1.C#对注册表的操作-------------------------------------------------------------------------------------------------3

2.choosesubject--------------------------------------------------------------------------------------------------------4

3.n个数排序------------------------------------------------------------------------------------------------------------5

4.unknown---------------------------------------------------------------------------------------------------------------6

5.猜数字------------------------------------------------------------------------------------------------------------------8

6.猜数字个人版---------------------------------------------------------------------------------------------------------9

7.词频统计--------------------------------------------------------------------------------------------------------------10

8.递归方法求阶乘字符串反转------------------------------------------------------------------------------------12

9.第二种方法求一系列数的和------------------------------------------------------------------------------------14

10.订票-------------------------------------------------------------------------------------------------------------------15

11.发奖学金-------------------------------------------------------------------------------------------------------------17

12.构造函数属性------------------------------------------------------------------------------------------------------20

13.关闭特定程序------------------------------------------------------------------------------------------------------21

14.何意数排序----------------------------------------------------------------------------------------------------------22

15.加密--------------------------------------------------------------------------------------------------------------------23

16.解一元二次方程的解矩阵相乘--------------------------------------------------------------------------------24

17.九九乘法表-----------------------------------------------------------------------------------------------------------25

18.矩形---------------------------------------------------------------------------------------------------------------------25

19.矩阵相乘---------------------------------------------------------------------------------------------------------------27

20.矩阵相乘-改进加入异常处理------------------------------------------------------------------------------------29

21.利用列表排n个数的序--------------------------------------------------------------------------------------------33

22.朦胧诗-------------------------------------------------------------------------------------------------------------------34

23.判断是否为汉字------------------------------------------------------------------------------------------------------35

24.棋盘-----------------------------------------------------------------------------------------------------------------------36

25.求N个数的阶乘------------------------------------------------------------------------------------------------------37

26.求次幂的简便方法----------------------------------------------------------------------------------------------------37

27.求和平均值--------------------------------------------------------------------------------------------------------------38

28.求阶乘---------------------------------------------------------------------------------------------------------------------39

29.求平均成绩---------------------------------------------------------------------------------------------------------------40

30.求平均成绩使用二维数组--------------------------------------------------------------------------------------------41

31.求三阶行列式的值------------------------------------------------------------------------------------------------------42

32.求素数----------------------------------------------------------------------------------------------------------------------43

33.求素数最简单-------------------------------------------------------------------------------------------------------------44

34.求一系列数的和----------------------------------------------------------------------------------------------------------45

35.求一系列整数的和-------------------------------------------------------------------------------------------------------46

36.求质数-----------------------------------------------------------------------------------------------------------------------46

37.三个数排序-----------------------------------------------------------------------------------------------------------------47

38.三个数最大数--------------------------------------------------------------------------------------------------------------48

39.身份验证---------------------------------------------------------------------------------------------------------------------49

40.十进制转换成二进制----------------------------------------------------------------------------------------------------50

41.实时获取CPU使用率----------------------------------------------------------------------------------------------------51

42.实现关机-危险勿试-------------------------------------------------------------------------------------------------------53

43. 实现一个数的N次方---------------------------------------------------------------------------------------------------54

44.输出素数----------------------------------------------------------------------------------------------------------------------55

45.输出随机数-----------------------------------------------------------------------------------------------------------------56

46.输出图形---------------------------------------------------------------------------------------------------------------------57

47.宿舍值日---------------------------------------------------------------------------------------------------------------------58

48.验证概率---------------------------------------------------------------------------------------------------------------------59

49.一到一百之间的素数-----------------------------------------------------------------------------------------------------61

50.以二进制读取文本文件--------------------------------------------------------------------------------------------------62

C#中对注册表的操作

Windows 操作系统的注册表包含了很多有关计算机运行的配置方式，打开注册表我们可以看到注册表是按类似于目录的树结构组织的，其中第二级目录包含了五个预定义主键分别是：HKEY\_CLASSES\_ROOT，HKEY\_CURRENT\_USER，HKEY\_LOCAL\_MACHINE，HKEY\_USERS，HKEY\_CURRENT\_CONFIG。

下面我们来分别解释这5个类的作用

　　HKEY\_CLASSES\_ROOT该主键包含了文件的扩展名和应用程序的关联信息以及Window Shell和OLE用于储存注册表的信息。该主键下的子键决定了在WINDOWS中如何显示该类文件以及他们的图标，该主键是从HKEY\_LCCAL\_MACHINE\SOFTWARE\Classes映射过来的。

