实验一　验证码生成及其应用（验证）

实验说明：

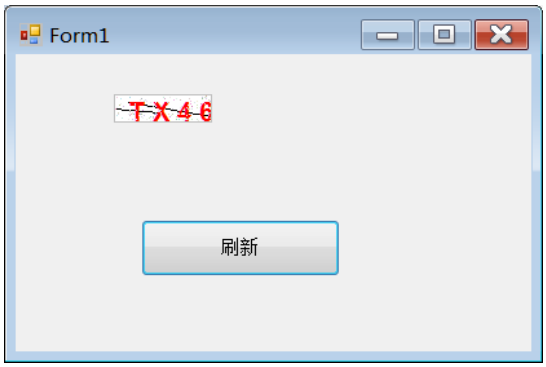
(一)　验证码可以用来防止用户利用机器人自动注册、登录、灌水等暴力攻击，防止大规模匿名回帖等，使用验证码技术也是现在很多网站通行的方式。本实验以教材 P86 为参考资料，但本实验以 C#语言实现，需要安装 Microsoft Visual Studio 2012（版本不限，建议 2008 以上版本）。

(二)　本大实验包括三个小实验（目的为了提高同学们的动手能力）

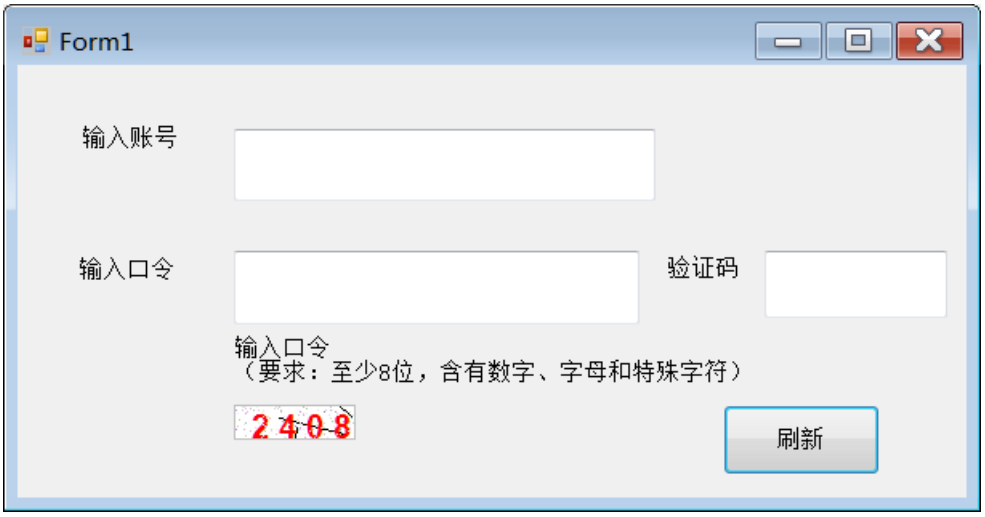
（1） 教材 P86，实验 5A 的第一个实验（1），是验证密码强度程序。同学们在做实验时，可以先理解程序，然后将该程序修改为一个实用版的密码强度检验程序，要求：输入的密码长度至少是 8 位，而且输入的密码中含有数字、字母和特殊符号才是合乎要求的密码（该实验可作为课外练习）

（2） 教材 P86，实验 5A 的第二个实验（2），是一个图片式的验证码程序，但教材上是用 VB 实现的。但我们做实验时，用 C#实现。改编后的代码附在本实验后。要求：同学们理解和掌握该程序是设计思想（或方法）。（**该实验是必做实验**）

运行后的结果为下图：



（3） 实验三，也是一个综合性的实验，将以上的实验内容结合起来，实现一个完整的、具有实用性的登录界面（或登录功能）。即运行后如下图：
（该实验是必做实验，该实验是提高实战能力）



