

---

# 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 <b>1</b> 周			
小学期 第 <b>2</b> 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。

---

# 学生成绩管理系统

## 目录

学生成绩管理系统.....	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
成绩管理系统模块图.....	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
stu.h .....	13
main.cpp .....	27
md5.h(来自互联网).....	31
md5.cpp(来自互联网).....	32
附录 2: 评分表 .....	41

---

## 1. 系统需求分析

学生成绩管理系统记录了学生的成绩信息，记录了每名学生的各科成绩以及 GPA。设计成绩管理系统，应该具有以下特性

1. 在用户上应该进行分权限管理，整个系统具有两套界面——只有管理员有权进行成绩录入并查看所有人的成绩；而学生只有查询本人成绩的权力。整个认证系统使用密码进行加密。
2. 管理员可以随意增减课程。
3. 管理员可以增加学生。
4. 管理员可以查看学生成绩情况，按照 GPA 进行排列。
5. 学生可以查询本人的成绩和 GPA。
6. 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。
7. 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

学生成绩管理系统包含两个模块，分别是学生模块和管理员模块。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。

### 管理员模块

使用编号 0 登陆。初始密码为 123456。管理员模块包括 9 个功能，分别是：

- 1) 显示成绩排名。按照 GPA 高低依次显示所有学生的姓名、学号以及各科成绩
- 2) 查询平均 GPA。
- 3) 显示所有课程。
- 4) 显示所有学生，按学号顺序排列。
- 5) 查询某学生信息。以某学生的身份登陆系统。

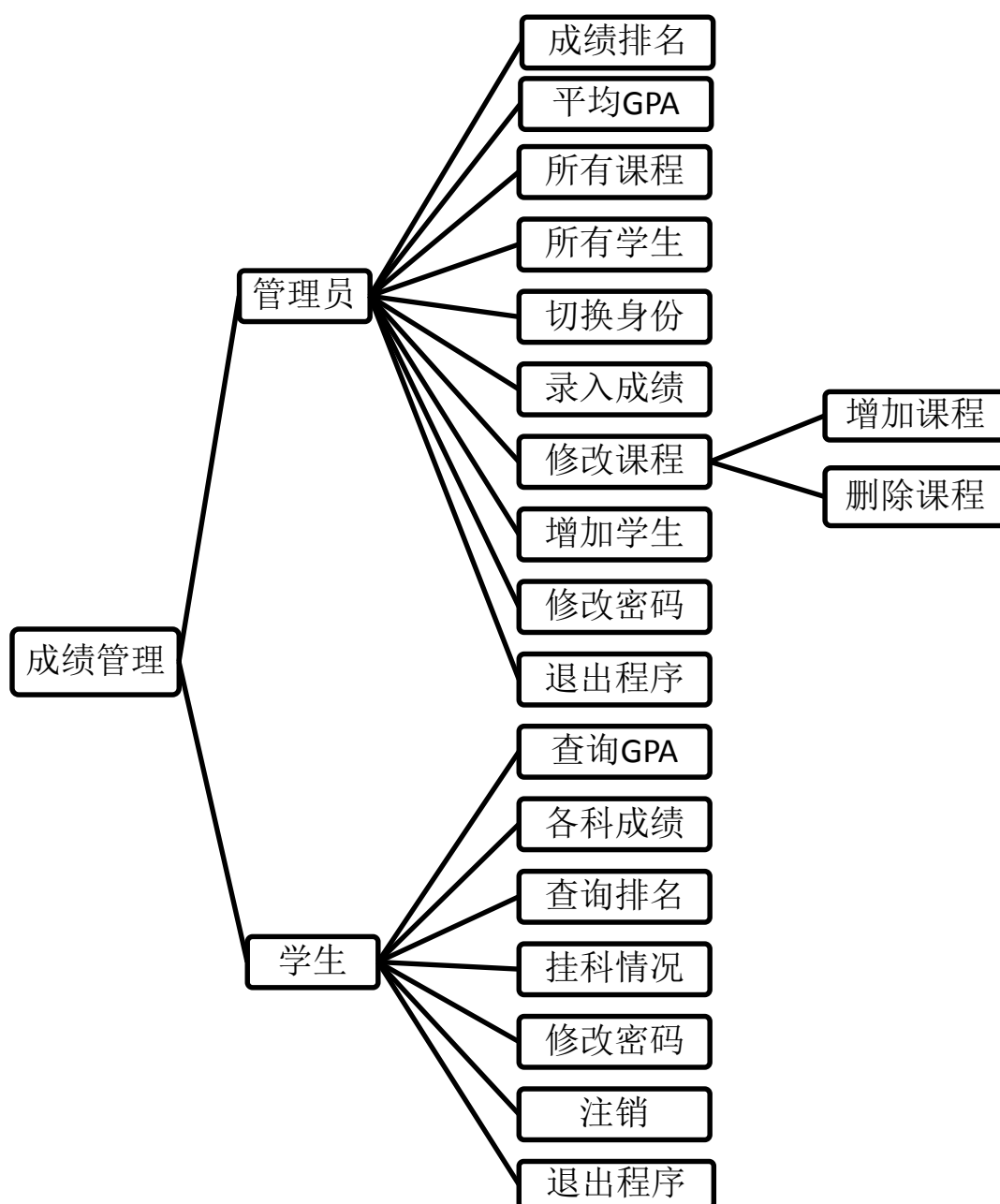
- 
- 6) 录入成绩。选择相应课程（选择记通过或记成绩），按顺序依次录入各学生的成绩。
  - 7) 修改课程信息。可以增加或删除课程
  - 8) 修改学生信息。可以增加学生
  - 9) 修改密码。修改管理员密码
  - 0) 退出程序。