HKEY\_CURRENT\_USER该主键包含了如用户窗口信息，桌面设置等当前用户的信息。

HKEY\_LOCAL\_MACHINE主键包含了计算机软件和硬件的安装和配置信息，该信息可供所有用户使用

　　HKEY\_USERS该主键记录了当前用户的设置信息，每次用户登入系统时，就会在该主键下生成一个与用户登入名一样的子键，该子键保存了当前用户的桌面设置、背景位图、快捷键，字体等信息。一般应用程序不直接访问改主键，而是通过主键HKEY\_CURRENT\_USER进行访问。

　　HKEY\_CURRENT\_CONFIG该主键保存了计算机当前硬件的配置信息，这些配置可以根据当前所连接的网络类型或硬件驱动软件安装的改变而改变。

　　C#也支持对注册表的编辑，.NET框架在Microsoft.Win32名字空间中提供了两个类来操作注册表：Registry和RegistryKey。这两个类都是密封类不允许被继承。下面我们分别来介绍这两个类。

　　Registry类提供了7个公共的静态域，分别代表7个基本主键（其中两个在XP系统中没有，在这就不介绍了）分别是：Registry.ClassesRoot，Registry.CurrentUser，Registry.LocalMachine，Registry.Users，Registry.CurrentConfig。它们分别对应哪几个键我想各位一看就会知道吧。

RegistryKey类中提供了对注册表操作的方法。要注意的是操作注册表必须符合系统权限，否则将会抛出错误。

　　下面我们就来几个操作注册表常用的几个方法

　　创建子键的方法原型为：

　　public RegistryKey CreateSubKey(string sunbkey);

　　参数sunbkey表示要创建的子键的名称或路径名。创建成功返回被创建的子键，否则返回null。

　　打开子键的方法原型为：

　 public RegistryKey OpenSubKey(string name);

　　public RegistryKey OpenSubKey(string name,bool writable);

　　参数name表示要打开的子键名或其路径名，参数writable表示被打开的子键是否允许被修改，第一个方法打开的子键是只读的。Microsoft.Win32类还为我们提供了另一个方法，用于打开远程计算机上的注册表，方法原型为：

　　public static RegistryKey OpenRemoteBaseKey(RegistryHive hKey,string machineName);

　　删除子键的方法原型为：

　　public void DeleteKey(string subkey);

　　该方法用于删除指定的主键。如果要删除的子键还包含主键则删除失败，并返回一个异常，如果要彻底删除该子键极其目录下的子键可以用方法DeleteSubKeyTree，该方法原型如下：

　　public void DeleteKeyTree(string subkey);

　　读取键值的方法原型如下：

　public object GetValue(string name);

　　public object GetValue(string name,object defaultValue);

参数name表示键的名称，返回类型是一个object类型，如果指定的键不存在则返回null。如果失败又不希望返回的值是null则可以指定参数defaultValue，指定了参数则在读取失败的情况下返回该参数指定的值。

　　设置键值的方法原型如下：public object SetValue(string name,object value);

Choosesubject

using System;

class ChooseSubject

{

static void Main()

{

int i;

string str;

Console.WriteLine ("Please choose your favorite subjects:-1 is quit.");

Console.WriteLine ("1.Chinese.");

Console.WriteLine ("2.Maths.");

Console.WriteLine ("3.English.");

Console.WriteLine ("4.Physical.");

Console.WriteLine ("-1 is quit.");

do

{

str=Console.ReadLine ();

i=Int32.Parse (str);

switch(i)

{

case 1:

Console.WriteLine ("You like chinese very much.");

break;

case 2:

Console.WriteLine ("You like maths very much.");

break;

case 3:

Console.WriteLine ("You like english very much.");

break;

case 4:

Console.WriteLine ("You like physical very much.");

break;

case -1:

break;

default:

Console.WriteLine ("I'm sorry.You can't choose this subject.");

break;

}

}while(i!=-1);

Console.WriteLine ("Byebye!");

}

}

n个数排序

using System;

namespace n个数排序

{

class Class1

{

static void Main(string[] args)

{Console.WriteLine("请输入你要排几个数的顺序:\n");

int n;//接收你要几个数排序

n=int.Parse(Console.ReadLine());

int[] A=new int[n];

Console.WriteLine("请依次输入你要排序的数(数的结束用回车表示):\n");

for(int i=0;i<n;i++)//接收你要排序的数

A[i]=int.Parse(Console.ReadLine());

for(int i=0;i<n;i++)//排序

{

for(int j=i+1;j<n;j++)

{

int temp;

if(A[i]>A[j])

{

temp=A[j];

A[j]=A[i];

A[i]=temp;

}

}

}

Console.WriteLine("排序的结果为:\n");

for(int i=0;i<n;i++)

Console.Write(" "+A[i]);

System.Threading.Thread.Sleep(10000);