(三) 实验报告

提交实验报告，实验报告要求：

（1）写出你的设计思想或方法

（2）写出实验过程中你遇到的问题， 你的收益和体会。

■ 附部分代码：

说明：下列的验证码分三种情况：

1) 随机生成四位（或几个）数字（这种验证码已不安全，不提倡使用）；

2) 随机生成几个字字符（这种验证码）；

3）随之生成几个数字加字符的混合式的验证码；

4)随之生成几个数字加字符的混合式的并有背景图案的(增加噪声干扰）验证码（目前建议使用这一种）。

1. **随机生成四位（或几个）数字的验证码（这种验证码已不安全，不提倡使用），代码如下：**

```
private void button1_Click(object sender, EventArgs e)
        {
            int seekSeek = unchecked((int)DateTime.Now.Ticks);
            Random rnd = new Random(seekSeek); //随机函数
            textBox1.Text = rnd.Next(20).ToString();
    }
```

或者：

```
//生成 1 到 9 之间的 4 个随机数
        private void button1_Click(object sender, EventArgs e)
 {
            Random r = new Random();
            string str = "";
            for (int i = 0; i < 4; i++)
            {
                int rNumber = r.Next(0, 10);
                //累加到空字符串中
                str += rNumber;
```

```
        }
    }
```

2. 随机生成几个字字符（这种验证码）：

```
    private void button1_Click(object sender, EventArgs e)
        {
            string chkCode="";
            //验证码的字符集，去掉了一些容易混淆的字符
            char[] character ={'A', 'B', 'C', 'D', 'E','F', 'G', 'H', 'I','J', 'K',
'L', 'M', 'N','O',  'P','Q', 'R', 'S', 'T', 'W', 'X', 'Y' ,'Z'};
             Random rnd = new Random();
             //生成验证码字符串
            for (int i = 0; i < 4; i++)
            {
                chkCode += character[rnd.Next(character.Length)];
            }
            textBox1.Text = chkCode;
        }
```

3. 随之生成几个数字加字符的混合式的验证码:

```
    private void button1_Click(object sender, EventArgs e)
        {
            string chkCode="";
            //验证码的字符集，去掉了一些容易混淆的字符
            char[] character ={ '2', '3', '4', '5', '6', '8', '9', 'A', 'B', 'C', 'D',
'E','F', 'G', 'H', 'J', 'K', 'L', 'M', 'N', 'P', 'R', 'S', 'T', 'W', 'X', 'Y' };
             Random rnd = new Random();
             //生成验证码字符串
            for (int i = 0; i < 4; i++)
            {
                chkCode += character[rnd.Next(character.Length)];
            }
            textBox1.Text = chkCode;
        }
```

4. 生成几个数字加字符的混合式的并有背景图案的（增加噪声干扰）验证码（目前建议使用这一种）,参考代码如下：

■ 图形验证码：（C#版）

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
```

```
using System.Windows.Forms;
namespace DrawValidateCode
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            CodeImage(CheckCode());
        }

        private string CheckCode()          //此方法生成
        {
            int number;
            char code;
            string checkCode = String.Empty;  //声明变量存储随机生成的 4 位英文或数字
            Random random = new Random();             //生成随机数
            for (int i = 0; i < 4; i++)
            {
                number = random.Next();                 //返回非负随机数
                if (number % 2 == 0)                    //判断数字是否为偶数
                    code = (char)('0' + (char)(number % 10));
                else                                            //如果不是偶数
                    code = (char)('A' + (char)(number % 26));
                checkCode += " " + code.ToString();         //累加字符串
            }
            return checkCode;                                   //返回生成的字符串
        }

        private void CodeImage(string checkCode)
        {
            if (checkCode == null || checkCode.Trim() == String.Empty)
                return;
            System.Drawing.Bitmap image = new
System.Drawing.Bitmap((int)Math.Ceiling((checkCode.Length * 9.5)), 22);
            Graphics g = Graphics.FromImage(image);         //创建 Graphics 对象
            try
            {
                Random random = new Random();       //生成随机生成器
                g.Clear(Color.White);                       //清空图片背景色
```