## 学生模块

使用学号登陆后自动跳转至该页面。

- 1) 查询本人 GPA
- 2) 查询各科成绩
- 3) 查询本人 GPA 排名
- 4) 查询本人挂科情况。可以看到挂科科目数和学分。
- 5) 修改密码
- 6) 注销。退回登陆界面
- 7) 退出

## 成绩管理系统模块图



图一

### 3. 详细设计

由于学生与课程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，形成类似于二维数组的结构。但每个同学采用一样的课程列表。



---

本程序 UML 图如图二。

本程序使用 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、Course、Clist 中，而 ALL 则通过继承和直接声明的方式进行二次封装，通过该类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，将 ALL 的对象 all 中的数据写入文件即可。

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，使用了公布在互联网上的部分代码，对程序中的密码进行了 md5 加密。

成绩的传输写成函数，需要传递课程号和成绩两个信息，但在传递时使用数组会出现很麻烦的情况，采用了 课程号\*101+成绩 或 课程号\*10+通过情况（优秀 3 良好 2 及格 1 不及格 0）的方式，使用 unsigned char 型数据（会导致最多只有 25 门课，但够了）。

菜单按照成绩管理系统模块图，采用单级菜单即可。

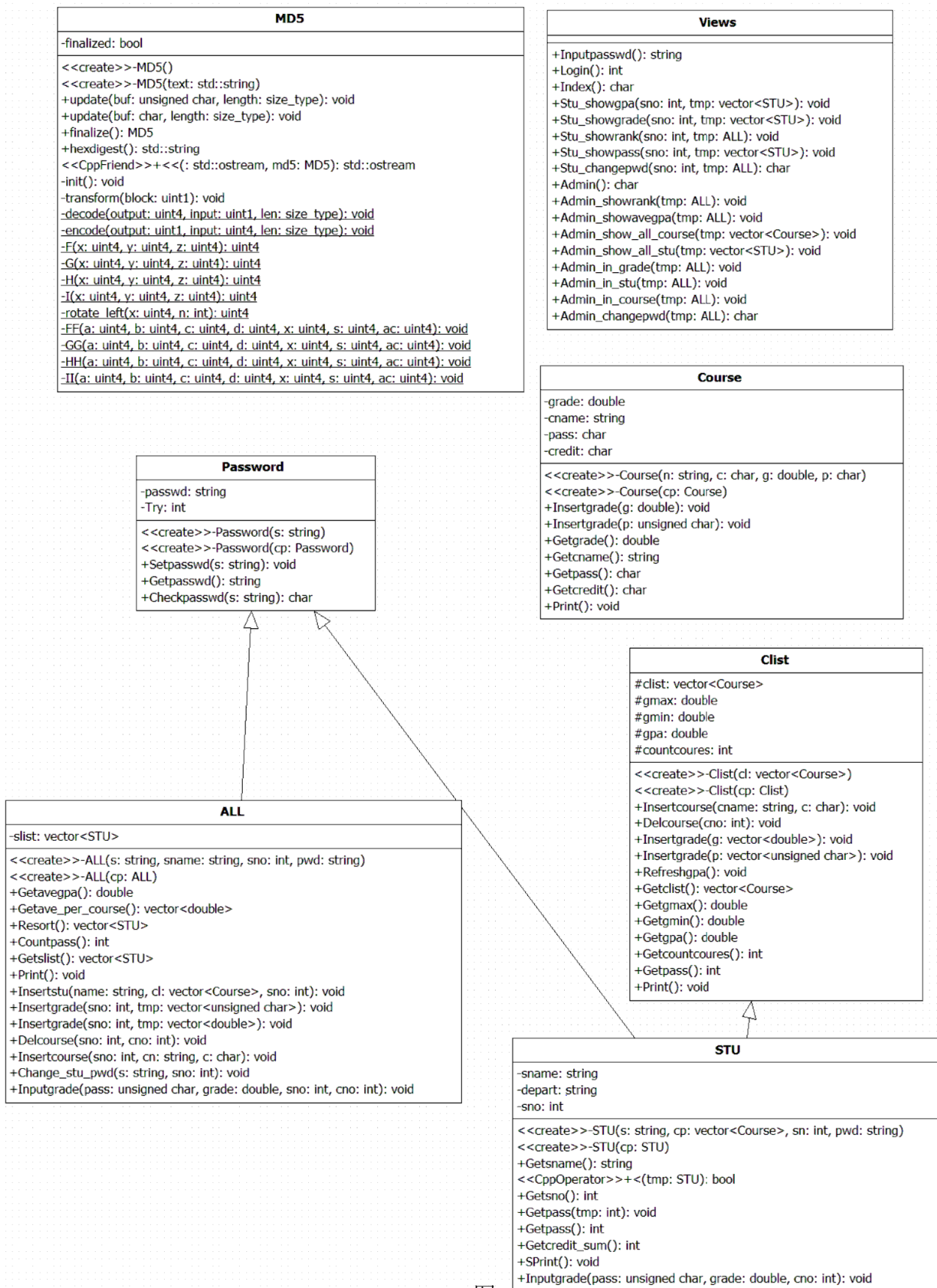


图 2

---

## 4. 系统调试

调试过程中因为输入错误而导致的问题不必多提。

调试过程中文件读写出现了很大的问题，原因在于，类 ALL 中存储的数据是以 `vector` 的形式存储的，将其对象写入文件并不会真正写入 `vector` 所存储的值，再次读入只会得到无用的指针，只能遍历所有的数据逐个输出：