}

}

}

S

Unknown

using System;

namespace dirtysalt

{

public class Car//class of car,has attribute of 'weight' and 'speed'

{

private int weight;

private int speed;

public Car(int Weight,int Speed)

{

weight=Weight;

speed=Speed;

}

public void setweight(int Weight)

{

weight=Weight;

}

public void setspeed(int Speed)

{

speed=Speed;

}

public int getspeed()

{

return speed;

}

public int getweight()

{

return weight;

}

};

public class Sportcar:Car//inherit class of Car,has attributes of 'weight','speed','color'

{

private string color;

public Sportcar(int Weight,int Speed,string Color):base(Weight,Speed)

{

setweight(Weight);

setspeed(Speed);

color=Color;

}

public void setcolor(string Color)

{

color=Color;

}

public string getcolor()

{

return color;

}

public static void Main()

{

Car car=new Car(100,100);

Sportcar sportcar=new Sportcar(100,200,"blcak");//here has a problem

Console.WriteLine("car's weight is "+car.getweight());

Console.WriteLine("car's speed is "+car.getspeed());

Console.WriteLine("sportcar's weight is "+sportcar.getweight());

Console.WriteLine("sportcar's speed is "+sportcar.getspeed());

Console.WriteLine("sportcar's speed is "+sportcar.getcolor());

}

}

}

猜数字

using System;

namespace \_02\_26

{

class Class\_02\_26

{

public static void Main()

{

string sTemp;

int iNum=new Random ().Next ()%100;

int iGuess=0,iCount=0;

Console.WriteLine ("请猜猜我想到的一到一百之间的数字.");

do

{

sTemp=Console.ReadLine ();

try

{

iGuess=Int32.Parse (sTemp);

if(iGuess>iNum)

{

Console.WriteLine ("太大了.");

}

else if (iGuess<iNum)

{

Console.WriteLine ("太小了.");

}

}

catch(Exception e)

{

Console.WriteLine ("你输入的不是一个有效整数.");

}

finally

{

Console.WriteLine ("你已经猜了{0}次了.",++iCount);

}

}while(iGuess!=iNum);

}

}

}

猜数字个人版

using System;

public class guess\_number

{

public static void Main()

{

int i\_random=new Random ().Next (100);

int i\_guess=0;

int i\_count=0;

Console.WriteLine ("猜一猜这样一个随机数,它的范围是在零到一百之间.");

for(i\_count=0;i\_count<10;i\_count++)

{

i\_guess=int.Parse (Console.ReadLine ());

if(i\_guess<i\_random)

{

Console.WriteLine ("太小了,大一点儿好些.");

}

if(i\_guess>i\_random)

{

Console.WriteLine ("太大了,小一些儿好些.");

}

}

if(i\_guess==i\_random&&i\_count!=10)

{

Console.WriteLine ("不错,不错,猜对了!");

}

else if(i\_guess==i\_random&&i\_count==10)

{

Console.WriteLine ("在这功败垂成的时候,你猜对了,更加幸运!");

}

else if(i\_count==10&&i\_guess!=i\_random)

{

Console.WriteLine ("对不起,你已经猜了十次,不能再猜了.");

}

Console.WriteLine (i\_random);

}

}

词频统计

using System;

using System.Drawing;

using System.Collections;

using System.ComponentModel;

using System.Windows.Forms;

using System.Data;

namespace WindowsApplication1

{

public class Form1 : System.Windows.Forms.Form

{

private System.Windows.Forms.TextBox textBox1;

private System.Windows.Forms.TextBox textBox2;

private System.Windows.Forms.Button button1;

/// <summary>

/// 必需的设计器变量。

/// </summary>

private System.ComponentModel.Container components = null;

public Form1()

{

//

// Windows 窗体设计器支持所必需的

//

InitializeComponent();

//

// TODO: 在 InitializeComponent 调用后添加任何构造函数代码

//

}

/// <summary>

/// 清理所有正在使用的资源。

/// </summary>

protected override void Dispose( bool disposing )

{

if( disposing )

{

if (components != null)

{

components.Dispose();

}

}

base.Dispose( disposing );

}

#region Windows 窗体设计器生成的代码

/// <summary>

/// 设计器支持所需的方法 - 不要使用代码编辑器修改

/// 此方法的内容。

/// </summary>

private void InitializeComponent()

{

this.textBox1 = new System.Windows.Forms.TextBox();

this.textBox2 = new System.Windows.Forms.TextBox();

this.button1 = new System.Windows.Forms.Button();

this.SuspendLayout();

//

// textBox1

//

this.textBox1.Location = new System.Drawing.Point(8, 32);

this.textBox1.Multiline = true;

this.textBox1.Name = "textBox1";

this.textBox1.ScrollBars = System.Windows.Forms.ScrollBars.Both;

this.textBox1.Size = new System.Drawing.Size(176, 344);

this.textBox1.TabIndex = 0;

this.textBox1.Text = "";