```
                for (int i = 0; i < 3; i++)                    //画图片的背景噪音线
                {
                    int x1 = random.Next(image.Width);
                    int x2 = random.Next(image.Width);
                    int y1 = random.Next(image.Height);
                    int y2 = random.Next(image.Height);
                    g.DrawLine(new Pen(Color.Black), x1, y1, x2, y2);
                }
                Font font = new System.Drawing.Font("Arial", 12,
(System.Drawing.FontStyle.Bold));
                g.DrawString(checkCode, font, new SolidBrush(Color.Red), 2, 2);
                for (int i = 0; i < 150; i++)          //画图片的前景噪音点
                {
                    int x = random.Next(image.Width);
                    int y = random.Next(image.Height);
                    image.SetPixel(x, y, Color.FromArgb(random.Next()));
                }
                //画图片的边框线
            g.DrawRectangle(new Pen(Color.Silver), 0, 0, image.Width - 1, image.Height - 1);
            this.pictureBox1.Width = image.Width;       //设置 PictureBox 的宽度
            this.pictureBox1.Height = image.Height;     //设置 PictureBox 的高度
            this.pictureBox1.BackgroundImage = image;   //设置 PictureBox 的背景图像
            }
            catch
            { }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            CodeImage(CheckCode());
        }
    }
}
```

生成几个数字加字符的混合式的并有背景图案的（增加噪声干扰）验证码（目前建议使用这一种），或者参考下列程序：

■ 图形验证码：（C#版）

```
    private void button1_Click(object sender, EventArgs e)
            {
              string chkCode = string.Empty;
            //颜色列表，用于验证码、噪线、噪点
    Color[] color ={ Color.Black, Color.Red, Color.Blue, Color.Green,
Color.Orange,Color.Brown, Color.DarkBlue };
                //字体列表，用于验证码
                string[] font ={ "Times New Roman", "MS Mincho", "Book Antiqua",
```

```
"Gungsuh", "PMingLiU", "Impact" };
                    //验证码的字符集，去掉了一些容易混淆的字符
                    char[] character ={ '2', '3', '4', '5', '6', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'J',
'K', 'L', 'M', 'N', 'P', 'R', 'S', 'T', 'W', 'X', 'Y' };
                    Random rnd = new Random();
                    //生成验证码字符串
                    for (int i = 0; i < 4; i++)
                    {
                        chkCode += character[rnd.Next(character.Length)];
                    }

                    Bitmap bmp = new Bitmap(150, 50);
                    Graphics g = Graphics.FromImage(bmp);
                    g.Clear(Color.White);

                    //画噪线
                    for (int i = 0; i < 3; i++)
                    {
                        int x1 = rnd.Next(150);
                        int y1 = rnd.Next(30);
                        int x2 = rnd.Next(150);
                        int y2 = rnd.Next(30);
                        Color clr = color[rnd.Next(color.Length)];
                        g.DrawLine(new Pen(clr), x1, y1, x2, y2);
                    }

                    //画验证码字符串
                    for (int i = 0; i < chkCode.Length; i++)
                    {
                        string fnt = font[rnd.Next(font.Length)];
                        Font ft = new Font(fnt, 16);
                        Color clr = color[rnd.Next(color.Length)];
 g.DrawString(chkCode[i].ToString(), ft, new SolidBrush(clr), (float)i * 20 + 20, (float)6); }

                    //画噪点
                    for (int i = 0; i < 50; i++)
                    {
                        int x = rnd.Next(bmp.Width);
                        int y = rnd.Next(bmp.Height);
                        Color clr = color[rnd.Next(color.Length)];
                        bmp.SetPixel(x, y, clr);
                    }
                    pictureBox1.Image = bmp;
            }
```

实验二　计算给定报文的 HASH 值，其中包括 SHA1、SHA256、MD5 等函数的使用。

(一)本实验的目的：理解信息安全技术中一个很重要的元素或技术：hash 函数，了解它的性质和意义，掌握如何使用 HASH 函数。

（参考教材 P166）

(二)实验内容：计算一个报文（信息）的 hash 值。

(三)操作步骤（说明:利用 Microsoft Visual Studio 环境，同学们可自己下载）：

（1） 打开 Microsoft Visual Studio，-----文件-----新建项目------（选择语言：Visual C#）控制台应用程序

（2） 在命名空间中输入：

```
using System.IO;
using System.Security.Cryptography;
```

（3） 在 static void Main(string[] args)方法中输入以下代码：