另一个问题是 `STU` 类中每一个学生都有一个学号，最开始将学号直接设定为学生在 `vector slist` 中的位置，但考虑到学号有一定的独立性，所以将其设为了独立变量，由此不得不对程序进行大改，出现了很多次 `vector` 越界的情况；

还有就是为了节省内存空间，`STU` 的 `pass` 设置成了 `unsigned char` 型，在之后的 `io` 中因此出现了很大的问题，尤其是从文件中读入时会出现很多的‘1’，‘2’再这样的情况，不得不进行手工转换；

除此之外，登陆的问题也很麻烦，因为 `sno` 和实际的 `stu` 存储位置没有关系，所以不得不在 `main` 函数中增加了对 `slist` 的遍历来确定其实际位置进行验证；

权限问题也很麻烦，ALL 对其中的数据缺乏足够的权限，因此在 `Views` 中调用时不得不使用一层层的 `Get*()` 类函数，但是这些 `Get` 函数只能用于展示数据，而无法进行修改，因此又不得不加入一层层的 `Change` 函数，导致了代码量的膨胀。

密码验证时，使用了 `conio.h` 头文件，但是前几次调试中，始终无法达到隐藏输入密码的功能，后来才发现，把 `getch` 当成了 `getchar`。

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。

登录时，同一个账号密码输错三次就会自动退出。

本程序的管理员账户用户账号为 0，默认无密码。

学生使用学号（无长度限制）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。

---

未初始化时，仅有一个名为 `nh`，学号为 `2013011280`，无密码的学生账号。  
未初始化时没有课程。

测试数据已经录入，但通过删除 `stu.dat` 可以初始化程序，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，管理员可以增添课程（需要学分），删除课程（会丢失数据）和增加学生、录入成绩（按课程编号）。请严格按照学号 姓名的格式输入，否则可能导致程序崩溃。