//

// textBox2

//

this.textBox2.Location = new System.Drawing.Point(328, 32);

this.textBox2.Multiline = true;

this.textBox2.Name = "textBox2";

this.textBox2.ScrollBars = System.Windows.Forms.ScrollBars.Both;

this.textBox2.Size = new System.Drawing.Size(168, 344);

this.textBox2.TabIndex = 1;

this.textBox2.Text = "";

//

// button1

//

this.button1.Location = new System.Drawing.Point(216, 8);

this.button1.Name = "button1";

this.button1.TabIndex = 2;

this.button1.Text = "词频统计";

this.button1.Click += new System.EventHandler(this.button1\_Click);

this.AutoScaleBaseSize = new System.Drawing.Size(6, 14);

this.ClientSize = new System.Drawing.Size(512, 397);

this.Controls.Add(this.button1);

this.Controls.Add(this.textBox2);

this.Controls.Add(this.textBox1);

this.Name = "Form1";

this.Text = "Form1";

this.ResumeLayout(false);

}

#endregion

/// <summary>

/// 应用程序的主入口点。

/// </summary>

[STAThread]

static void Main()

{

Application.Run(new Form1());

}

private void button1\_Click(object sender, System.EventArgs e)

{

string s=textBox1.Text ;

char[] c={' ',',','.','?','\'',':',';','!'};

string[] ss=s.Split (c);

Hashtable ha=new Hashtable ();

foreach(string sss in ss)

{

if(ha.Contains (sss))

{

ha[sss]=(int)ha[sss]+1;

}

else

{

ha.Add (sss,1);

}

}

foreach(DictionaryEntry de in ha)

{

textBox2.AppendText (de.Key +":"+de.Value +"\n");

}

}

}

}

递归方法求阶乘字符串反转

using System;

class UsingRecursive

{

static void Main()

{

int intResult;

string strResult;

UsingRecursive myURec=new UsingRecursive ();

intResult=myURec.DoFactorial (10);

Console.WriteLine ("10的阶乘是:"+intResult);

strResult=myURec.DoStrRev ("This is a test string.");

Console.WriteLine ("字符串反转后为:"+"\n"+strResult);

}

int DoFactorial(int pF)

{

int result=0;

if(pF==1)

return 1;

result=DoFactorial(pF-1)\*pF;

return result;

}

string DoStrRev(string strTest)

{

if(strTest.Length==1)

return strTest;

string strResult=strTest.Substring (strTest.Length-1,1);

strResult+=DoStrRev(strTest.Substring (0,strTest.Length-1));

return strResult;

}

}

第二种方法求一系列数的和

using System;

using System.Diagnostics; //增加名字空间

namespace Count

{

class Test

{

public static void Main()

{

int x;

int y;

long temp = 0;

long sum = 0;

Console.WriteLine("input times:");

x = int.Parse(Console.ReadLine());

Console.WriteLine("input base:");

y = int.Parse(Console.ReadLine());

for(int i=1;i<=x;i++)

{

temp = 0; //这里你忘记要在每一次循环前清0

for(int j=1;j<=i;j++)

{

double m = Math.Pow(10,(j-1)); //次方不是^，这个是位运算符

temp+=y\*(int)m;

}

sum+=temp;

}

Console.WriteLine(sum);

Console.ReadLine();

}

}

}

订票

using System;

public class book\_ticket

{

public static void Main()

{

int[] A=new int [10];

int next;

Console.WriteLine ("欢迎使用南方航空公司订票系统:");

Console.WriteLine ("1预订头等仓(1-5号座位)\n2预订经济仓(6-10号座位)\n-1退出");

do

{

Console.WriteLine ("请选择:");

next=int.Parse (Console.ReadLine ());

switch(next)

{

case 1:

Console.WriteLine ("请输入座位号:");

int next1=int.Parse (Console.ReadLine ());

if((next1<1)||(next1>5))

{

Console.WriteLine ("您的输入有误,请重新输入(头等仓的座号范围是1-5)");

continue;

}

if(A[next1-1]==1)

{

Console.WriteLine ("抱歉...该座位{0}已被预订!",next1);

}

else

{

A[next1-1]=1;

Console.WriteLine ("ok...预订成功,您的座位是{0}号.",next1);

}

break;

case 2:

Console.WriteLine ("请输入座位号:");

int next2=int.Parse (Console.ReadLine ());

if((next2<6)||(next2>10))

{

Console.WriteLine ("您的输入有误,请重新输入(经济仓的座号范围是6-10)");

continue;