```
HashAlgorithm   hash=HashAlgorithm.Create();
Console.WriteLine("Enter a File Name:");
string fileName=Console.ReadLine();
FileStream fs=new FileStream(fileName, FileMode.Open);
byte[] hashBytes=hash.ComputeHash(fs);
fs.Close();
//display the hash data
Console.Write("Hash:"+BitConverter.ToString(hashBytes));
Console.Read();
```

（4） 需要在指定的位置下建一个文件

例如：在 c 盘的根目录下建一个 aa.txt 文件，内容自己确定

（5） 然后运行程序，观察结果

进一步观察实验：将上题中的报文内容略微改动，再运行、观察其散列值，看有什么变化。

注：该句 HashAlgorithm　hash=HashAlgorithm.Create();是实现是 SHA1 类的实例，生成的是 160 位的散列码。

如果将上句改为：HashAlgorithm hash=HashAlgorithm.Create("SHA256");

则是生成 256 位的散列码。

或者：SHA256Managed　hash=new　SHA256Managed();　生成 256 位的散列码。

- 实验内容的变化：

将上例中的语句 HashAlgorithm　hash=HashAlgorithm.Create(); 改为：

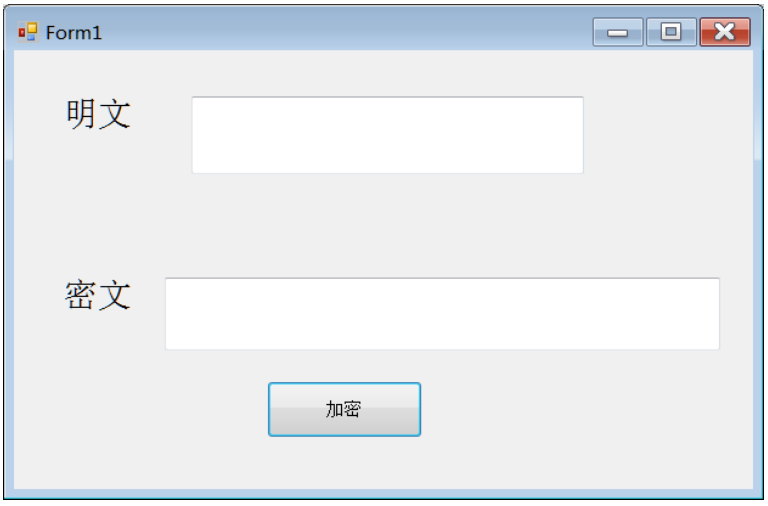
MD5 md5 = new MD5CryptoServiceProvider();

语句 byte[] hashBytes=hash.ComputeHash(fs);改为：

byte[] hashBytes = md5.ComputeHash(fs);则是 hash 函数 MD5 的哈希值（128 位）。

- 更进一步：

为便于应用和操作，可以将以上实验内容改为 Windows 界面输入和输出



则程序稍作更改即可：

```
private void button1_Click(object sender, EventArgs e)
        {
//textBox1 为输入密码的文本框
        byte[] result = Encoding.Default.GetBytes(this.textBox1.Text.Trim());
        HashAlgorithm hash = HashAlgorithm.Create();
        byte[] output = hash.ComputeHash(result);
        //textBox2 为输出加密文本的文本框
        this.textBox2.Text = BitConverter.ToString(output);
    }
```

将上例中的语句：

```
HashAlgorithm hash = HashAlgorithm.Create();
byte[] output = hash.ComputeHash(result);
```

改为:

```
MD5 md5 = new MD5CryptoServiceProvider();
byte[] output = md5.ComputeHash(result);
```

则是应用 MD5，计算报文的 HASH 值。

（四） 要求：提交实验报告，实验报告内容要求如下：

写出你的实验过程，利用的是哪个散列函数，你的原报文内容和运算后哈希值。写出你实验过程中遇到的问题，有哪些收获等？

实验三　RSA 加密实验

(五)本实验的目的：理解 RSA 加密算法；掌握 RSA 加密算法的实现。
（参考教材 P166）

(六)实验内容：RSA 加密算法的实现，并对给出的信息加密。

(七)操作步骤（说明:利用 Microsoft Visual Studio 环境，同学们可自己下载）：

（6） 打开 Microsoft Visual Studio，-----文件-----新建项目------（选择语言：Visual C#）控制台应用程序

（7） 在命名空间中输入：

```
using System.IO;
using System.Security.Cryptography;
```

（8） RSA 加解密算法：（下面的代码已运行）

```
namespace ConsoleApplication63
{
    class Program
    {

        static void Main(string[] args)
        {
            try
            {
                string     str_Plain_Text = "How are you? ";
                Console.WriteLine("明文：" + str_Plain_Text);
                Console.WriteLine("      长      度      :      " +
str_Plain_Text.Length.ToString());
                Console.WriteLine();