查询某学生信息可以以某学生的身份登陆。

需要注意的是，由于程序设计所限，最多只能有 `25` 门课（只有前 `25` 门课可以设置为记通过，之后的课程可以为录入成绩），而学生数量取决于所运行的计算机位数以及内存大小，在大多数情况下是够用的。

学生：

学生只有查询的权利，考虑到隐私问题，只能看到本人的成绩。

## 6. 编程体会

由于之前经验的影响，试图以 `mvc` 这样的模块化编程来完成本次大作业，这样的思想确实化简了整个编程过程的思维负担，也在一定程度上简化了编程，这是本次编程的亮点之一。

其次，为了实现密码模块，我使用了 `md5` 这样的函数进行加密，同时使用了 `conio.h`（但这样导致了兼容性的问题，只能运行在 `win` 下）来实现密码的输入。这是一个亮点。

最后，`stu` 和 `course` 都是通过 `vector` 容器进行存储，类似一个动态的二维数组，使我对容器有了更进一步的了。这也是一个亮点。

但是，由于文件读写规划上的失误，导致不得不在 `main` 函数中添加了大量

---

的代码来完成数据读写，而 `model` 部分设计的缺陷使类的权限问题相当麻烦，最后，不得不增添了很多函数来完成这些操作。最终的程序有一些偏离 `mvc` 结构，这是一个遗憾。

还有一个遗憾，那便是由于时间仓促，在最终版本中，存储在文件中的成绩与学生信息是没有加密的，这样真的十分危险。同时，对于增添学生和课程的输入检验也没有做，这也很危险。也没有设计成每个同学可以有不同的课程，虽然数据结构允许，但整个操作逻辑会非常麻烦，所以没有做。

在最初的几次编译中，出现了大量的语法错误，最终消耗了不少精力，但这一方面警示我在编程时要更加集中注意力，同时也说明，`debug` 时需要耐心，一个一个问题的解决，再难的问题也能解决。

## 附录 1：源程序

### `stu.h`

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

class Course
{
public:
    Course(string n="", char c=0, double g=-1, char p=-1) {
        cname=n;
        credit=c;
        grade=g;
        pass=p;
    }
    Course(const Course
&cp):grade(cp.Getgrade()), cname(cp.Getcname()), pass(cp.Getpass()), cre
```

---

```

dit(cp.Getcredit()) {}
    void Insertgrade(double g);
    void Insertgrade(unsigned char p) {pass=p;}
    double Getgrade() const {return grade;}
    string Getcname() const {return cname;}
    char Getpass() const {return pass;}
    char Getcredit() const {return credit;}
    void Print() {
        cout<<cname<<":\t";
        if(grade!=-1) cout<<grade;
        if(pass==1)
            cout<<"\t通过\t";
        if(pass==2)
            cout<<"\t良好\t";
        if(pass==3)
            cout<<"\t优秀\t";
        if(pass==0)
            cout<<"\t不及格\t";
        if(pass==-1)
            cout<<"\t无信息\t";
    }
private:
    double grade;
    string cname;
    char pass;//1表示通过, 2良好, 3优秀
    char credit;
};

```

```

void Course::Insertgrade(double g) {
    grade=g;
    if(g>=80) {
        if(g>=90) pass=3;
        else pass=2;
    }
    else{
        if(g>=60) pass=1;
        else pass=0;
    }
}

```

```

class Clist
{
public:
    Clist(vector<Course> c1):clist(c1) {

```

---

```

        gpa=0;
        gmin=100;
        gmax=0;
        countcours=0;
    }
    Clist(const Clist
&cp):clist(cp.Getclist()), gmax(cp.Getgmax()), gmin(cp.Getgmin()), gpa(c
p.Getgpa()), countcours(cp.Getcountcours()) {}
    void Insertcourse(string cname, char c) {
        Course tmp(cname, c, (double)-1, (char)-1);
        clist.push_back(tmp);
        countcours++;
    }
    void Delcourse(int cno) {
        clist.erase(clist.begin()+cno);
    }
    void Insertgrade(vector<double> g)
    {
        for(int i=0;i<g.size();i++){