}

if(A[next2-1]==1)

{

Console.WriteLine ("抱歉...该座位{0}已被预订!",next2);

}

else

{

A[next2-1]=1;

Console.WriteLine ("ok...预订成功,您的座位是{0}号.",next2);

}

break;

case -1:

break;

default:

break;

}

}while(next!=-1);

}

}

发奖学金

using System;

public class student

{

protected string name;

protected int age;

protected decimal score;

public student(string name,int age,decimal score)

{

this.name=name;

this.age=age;

this.score=score;

}

public virtual string Name

{

get

{

return name;

}

set

{

name=value;

}

}

public virtual int Age

{

get

{

return age;

}

set

{

age=value;

}

}

public virtual decimal Socore

{

get

{

return score;

}

set

{

score=value;

}

}

}

public class good\_student:student

{

public decimal bursary=0.0m;

public good\_student(string g\_name,int g\_age,decimal g\_score,decimal g\_bursary):base(g\_name,g\_age,g\_score)

{

this.bursary =bursary;

}

public override string Name

{

get

{

return name;

}

set

{

name = value;

}

}

public override int Age

{

get

{

return age;

}

set

{

age = value;

}

}

public override decimal Socore

{

get

{

if(score>80.0m)

{

bursary+=1200.0m;

}

else

{

bursary=0.0m;

}

return score;

}

set

{

score = value;

}

}

}

public class MainClass

{

public static void Main()

{

student s=new student ("大明",23,32.0m);

Console.WriteLine ("\t{0},\t{1},\t{2}",s.Name ,s.Age ,s.Socore );

good\_student gs=new good\_student ("小明",32,89.0m,0);

Console.WriteLine ("\t{0},\t{1},\t{2},\t{3}",gs.Name,gs.Age ,gs.Socore,gs.bursary );

}

}

构造函数属性

using System;

public class Square

{

private int sidelong;

public Square()

{

Console.WriteLine ("我没有参数.");

}

public Square(int sidelong)

{

this.sidelong=sidelong;

}

public int Setsidelong(int sidelong)

{

this.sidelong=sidelong;

return sidelong;

}

public int Getsidelong(int sidelong)

{

return sidelong;

}

public void Print()

{

Console.WriteLine ("当前正方形的边长是：{0}",sidelong);

}

}

class Square\_Test

{

public static void Main()

{

Square s1=new Square ();

s1.Setsidelong (4);

s1.Print ();

Square s2=new Square ();

}

}

关闭特定程序

using System;

using System.Diagnostics;

class close\_special\_exe

{

static void Main()

{

Process[] myProcess;

myProcess=Process.GetProcessesByName ("Notepad");

foreach(Process instance in myProcess)

{

instance.WaitForExit (3000);

instance.CloseMainWindow ();

}

}

}

何意数排序

using System;

public class Test

{

public static void Main()

{ //不知道怎么搞的,两个不能同进用,想不明白,可能是前一种方法已经把值给改变了

int x;

int temp;

Console.WriteLine ("你想排几个数的序:");

x=int.Parse (Console.ReadLine ());

int[] array\_previous=new int [x];

for(int i=0;i<array\_previous.Length;i++)

{

Console.WriteLine ("请输入第{0}个数:",i+1);

array\_previous[i]=int.Parse (Console.ReadLine ());

}

for(int index=1;index<array\_previous.Length;index++)

{

if(array\_previous[index-1]<array\_previous[index])

{

temp=array\_previous[index-1];

array\_previous[index-1]=array\_previous[index];

array\_previous[index]=temp;

}

}

Console.WriteLine ("正序排列为:");

foreach(int pin in array\_previous)

{

Console.Write (pin+"\t");

}

Console.WriteLine ();

for(int index=1;index<array\_previous.Length;index++)

{

if(array\_previous[index-1]>array\_previous[index])

{

temp=array\_previous[index-1];

array\_previous[index-1]=array\_previous[index];

array\_previous[index]=temp;

}

}

Console.WriteLine ("反序排列为:");

foreach(int pin in array\_previous)

{

Console.Write (pin+"\t");

}

Console.WriteLine ();

}

}

加密

using System;

using System.Threading;

class My303

{

static void Main()

{

Console.WriteLine("请输入四位整数：");

int i=int.Parse(Console.ReadLine());

int first = i/1000;

int second = (i/100)%10;

int third = (i/10)%10;

int forth = i%10;

Console.WriteLine(first+" "+second+" "+third+" "+forth);

first = (first+7)%10;

second =(second+7)%10;

third = (third+7)%10;

forth = (forth+7)%10;

int temp=third;

third=first ;

first = temp;

temp= forth;

forth= second;

second = temp;

int pass = first\*1000+second\*100+third\*10+forth;