                RSACryptoServiceProvider      RSA      =      new
RSACryptoServiceProvider();

                string str_Public_Key;
                string str_Private_Key;
                string  str_Cypher_Text  =  RSA_Encrypt(str_Plain_Text,
out str_Public_Key, out str_Private_Key);
                Console.WriteLine("密文：" + str_Cypher_Text);
                Console.WriteLine("公钥：" + str_Public_Key);
                Console.WriteLine("私钥：" + str_Private_Key);

                string  str_Plain_Text2  =  RSA_Decrypt(str_Cypher_Text,
str_Private_Key);
                Console.WriteLine("解密：" + str_Plain_Text2);

                Console.WriteLine();
```

```
            }
            catch (ArgumentNullException)
            {
                Console.WriteLine("Encryption failed.");
            }

        }
        //RSA 加密,随机生成公私钥对并作为出参返回
        static public string RSA_Encrypt(string str_Plain_Text, out string str_Public_Key, out string str_Private_Key)
        {
            str_Public_Key = "";
            str_Private_Key = "";
            UnicodeEncoding ByteConverter = new UnicodeEncoding();
            byte[] DataToEncrypt = ByteConverter.GetBytes(str_Plain_Text);
            try
            {
                RSACryptoServiceProvider RSA = new RSACryptoServiceProvider();
                str_Public_Key = Convert.ToBase64String(RSA.ExportCspBlob(false));
                str_Private_Key = Convert.ToBase64String(RSA.ExportCspBlob(true));

                //OAEP padding is only available on Microsoft Windows XP or later.
                byte[] bytes_Cypher_Text = RSA.Encrypt(DataToEncrypt, false);
                str_Public_Key = Convert.ToBase64String(RSA.ExportCspBlob(false));
                str_Private_Key = Convert.ToBase64String(RSA.ExportCspBlob(true));
                string str_Cypher_Text = Convert.ToBase64String(bytes_Cypher_Text);
                return str_Cypher_Text;
            }
            catch (CryptographicException e)
            {
                Console.WriteLine(e.Message);
                return null;
            }
        }
```

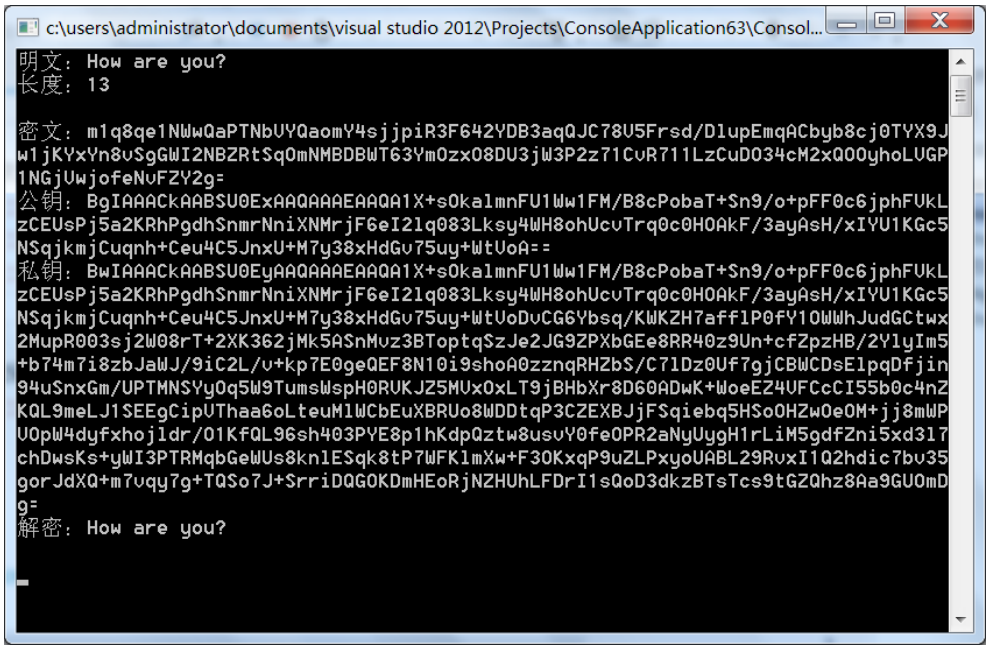
```
        //RSA 解密
        static public string RSA_Decrypt(string str_Cypher_Text, string str_Private_Key)
        {
            byte[] DataToDecrypt = Convert.FromBase64String(str_Cypher_Text);
            try
            {
                RSACryptoServiceProvider RSA = new RSACryptoServiceProvider();
                //RSA.ImportParameters(RSAKeyInfo);
                byte[] bytes_Public_Key = Convert.FromBase64String(str_Private_Key);
                RSA.ImportCspBlob(bytes_Public_Key);