            clist[(int)g[i]/101].Insertgrade(g[i]-101*((int)g[i]/101));
            if((g[i]-(int)g[i]/101)<gmin)
gmin=g[i]-101*((int)g[i]/101);
            if((g[i]-(int)g[i]/101)>gmax)
gmax=g[i]-101*((int)g[i]/101);
            //将成绩传输时写成 课程号*101+成绩的形式，方便传输
        }
        Refreshgpa();
    }
    void Insertgrade(vector<unsigned char> p)
    {
        for(int i=0;i<p.size();i++){
            clist[(int)p[i]/10].Insertgrade((unsigned
char)(p[i]%10)); //副作用是只能有25门课程
        }
    }
    void Refreshgpa();
    vector<Course> Getclist() const{return clist;}
    double Getgmax() const{return gmax;}
    double Getgmin() const{return gmin;}
    double Getgpa() const{return gpa;}
    int Getcountcours() const{return countcours;}
    int Getpass();
    void Print();

```

---

```

protected:
    vector<Course> clist;
    double gmax;
    double gmin;
    double gpa;
    int countcours;
};

void Clist::Refreshgpa() {
    int credit=0;
    gpa=0;
    for(int i=0;i<clist.size();i++) {
        if(clist[i].Getgrade()==-1) continue;//-1表示记通过
        gpa+=clist[i].Getgrade()*clist[i].Getcredit();
        credit+=clist[i].Getcredit();
    }
    gpa/=credit;
}

void Clist::Print() {
    for(int i=0;i<clist.size();i++) {
        clist[i].Print();
        if((i%2)==1) cout<<endl;
        else cout<<"\t";
    }
}

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);
        }
    }
}

```



---

```

    }
private:
    string passwd;
    int Try;//限制尝试次数

};

class STU:public Clist,public Password{
public:
    STU(const string s,vector<Course>& cp,int sn,string
pwd=md5("")):sname(s),sno(sn),Clist(cp),Password(pwd){}
    STU(const STU&
cp):sname(cp.Getcname()),Clist(cp),Password(cp),sno(cp.sno){}
    string Getcname() const {return sname;}
    bool operator<(const STU& tmp){return (gpa<tmp.gpa);}
    int Getsno(){return sno;}
    void Getpass(int* tmp);//该数组第一个值是挂科数，第二个是挂科学分
    int Getpass();
    int Getcredit_sum();
    void SPrint(){cout<<sno<<". "<<sname<<"\tGPA:"<<gpa<<endl;
Print();}
    void Inputgrade(unsigned char pass,double grade, int cno){
        clist[cno].Insertgrade(grade);
        clist[cno].Insertgrade(pass);
        if(grade!=-1){
            if(grade>gmax) gmax=grade;
            if(grade<gmin) gmin=grade;
        }
        Refreshgpa();
    }
private:
    string sname;
    string depart;
    int sno;
};

void STU::Getpass(int* tmp){
    for(int i=0;i<clist.size();i++){
        if(clist[i].Getpass()==0){
            tmp[0]++;
            tmp[1]+=clist[i].Getcredit();
        }
    }
}
}

```

---

```

int STU::Getpass()
{
    int tmp=0;
    for(int i=0;i<clist.size();i++){
        if(clist[i].Getpass()==0){
            tmp++;
        }
    }
    return tmp;
}

int STU::Getcredit_sum() {
    int sum=0;
    for(int i=0;i<clist.size();i++) sum+=clist[i].Getcredit();
    return sum;
}

class ALL:public Password{
public:
    ALL(string s=md5(""),string sname="nh",int sno=1,string
pwd=md5("")):Password(s){
        vector<Course> clist;
        STU t(sname,clist,sno,pwd);
        slist.push_back (t);
    }
    ALL(const ALL& cp):Password(cp),slist(cp.slist){}
    double Getavegpa();
    vector<double> Getave_per_course();
    vector<STU> Resort();
    int Countpass();
    vector<STU> Getslist() const { return slist;}
    void Print();
    void Insertstu(string& name,vector<Course> cl,int sno){
        STU tmp(name,cl,sno);
        slist.push_back(tmp);
    }
    void Insertgrade(int sno,vector<unsigned char> tmp){
        slist[sno].Insertgrade(tmp);
    }
    void Insertgrade(int sno,vector<double> tmp){
        slist[sno].Insertgrade(tmp);
    }
    void Delcourse(int sno,int cno){
        slist[sno].Delcourse(cno);
    }
}

```

---

