&z | y&~z;
}

inline MD5::uint4 MD5::H(uint4 x, uint4 y, uint4 z) {
    return x^y^z;
}

inline MD5::uint4 MD5::I(uint4 x, uint4 y, uint4 z) {
    return y ^ (x | ~z);
}

// rotate_left rotates x left n bits.
inline MD5::uint4 MD5::rotate_left(uint4 x, int n) {
    return (x << n) | (x >> (32-n));
}

// FF, GG, HH, and II transformations for rounds 1, 2, 3, and 4.

```

```

// Rotation is separate from addition to prevent recomputation.
inline void MD5::FF(uint4 &a, uint4 b, uint4 c, uint4 d, uint4 x, uint4 s, uint4
ac) {
    a = rotate_left(a+ F(b,c,d) + x + ac, s) + b;
}

inline void MD5::GG(uint4 &a, uint4 b, uint4 c, uint4 d, uint4 x, uint4 s, uint4
ac) {
    a = rotate_left(a + G(b,c,d) + x + ac, s) + b;
}

inline void MD5::HH(uint4 &a, uint4 b, uint4 c, uint4 d, uint4 x, uint4 s, uint4
ac) {
    a = rotate_left(a + H(b,c,d) + x + ac, s) + b;
}

inline void MD5::II(uint4 &a, uint4 b, uint4 c, uint4 d, uint4 x, uint4 s, uint4
ac) {
    a = rotate_left(a + I(b,c,d) + x + ac, s) + b;
}

////////////////////////////////////

// default ctor, just initailize
MD5::MD5()
{
    init();
}

////////////////////////////////////

// nifty shortcut ctor, compute MD5 for string and finalize it right away
MD5::MD5(const std::string &text)
{
    init();
    update(text.c_str(), text.length());
    finalize();
}

////////////////////////////////////

void MD5::init()
{
    finalized=false;

```

```

    count[0] = 0;
    count[1] = 0;

    // load magic initialization constants.
    state[0] = 0x67452301;
    state[1] = 0xefcdab89;
    state[2] = 0x98badcfe;
    state[3] = 0x10325476;
}

////////////////////////////////////

// decodes input (unsigned char) into output (uint4). Assumes len is a multiple of
4.
void MD5::decode(uint4 output[], const uint1 input[], size_type len)
{
    for (unsigned int i = 0, j = 0; j < len; i++, j += 4)
        output[i] = ((uint4)input[j]) | (((uint4)input[j+1]) << 8) |
            (((uint4)input[j+2]) << 16) | (((uint4)input[j+3]) << 24);
}

////////////////////////////////////

// encodes input (uint4) into output (unsigned char). Assumes len is
// a multiple of 4.
void MD5::encode(uint1 output[], const uint4 input[], size_type len)
{
    for (size_type i = 0, j = 0; j < len; i++, j += 4) {
        output[j] = input[i] & 0xff;
        output[j+1] = (input[i] >> 8) & 0xff;
        output[j+2] = (input[i] >> 16) & 0xff;
        output[j+3] = (input[i] >> 24) & 0xff;
    }
}

////////////////////////////////////

// apply MD5 algo on a block
void MD5::transform(const uint1 block[blocksize])
{
    uint4 a = state[0], b = state[1], c = state[2], d = state[3], x[16];
    decode (x, block, blocksize);

```

```

/* Round 1 */
FF (a, b, c, d, x[ 0], S11, 0xd76aa478); /* 1 */
FF (d, a, b, c, x[ 1], S12, 0xe8c7b756); /* 2 */
FF (c, d, a, b, x[ 2], S13, 0x242070db); /* 3 */
FF (b, c, d, a, x[ 3], S14, 0xc1bdceee); /* 4 */
FF (a, b, c, d, x[ 4], S11, 0xf57c0faf); /* 5 */
FF (d, a, b, c, x[ 5], S12, 0x4787c62a); /* 6 */
FF (c, d, a, b, x[ 6], S13, 0xa8304613); /* 7 */
FF (b, c, d, a, x[ 7], S14, 0xfd469501); /* 8 */
FF (a, b, c, d, x[ 8], S11, 0x698098d8); /* 9 */
FF (d, a, b, c, x[ 9], S12, 0x8b44f7af); /* 10 */
FF (c, d, a, b, x[10], S13, 0xffff5bb1); /* 11 */
FF (b, c, d, a, x[11], S14, 0x895cd7be); /* 12 */
FF (a, b, c, d, x[12], S11, 0x6b901122); /* 13 */
FF (d, a, b, c, x[13], S12, 0xfd987193); /* 14 */
FF (c, d, a, b, x[14], S13, 0xa679438e); /* 15 */
FF (b, c, d, a, x[15], S14, 0x49b40821); /* 16 */