                //OAEP padding is only available on Microsoft Windows XP or later.
                byte[] bytes_Plain_Text = RSA.Decrypt(DataToDecrypt, false);
                UnicodeEncoding ByteConverter = new UnicodeEncoding();
                string str_Plain_Text = ByteConverter.GetString(bytes_Plain_Text);
                return str_Plain_Text;
            }
            catch (CryptographicException e)
            {
                Console.WriteLine(e.ToString());
                return null;
            }
        }
    }
}
```

（八） 要求：提交实验报告，实验报告内容要求如下：

(1) 写出你的实验过程，利用 RSA 加密算法讲你的姓名加密，写出你的加密结果，并给出公钥和私钥。

(2) 写出你实验过程中遇到的问题，有哪些收获等？


c:\users\administrator\documents\visual studio 2012\Projects\ConsoleApplication63\Consol...
明文：How are you?
长度：13
密文：m1q8qe1NWwQaPTNbUYQaomY4sjjpiR3F642YDB3aqQJC78U5Frsd/DlupEmqACbyb8cj0TYX9J
w1jKYxYn8vSgGWI2NBZRtSqOmNMBDBWT63YmOzxO8DU3jW3P2z71CvR711LzCuDO34cM2xQOOyhoLUGP
1NGjUwjofeNvFZY2g=
公钥：BgIAAACkAABSU0ExAAQAAAEAAQA1X+sOkalmnFU1Ww1FM/B8cPobaT+Sn9/o+pFF0c6jphFUkL
zCEUsPj5a2KRhPgdhSnmrNniXNMrjF6eI2lq083Lksy4WH8ohUcvTrq0c0HOAkF/3ayAsH/xIYU1KGc5
NSqjkmjCuqnh+Ceu4C5JnxU+M7y38xHdGv75uy+WtVoA==
私钥：BwIAAACkAABSU0EyAAQAAAEAAQA1X+sOkalmnFU1Ww1FM/B8cPobaT+Sn9/o+pFF0c6jphFUkL
zCEUsPj5a2KRhPgdhSnmrNniXNMrjF6eI2lq083Lksy4WH8ohUcvTrq0c0HOAkF/3ayAsH/xIYU1KGc5
NSqjkmjCuqnh+Ceu4C5JnxU+M7y38xHdGv75uy+WtVoDvCG6Ybsq/KWKZH7affl P0fY1OWWhJudGCtwx
2MupR003sj2W08rT+2XK362jMk5ASnMvz3BToptqSzJe2JG9ZPXbGEe8RR40z9Un+cfZpzHB/2Y1yIm5
+b74m7i8zbJaWJ/9iC2L/v+kp7E0geQEF8N10i9shoA0zznqRHZbS/C71Dz0Uf7gjCBWCDsElpqDfjin
94uSnxGm/UPTMNSYyOq5W9TumsWspH0RVKJZ5MVxOxLT9jBHbXr8D60ADwK+WoeEZ4VFCcCI55b0c4nZ
KQL9meLJ1SEEgCipVThaa6oLteuMlWCbEuXBRUo8WDDtqP3CZEXBJjFSqiebq5HSoOHZwOeOM+jj8mWP
UOpW4dyfxhojldr/O1KfQL96sh403PYE8p1hKdpQztw8usvY0feOPR2aNyUygH1rLiM5gdfZni5xd317
chDwsKs+yWI3PTRMqbGeWUs8knlESqk8tP7WFKlmXw+F3OKxqP9uZLPxyoUABL29RvxI1Q2hdic7bv35
gorJdXQ+m7vqy7g+TQSo7J+SrriDQGOKDmHEoRjNZHUhLFDrI1sQoD3dkzBTsTcs9tGZQhz8Aa9GUOmD
g=
解密：How are you?

实验四　　注册密码的 Hash 值存入数据库，并实现登录

(一) 实验目的：理解和掌握 Hash 函数的实际应用，提升学生的解决实际问题的能力

(二) 实验内容：

（1） 掌握把注册密码的 Hash 值存入数据库的方法和技术

（2） 掌握实现登录的方法和技术

（3） 掌握连接后台数据库的方法和技术

(三) 实验步骤：

1） 建立数据库 shiyan:

打开 SQL Server-----建立数据库（shiyan）------建立表(denglu). 表结构如下：

zhanghu	nchar(20)
mima	nchar(200)

2) 设计如下界面：

3. 注册的程序部分：

1）引入命名空间：

using System.Data.SqlClient;

using System.Security.Cryptography;

代码：