```

    void Insertcourse(int sno, string cn, char c) {
        slist[sno].Insertcourse(cn, c);
    }
    void Change_stu_pwd(string s, int sno) {
        slist[sno].Setpasswd(s);
    }
    void Inputgrade(unsigned char pass, double grade, int sno, int cno) {
        slist[sno].Inputgrade(pass, grade, cno);
    }
private:
    vector<STU> slist;
};
double ALL::Getavegpa() {
    double sum=0;
    for(int i=0;i<slist.size();i++)
        sum+=slist[i].Getgpa();
    return sum/slist.size();
}
vector<double> ALL::Getave_per_course() {
    double sum=0;
    vector<double> c;
    for(int j=0;j<(slist[0].Getclist()).size();j++) {
        if((slist[0].Getclist())[j].Getgrade()==-1) continue;
        for(int i=0;i<slist.size();i++) {
            sum+=(slist[i].Getclist())[j].Getgrade();
        }
        c.push_back(sum/slist.size());
        sum=0;
    }
    return c;
}
vector<STU> ALL::Resort() {
    vector<STU> tslist=slist;
    for(int i=1;i<tslist.size();i++) {
        for(int j=0;j<(tslist.size()-i);j++) {
            if(tslist[j]<tslist[j+1])
                swap(tslist[j], tslist[j+1]);
        }
    }
    return tslist;
}
int ALL::Countpass() {
    int count=0;
    for(int i=1;i<slist.size();i++) count+=slist[i].Getpass();
}

```

---

```

        return count;
    }
    void ALL::Print() {
        for(int i=0;i<slist.size();i++) {
            slist[i].SPrint();
            cout<<endl;
        }
    }
}

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 select=42;
    system("cls");
    //for win system("cls");

    cout<<"_____ \n
\n"
        <<"\t\t欢迎使用学生信息管理系统\n\n"

        <<"_____ \n\n\n
\n"
        <<"1. 查询本人GPA\n\n"
        <<"2. 查询各科成绩\n\n"
        <<"3. 查询本人GPA排名\n\n"
        <<"4. 查询本人挂科情况\n\n"
        <<"5. 修改密码\n\n"
        <<"6. 注销\n\n"
        <<"7. 退出\n\n\n\n"
        <<"请输入需要的操作: ";
    cin>>select;
    return select;
}

void Stu_showgpa(int sno, vector<STU> tmp) {
    system("cls");
    cout<<tmp[sno].Getsname()<<"同学, \n你的GPA为
"<<tmp[sno].Getgpa();
    cout<<"\n\n\n\n输入任意键返回";
    getchar();getchar();
}

void Stu_showgrade(int sno, vector<STU> tmp) {
    system("cls");
    tmp[sno].SPrint();
    cout<<"\n\n\n\n输入任意键返回";

```

---

```

        getchar();getchar();
    }
    void Stu_showrank(int sno, ALL& tmp) {
        system("cls");
        vector<STU> rank=tmp.Resort();
        for(int i=0;i<rank.size();i++)
        {
            if (rank[i].Getsno()==sno) {
                cout<<rank[i].Getsname()<<"，你的排名为:\t"<<i+1<<endl;
                break;
            }
        }
        cout<<"\n\n\n\n输入任意键返回";
        getchar();getchar();
    }
    void Stu_showpass(int sno, vector<STU> tmp) {
        system("cls");
        int t[2]={0,0};
        tmp[sno].Getpass(t);
        if(t[0]!=0) cout<<tmp[sno].Getsname()<<"\n\n\n\n同学，你的挂科
数为:"<<t[0]<<"\t挂科学分为:"<<t[1]<<endl;
        else cout<<tmp[sno].Getsname()<<"\n\n\n\n同学，恭喜你沒有挂科
"<<endl;
        cout<<"\n\n\n\n输入任意键返回";
        getchar();getchar();
    }
    char Stu_changepwd(int sno, ALL& tmp) {
        system("cls");
        cout<<"\n请输入旧密码:";
        string s=Inputpasswd(),s1;
        char judge=tmp.Getslis() [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;
}