/* Round 2 */
GG (a, b, c, d, x[ 1], S21, 0xf61e2562); /* 17 */
GG (d, a, b, c, x[ 6], S22, 0xc040b340); /* 18 */
GG (c, d, a, b, x[11], S23, 0x265e5a51); /* 19 */
GG (b, c, d, a, x[ 0], S24, 0xe9b6c7aa); /* 20 */
GG (a, b, c, d, x[ 5], S21, 0xd62f105d); /* 21 */
GG (d, a, b, c, x[10], S22, 0x2441453); /* 22 */
GG (c, d, a, b, x[15], S23, 0xd8a1e681); /* 23 */
GG (b, c, d, a, x[ 4], S24, 0xe7d3fbc8); /* 24 */
GG (a, b, c, d, x[ 9], S21, 0x21e1cde6); /* 25 */
GG (d, a, b, c, x[14], S22, 0xc33707d6); /* 26 */
GG (c, d, a, b, x[ 3], S23, 0xf4d50d87); /* 27 */
GG (b, c, d, a, x[ 8], S24, 0x455a14ed); /* 28 */
GG (a, b, c, d, x[13], S21, 0xa9e3e905); /* 29 */
GG (d, a, b, c, x[ 2], S22, 0xfcefa3f8); /* 30 */
GG (c, d, a, b, x[ 7], S23, 0x676f02d9); /* 31 */
GG (b, c, d, a, x[12], S24, 0x8d2a4c8a); /* 32 */

/* Round 3 */
HH (a, b, c, d, x[ 5], S31, 0xffffa3942); /* 33 */
HH (d, a, b, c, x[ 8], S32, 0x8771f681); /* 34 */
HH (c, d, a, b, x[11], S33, 0x6d9d6122); /* 35 */
HH (b, c, d, a, x[14], S34, 0xfde5380c); /* 36 */
HH (a, b, c, d, x[ 1], S31, 0xa4beea44); /* 37 */
HH (d, a, b, c, x[ 4], S32, 0x4bdecfa9); /* 38 */
HH (c, d, a, b, x[ 7], S33, 0xf6bb4b60); /* 39 */

```

```

HH (b, c, d, a, x[10], S34, 0xbebfb70); /* 40 */
HH (a, b, c, d, x[13], S31, 0x289b7ec6); /* 41 */
HH (d, a, b, c, x[ 0], S32, 0xea127fa); /* 42 */
HH (c, d, a, b, x[ 3], S33, 0xd4ef3085); /* 43 */
HH (b, c, d, a, x[ 6], S34, 0x4881d05); /* 44 */
HH (a, b, c, d, x[ 9], S31, 0xd9d4d039); /* 45 */
HH (d, a, b, c, x[12], S32, 0xe6db99e5); /* 46 */
HH (c, d, a, b, x[15], S33, 0x1fa27cf8); /* 47 */
HH (b, c, d, a, x[ 2], S34, 0xc4ac5665); /* 48 */