Console.WriteLine("Now is "+pass);

Thread.Sleep(5000);

}

}

解一元二次方程的解

using System;

class fangcheng

{

public static void Main()

{

//声名变量

double a;

double b;

double c;

double d;

double e;

double f;

double g;

double h;

double i;

double j;

double k;

Console.WriteLine("解一元二次方程");

//输入a的值

aa: Console.WriteLine("请输入a的值:");

a=double.Parse(Console.ReadLine());

//a的值不能为0

if(a==0)

{

Console.WriteLine("请注意a的值不能为0,请重新输入!");

//返回，重新输入

goto aa;

}

//输入b的值

Console.WriteLine("请输入b的值:");

b=double.Parse(Console.ReadLine());

//输入c的值

Console.WriteLine("请输入c的值:");

c=double.Parse(Console.ReadLine());

//一元二次方程的计算公式

d=b\*b;

e=4\*a\*c;

f=d-e;

g=(int)(Math.Sqrt(f));

i=-b+g;

j=-b-g;

h=i/(2\*a);

k=j/(2\*a);

//判断其根的状况

if(f==0)

{

Console.WriteLine("此方程有一根为:"+h);

}

else if(f>0)

{

Console.WriteLine("此方程有二根为:"+h);

Console.WriteLine(" "+k);

}

else

{

Console.WriteLine("此方程没有根");

}

}

}

九九乘法表

using System;

public class ChengFaBiao

{

public static void Main()

{

Console.WriteLine ("jiu jiu cheng fa biao");

for(int x=1;x<=9;x++)

{

for(int y=1;y<=x;y++)

{

Console.Write ("{0}\*{1}={2}\t",x,y,x\*y);

}

Console.WriteLine ();

}

}

}

矩形

using System;

class Rectangle

{

private float length=1.0f;

private float width=1.0f;

public float SetLength(float length)

{

this.length=length;

if((length<0.0f)||(length>2.0f))

{

Console.WriteLine ("长的取值范围应在1.0-2.0之内，请重新设值。");

return 0.0f;

}

return width;

}

public float SetWidth(float width)

{

this.width=width;

if((width<0.0f)||(width>2.0f))

{

Console.WriteLine ("宽的取值范围应在1.0-2.0之内，请重新设值。");

return 0.0f;

}

return width;

}

public float GetLength(float length)

{

return length;

}

public float GetWidth(float width)

{

return width;

}

public float perimeter()

{

return length\*2+width\*2;

}

public float area()

{

return length\*width;

}

}

class Rectangle\_Test

{

public static void Main()

{

Rectangle r1=new Rectangle ();

r1.SetLength (1.8f);

r1.SetWidth (1.2f);

Console.WriteLine (r1.perimeter ());

Console.WriteLine (r1.area ());

}

}

矩阵相乘

using System;

public class MatrixMultiply

{

public static void Main()

{

int a,b,c,d;

Console.WriteLine ("该程序将求出两个矩阵的积:");

Console.WriteLine ("请指定矩阵A的行数:");

a=int.Parse (Console.ReadLine ());

Console.WriteLine ("请指定矩阵A的列数:");

b=int.Parse (Console.ReadLine ());

int[,] MatrixA=new int [a,b];

for(int i=0;i<a;i++)

{

for(int j=0;j<b;j++)

{

Console.WriteLine ("请输入矩阵A第{0}行第{1}列的值:",i+1,j+1);

MatrixA[i,j]=int.Parse (Console.ReadLine ());

}

}

Console.WriteLine ("矩阵A输入完毕.");

Console.WriteLine ("请指定矩阵B的行数:");

c=int.Parse (Console.ReadLine ());

Console.WriteLine ("请指定矩阵B的列数:");

d=int.Parse (Console.ReadLine ());

int[,] MatrixB=new int [c,d];

for(int i=0;i<c;i++)

{

for(int j=0;j<d;j++)

{

Console.WriteLine ("请输入矩阵A第{0}行第{1}列的值:",i+1,j+1);

MatrixB[i,j]=int.Parse (Console.ReadLine ());

}

}

Console.WriteLine ("矩阵B输入完毕.");

Console.WriteLine ("矩阵A为:");

outputMatrix(MatrixA,a,b);

Console.WriteLine ("矩阵B为:");

outputMatrix(MatrixB,c,d);

if(b!=c)

{

Console.WriteLine ("矩阵A的列数与矩阵B的行数不相等,无法进行乘积运算!");

return;

}

else

{

Console.WriteLine ("矩阵A与矩阵B的乘积为:");

}

int[,] MatrixC=new int [a,d];

for(int i = 0; i < a; i++)

{

for(int j = 0; j < d; j++)