//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欢迎使用学生信息管理系统\n"

    <<"_____ \n\n"
        <<"\t\t\t\t\t查询\n\n\n"
        <<"1. 显示成绩排名\n"
        <<"2. 查询平均GPA\n"
        <<"3. 显示所有课程\n"
        <<"4. 显示所有学生\n"
        <<"5. 查询某学生信息\n\n"
        <<"\t\t\t\t\t修改\n\n"
        <<"6. 录入成绩\n"
        <<"7. 修改课程信息\n"
        <<"8. 修改学生信息\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_showrank(ALL& tmp) {
    system("cls");
    vector<STU> rank=tmp.Resort();
    for(int i=0;i<rank.size();i++)
    {
        rank[i].SPrint();
        cout<<endl;
    }
    cout<<"\n\n\n\n输入任意键返回";

```

---

```

        getchar();getchar();
    }
    void Admin_showavegpa(ALL& tmp) {
        system("cls");
        cout<<"平均GPA为:"<<tmp.Getavegpa();
        cout<<"\n\n\n\n输入任意键返回";
        getchar();getchar();
    }
    void Admin_show_all_course(vector<Course> tmp) {
        system("cls");
        for(int i=0;i<tmp.size();i++) {
            cout<<i+1<<". "<<tmp[i].Getcname();
            if(i%5==4) cout<<endl;
            else cout<<"\t";
        }
        cout<<"\n\n\n\n输入任意键返回";
        getchar();getchar();
    }
    void Admin_show_all_stu(vector<STU> tmp) {
        system("cls");
        for(int i=0;i<tmp.size();i++) {
            cout<<tmp[i].Getsno()<<". "<<tmp[i].Getsname();
            if(i%3==2) cout<<endl;
            else cout<<"      ";
        }
        cout<<"\n\n\n\n输入任意键返回";
        getchar();getchar();
    }
    void Admin_in_grade(ALL & tmp);
    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.Insertstu(sn, (tmp.Getslst())[0].Getclst(), sno);
            cout<<"是否继续[y/n]";
            cin>>select;
        }
    }
    void Admin_in_course(ALL & tmp);
    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;
    }
};

void Views::Admin_in_grade(ALL & tmp) {
    Admin_show_all_course(tmp.Getslislist()[0].Getclist());
    char select1=42,select2=42;
    cout<<"请选择要录入的课程:"<<endl;
    scanf("%d",&select1);
    select1-=1;
    cout<<"该课程是否为记通过? [y/n]";
    cin>>select2;
    if(select2=='y' &&select1<25) {
        vector<unsigned char> grade;
        unsigned char g=0;
        for(int i=0;i<tmp.Getslislist().size();i++) {

            cout<<tmp.Getslislist()[i].Getsno()<<". ";<<tmp.Getslislist()[i].Getsname
            ()<<":[不及格 0/及格 1/良好 2/优秀 3]";
            cin>>g;
            if(g=='1' || g=='2' || g=='3' || g=='0') {
                g=g-'0';
                grade.push_back(10*select1+g);
                tmp.Insertgrade(i, grade);
            }
        }
    }
}

```

---

```

        grade.pop_back();
    }
    else{
        cout<<"错误的输入"<<endl;
        i--;
    }
}

}
else{
    vector<double> grade;
    double g=0;
    for(int i=0;i<tmp.Getslis().size();i++){

        cout<<tmp.Getslis()[i].Getsno()<<". "<<tmp.Getslis()[i].Getsname
()<<": ";
        cin>>g;
        if(g>=0&&g<=100){
            grade.push_back(select1*101+g);
            tmp.Insertgrade(i, grade);
            grade.pop_back();
        }
        else{
            cout<<"错误的输入"<<endl;
            i--;
        }
    }
}

}

void Views::Admin_in_course(ALL & tmp){
    system("cls");
    char select,select2='y';
    cout<<"1. 增加课程"
    <<"\n\n2. 删除课程"
    <<"\n\n请输入需要的操作:";
    cin>>select;
    cout<<select;
    if(select=='1'){
        system("cls");
        string cn;
        char c;
        for(;select2=='y;){
            cout<<"请输入课程名与学分: ";
            cin>>cn>>c;

```

---