```
private void button1_Click(object sender, EventArgs e)
        {
            string   strMima;
        string sqlstr = "Data Source=AAA\\SQLEXPRESS; Initial Catalog=shiyan; Integrated Security=True";   连接数据库的字符串
            SqlConnection conn = new SqlConnection(sqlstr);   //连接数据库

            byte[] result = Encoding.Default.GetBytes(this.textBox1.Text.Trim());   //计算输入密码的 hash 值
```

```
            HashAlgorithm hash = HashAlgorithm.Create();
            byte[] output = hash.ComputeHash(result);
            //textBox2 为输出加密文本的文本框
           this.textBox2.Text = BitConverter.ToString(output); // hash 值转变为文本型
                strMima= this.textBox2.Text;
            string strsql = "insert  into  zhangh(zhanghu, mima) values (' " + textBox1.Text +
" ', ' " + textBox2.Text + " ')";  //把密码的 hash 值插入数据表中
            SqlCommand comm = new SqlCommand(strsql, conn);  //执行命令
            if (conn.State == ConnectionState.Closed) //如果数据库关闭的话，打开数据库
            {
                conn.Open();
            }
            if (Convert.ToInt32(comm.ExecuteNonQuery()) > 0)  //如果添加成功
            {
                label1.Text = "添加成功！ ";
            }
            else
            {
                label1.Text = "添加失败！ ";
            }
            conn.Close();  //关闭数据库
        }
```

■ 登录部分程序：

```
    private void button2_Click(object sender, EventArgs e)
        {
            string User,Pwd; //用户名，密码
            string  strHash;
            string username = textBox1.Text.Trim();
            string userpwd = textBox2.Text.Trim();

            byte[] result = Encoding.Default.GetBytes(this.textBox2.Text.Trim());
            HashAlgorithm hash = HashAlgorithm.Create();
            byte[] output = hash.ComputeHash(result);
            //textBox2 为输出加密文本的文本框
            strHash= BitConverter.ToString(output);

string sqlstr = "Data Source=SU-PC; Initial Catalog=shiyan; Integrated Security=True";
            SqlConnection conn = new SqlConnection(sqlstr);

            string sql = "select  *  from  denglu"  ;  //定义查询命令
            conn.Open();

            SqlCommand com = new SqlCommand(sql, conn);
```

```
            SqlDataReader sread = com.ExecuteReader();   //执行查询:提供一种读取数据库行的方式

            while (sread.Read())     //从数据库中读取用户信息
            {
                User = sread["zhanghu"].ToString();     //或者用 User = sread.GetString(0);
                Pwd= sread["mima"].ToString();          //或者用 Pwd = sread.GetString(1);
                if (User.Trim() == username & Pwd.Trim() == strHash)   //如果输入的账户和密码值正确的话
                {
                    //MessageBox.Show("成功！");
                    Form tt = new Form2();     //转到另一个窗口
                    tt.Show();
                }
            }
                conn.Close();//关闭连接
                conn.Dispose();//释放连接
                sread.Dispose();//释放资源
    }
```

(四) 根据以上实验示例，自己设计一个能实现注册和登录的程序，要求：

1） 注册密码的 Hash 值存入数据库

2） 实现登录后转到另一个窗口

3） 写出你的实验的遇到的问题 ，心得体会。

■ **说明：该实验是综合性实验，是选做内容，可做可不做。每位同学可可根据自己的实际情况决定。如果你做个实验后，你会收获多多，脑洞打开！**