{

MatrixC[i,j] = 0;

for(int k = 0; k < b; k++)

{

MatrixC[i,j] += MatrixA[i,k] \* MatrixB[k,j];

}

}

}

outputMatrix(MatrixC,a,d);

}

private static void outputMatrix(int[,] MatrixX, int rowCount, int columnCount)

{

for(int i = 0; i < rowCount; i++)

{

for(int j = 0; j < columnCount; j++)

{

Console.Write(MatrixX[i,j] + "\t");

}

Console.WriteLine();

}

}

}

矩阵相乘-改进加入异常处理

using System;

public class MatrixMultiply

{

public static void Main()

{

int a=0,b=0,c=0,d=0;//矩阵A.B的行数.列数

Console.WriteLine ("该程序将求出两个矩阵的积:");

Console.WriteLine ("请指定矩阵A的行数:");

int exception\_number=0;

do

{

try

{

a=int.Parse(Console.ReadLine ());

exception\_number++;

}

catch(Exception e)

{

Console.WriteLine (e.Message );

Console.WriteLine ("请输入一个数字:");

}

}while(exception\_number==0);

Console.WriteLine ("请指定矩阵A的列数:");

do

{

try

{

b=int.Parse (Console.ReadLine ());

exception\_number++;

}

catch(FormatException e)

{

Console.WriteLine (e.Message );

Console.WriteLine ("请输入一个数字:");

}

}while(exception\_number==0);

int[,] MatrixA=new int [a,b];

for(int i=0;i<a;i++)

{

for(int j=0;j<b;j++)

{

exception\_number=0;

do

{

Console.WriteLine ("请输入矩阵A第{0}行第{1}列的值:",i+1,j+1);

try

{

MatrixA[i,j]=int.Parse (Console.ReadLine ());

exception\_number++;

}

catch(Exception e)

{

Console.WriteLine (e.Message );

Console.WriteLine ("请输入一个数字:");

}

}while(exception\_number==0);

}

}

Console.WriteLine ("矩阵A输入完毕.");

Console.WriteLine ("请指定矩阵B的行数:");

do

{

try

{

c=int.Parse (Console.ReadLine ());

exception\_number++;

}

catch(FormatException e)

{

Console.WriteLine (e.Message );

Console.WriteLine ("请输入一个数字:");

}

}while(exception\_number==0);

Console.WriteLine ("请指定矩阵B的列数:");

do

{

try

{

d=int.Parse (Console.ReadLine ());

}

catch(FormatException e)

{

Console.WriteLine (e.Message );

Console.WriteLine ("请输入一个数字:");

}

}while(exception\_number==0);

int[,] MatrixB=new int [c,d];

for(int i=0;i<c;i++)

{

for(int j=0;j<d;j++)

{

exception\_number=0;

do

{

Console.WriteLine ("请输入矩阵A第{0}行第{1}列的值:",i+1,j+1);

try

{

MatrixB[i,j]=int.Parse (Console.ReadLine ());

}

catch(Exception e)

{

Console.WriteLine (e.Message );

Console.WriteLine ("请输入一个数字:");

}

}while(exception\_number==0);

}

}

Console.WriteLine ("矩阵B输入完毕.");

Console.WriteLine ("矩阵A为:");

outputMatrix(MatrixA,a,b);

Console.WriteLine ("矩阵B为:");

outputMatrix(MatrixB,c,d);

if(b!=c)

{

Console.WriteLine ("矩阵A的列数与矩阵B的行数不相等,无法进行乘积运算!");

return;

}

else

{

Console.WriteLine ("矩阵A与矩阵B的乘积为:");

}

int[,] MatrixC=new int [a,d];

for(int i = 0; i < a; i++)

{

for(int j = 0; j < d; j++)

{

MatrixC[i,j] = 0;

for(int k = 0; k < b; k++)

{

MatrixC[i,j] += MatrixA[i,k] \* MatrixB[k,j];

}

}

}

outputMatrix(MatrixC,a,d);

}

private static void outputMatrix(int[,] MatrixX, int rowCount, int columnCount)

{

for(int i = 0; i < rowCount; i++)

{

for(int j = 0; j < columnCount; j++)

{

Console.Write(MatrixX[i,j] + "\t");

}

Console.WriteLine();

}

}

}

利用列表排n个数的序

using System;

using System.Collections ;

class ArraySort

{

static void Main()

{

ArrayList myArrayList=new ArrayList ();

Console.WriteLine ("你想排几个数的序:");

int iNumber=int.Parse (Console.ReadLine ());

for(int i=1;i<=iNumber;i++)

{

Console.WriteLine ("请输入第{0}个数:",i);

int temp=int.Parse (Console.ReadLine ());

myArrayList.Add (temp);