```

        for(int i=0;i<tmp.Getslis() .size();i++){
            tmp.Insertcourse(i,cn,c-'0');
        }
        cout<<"\n是否继续:[y/n]";
        cin>>select2;
    }
}
else {
    if(select=='2'){
        for(;select2=='y'){
            Admin_show_all_course(tmp.Getslis()[0].Getclis());
            cout<<"\n请选择需要删除的课程:";
            int cno=0;
            cin>>cno;
            for(int i=0;i<tmp.Getslis().size();i++){
                tmp.Delcourse(i,cno-1);
            }
            cout<<"\n是否继续:[y/n]";
            cin>>select;
        }
    }
    else cout<<"错误的输入"<<endl;
}
}

```

## main.cpp

```

#include<iostream>
#include<string>
#include<vector>
#include<fstream>
#include "md5.h"
#include "stu.h"
class ALL;
int main(void){
    string s=md5(""),sname="nh",pwd=md5("");
    int sno=2013011280;
    fstream ifile("stu.dat",ios_base::in);
    if(ifile) {
        ifile>>s>>sno>>sname>>pwd;
    }
    ALL all(s,sname,sno,pwd);
}

```

---

```

if(ifile){
    //导入学生
    ifile>>sno;
    while(sno!=-1){
        ifile>>sname>>pwd;
        all.Insertstu(sname, (all.Getslislist())[0].Getclislist(), sno);
        all.Change_stu_pwd(pwd, all.Getslislist().size()-1);
        ifile>>sno;
    }
    //导入课程
    int credit=0;
    string cname;
    ifile>>credit;
    while(credit!=-1){
        ifile>>cname;
        for(int i=0;i<all.Getslislist().size();i++){
            all.Insertcourse(i, cname, (char)credit);
        }
        ifile>>credit;
    }
    //导入成绩
    int j=0, k=0;
    double grade;
    ifile>>j;
    while(j!=-1){
        ifile>>k;
        for(int
n=0; (n<all.Getslislist()[0].Getclislist().size())&&(k!=-3); n++){
            ifile>>grade;
            all.Inputgrade((unsigned char)k, grade, j, n);
            ifile>>k;
        }
        ifile>>j;
    }
}

```

```

Views view;
int select=42;
//登录模块
int user_id=42, i=0;
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{
Stu:
        for(i=0;(i<all.Getslislist().size())&&(all.Getslislist()[i].Getsno()!=u
ser_id);i++){
            if(i!=all.Getslislist().size()){
                cout<<"请输入密码:";
                string s=view.Inputpasswd();
                check=all.Getslislist()[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();
            switch(select){
                case 1: view.Stu_showgpa(i,all.Getslislist()); break;
                case 2: view.Stu_showgrade(i,all.Getslislist());break;
                case 3: view.Stu_showrank(user_id,all);break;
                case 4: view.Stu_showpass(i,all.Getslislist());break;
                case 5: view.Stu_changepwd(i,all);break;
                case 6: goto Begin; break;
                case 7: 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: view.Admin_showrank(all);break;
                case 2: view.Admin_showavegpa(all);break;
                case 3:
view.Admin_show_all_course(all.Getslis() [0].Getclis());break;
                case 4: view.Admin_show_all_stu(all.Getslis());break;
                case 5: view.Admin_show_all_stu(all.Getslis());
                    cout<<"请选择相应的学生:";
                    cin>>user_id;
                    goto Stu;
                    break;
                case 6: view.Admin_in_grade(all);break;
                case 7: view.Admin_in_course(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
    vector<STU> stmp=all.Getslis();
    fstream of("stu.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;
        for(int k=0;k<stmp[0].Getclis().size();k++)
            of<<(int) (stmp[0].Getclis()[k].Getcredit())<<"
"<<stmp[0].Getclis()[k].Getcname()<<" ";
            of<<-1<<endl;
            for(int j=0;j<stmp.size();j++){
                of<<j<<" ";
                for(int k=0;k<stmp[j].Getclis().size();k++){
                    of<<(int) stmp[j].Getclis()[k].Getpass()<<"
"<<stmp[j].Getclis()[k].Getgrade()<<" ";
                }
                of<<-3<<endl;
            }
        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, 0xc1bdcee5); /\* 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, 0xfffa3942); /* 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： \_\_\_\_