/* Round 4 */
II (a, b, c, d, x[ 0], S41, 0xf4292244); /* 49 */
II (d, a, b, c, x[ 7], S42, 0x432aff97); /* 50 */
II (c, d, a, b, x[14], S43, 0xab9423a7); /* 51 */
II (b, c, d, a, x[ 5], S44, 0xfc93a039); /* 52 */
II (a, b, c, d, x[12], S41, 0x655b59c3); /* 53 */
II (d, a, b, c, x[ 3], S42, 0x8f0ccc92); /* 54 */
II (c, d, a, b, x[10], S43, 0xffeff47d); /* 55 */
II (b, c, d, a, x[ 1], S44, 0x85845dd1); /* 56 */
II (a, b, c, d, x[ 8], S41, 0x6fa87e4f); /* 57 */
II (d, a, b, c, x[15], S42, 0xfe2ce6e0); /* 58 */
II (c, d, a, b, x[ 6], S43, 0xa3014314); /* 59 */
II (b, c, d, a, x[13], S44, 0x4e0811a1); /* 60 */
II (a, b, c, d, x[ 4], S41, 0xf7537e82); /* 61 */
II (d, a, b, c, x[11], S42, 0xbd3af235); /* 62 */
II (c, d, a, b, x[ 2], S43, 0x2ad7d2bb); /* 63 */
II (b, c, d, a, x[ 9], S44, 0xeb86d391); /* 64 */

state[0] += a;
state[1] += b;
state[2] += c;
state[3] += d;

// Zeroize sensitive information.
memset(x, 0, sizeof x);
}

////////////////////////////////////

// MD5 block update operation. Continues an MD5 message-digest
// operation, processing another message block
void MD5::update(const unsigned char input[], size_type length)
{
    // compute number of bytes mod 64

```

```

    size_type index = count[0] / 8 % blocksize;

    // Update number of bits
    if ((count[0] += (length << 3)) < (length << 3))
        count[1]++;
    count[1] += (length >> 29);

    // number of bytes we need to fill in buffer
    size_type firstpart = 64 - index;

    size_type i;

    // transform as many times as possible.
    if (length >= firstpart)
    {
        // fill buffer first, transform
        memcpy(&buffer[index], input, firstpart);
        transform(buffer);

        // transform chunks of blocksize (64 bytes)
        for (i = firstpart; i + blocksize <= length; i += blocksize)
            transform(&input[i]);

        index = 0;
    }
    else
        i = 0;

    // buffer remaining input
    memcpy(&buffer[index], &input[i], length-i);
}

////////////////////////////////////

// for convenience provide a version with signed char
void MD5::update(const char input[], size_type length)
{
    update((const unsigned char*)input, length);
}

////////////////////////////////////

// MD5 finalization. Ends an MD5 message-digest operation, writing the
// the message digest and zeroizing the context.

```

```

MD5& MD5::finalize()
{
    static unsigned char padding[64] = {
        0x80, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
    };

    if (!finalized) {
        // Save number of bits
        unsigned char bits[8];
        encode(bits, count, 8);

        // pad out to 56 mod 64.
        size_type index = count[0] / 8 % 64;
        size_type padLen = (index < 56) ? (56 - index) : (120 - index);
        update(padding, padLen);

        // Append length (before padding)
        update(bits, 8);

        // Store state in digest
        encode(digest, state, 16);

        // Zeroize sensitive information.
        memset(buffer, 0, sizeof buffer);
        memset(count, 0, sizeof count);

        finalized=true;
    }

    return *this;
}

////////////////////////////////////

// return hex representation of digest as string
std::string MD5::hexdigest() const
{
    if (!finalized)
        return "";

    char buf[33];
    for (int i=0; i<16; i++)

```

```

        sprintf(buf+i*2, "%02x", digest[i]);
        buf[32]=0;

        return std::string(buf);
    }

    //////////////////////////////////////

    std::ostream& operator<<(std::ostream& out, MD5 md5)
    {
        return out << md5.hexdigest();
    }

    //////////////////////////////////////

    std::string md5(const std::string str)
    {
        MD5 md5 = MD5(str);

        return md5.hexdigest();
    }

```

附录 2：评分表

课程名称： 计算机程序设计基础 2

项 目	评 价	
设计方案的合理性与创新性	3×2	
设计与调试结果	4×2	
设计说明书的质量	1×2	
程序基本要求涵盖情况	4×2	
程序代码编写素养情况	2×2	
课程设计周表现情况	1×2	
综合成绩	15×2	

教师签名：

日 期：