}

Console.WriteLine ("\n排序前的内容:");

ArraySort myArraySort=new ArraySort ();

myArraySort.WriteList (myArrayList);

myArrayList.Sort ();

Console.WriteLine ("\n逆序后的内容:");

myArraySort.WriteList (myArrayList);

myArrayList.Reverse ();

Console.WriteLine ("\n正序后的内容:");

myArraySort.WriteList (myArrayList);

}

void WriteList(ArrayList myArrayList)

{

Console.WriteLine ();

int pSize=myArrayList.Count;

for(int i=0;i<pSize;i++)

{

Console.Write (myArrayList[i]+"\t");

}

Console.WriteLine ();

}

}

朦胧诗

using System;

class Poem

{

public static string[] adjective=new string [8]{"美丽的","高兴的","伤心的","失落的","可爱的","调皮的","活波的","笨拙的"};

public static string[] subject=new string [8]{"太阳","花朵","星星","月亮","女孩","小猫","蝴蝶","燕子"};

public static string[] predicate=new string [7]{"思索着","追赶着","想念着","渴望着","怒视着","奔跑着","狂喊着"};

public static void Main()

{

string next;

do

{

Random r=new Random ();

for(int i=1;i<=2;i++)

{

Console.WriteLine (adjective[r.Next (adjective.Length )]+""+subject[r.Next (subject.Length )]+""+predicate[r.Next (predicate.Length )]+""+subject[r.Next (subject.Length )]);

}

Console.WriteLine ("你还想看吗?y/n");

next=Console.ReadLine ();

}while(next.ToUpper ()!="N");

}

}

判断是否为汉字

public bool IsChina(string CString)

{

bool BoolValue=false;

for (int i=0; i<CString.Length; i++)

{

if(Convert.ToInt32(Convert.ToChar(CString.Substring(i,1)))<Convert.ToInt32(Convert.ToChar(128)))

BoolValue = false;

}

else

{

BoolValue = true;

}

}

return BoolValue;

}

棋盘

using System;

class ChessBoard

{

public char[,] SquareColor=new char [8,8];

public ChessBoard()

{

for(int i=0;i<SquareColor.GetLength (0);i++)

{

for(int x=0;x<SquareColor.GetLength (1);x++)

{

if((x%2)==0)

if((i%2)==0)

SquareColor[i,x]='W';

else

SquareColor[i,x]='B';

else

if((i%2)==0)

SquareColor[i,x]='B';

else

SquareColor[i,x]='W';

}

}

}

void DrawBoard()

{

for(int i=0;i<SquareColor.GetLength (0);i++)

{

for(int x=0;x<SquareColor.GetLength (1);x++)

{

Console.Write (SquareColor[i,x]);

}

Console.WriteLine();

}

}

static void Main()

{

ChessBoard MyChessBoard=new ChessBoard ();

MyChessBoard.DrawBoard ();

}

}

求N个数的阶乘

using System;public class Factorial{

public long factorial(long x){

return ((x<=1) ? 1 : x\*(factorial(x-1)));

}

}

public class FactorialSum

{

public static void Main()

{

long temp=0;

long sum=0;

Console.WriteLine ("输入一个数:");

int x=int.Parse (Console.ReadLine ());

for(int i=1;i<=x;i++)

{

Factorial f1=new Factorial ();

temp+=f1.factorial (i);

}

sum+=temp;

Console.WriteLine ("1!+2!+3!+...+a!="+sum);

}}

求次幂的简便方法

using System;

class My

{

static void Main()

{

Pow r=new Pow ();

System.Threading .Thread.Sleep (10000);

}

}

class Pow

{

public Pow()

{

Console.WriteLine("请输入你要求幂的数：");

int x=int.Parse (Console.ReadLine());

Console.WriteLine ("数的幂是：");

int y=int.Parse (Console.ReadLine ());

int p=(int)Math.Pow (x,y);

Console.WriteLine ("数"+x+"的"+y+"次方是"+p);

}

}

求和平均值

using System;

using System.Collections;

class Sum\_Average

{

public static void Main()

{

int sum=0;

float average=0.0f;

Random r=new Random ();

ArrayList myArrayList=new ArrayList ();

Console.WriteLine ("你想算几个数的和,并求它的平均值:");

int iMax=int.Parse (Console.ReadLine ());

for(int i=0;i<iMax;i++)

{

myArrayList.Add (r.Next (0,100));

}

Console.WriteLine ("这{0}个数是:",iMax);

foreach(int pins in myArrayList)

{

Console.WriteLine (pins);

}

for(int j=0;j<iMax;j++)

{

sum+=(int)myArrayList[j];

}

Console.WriteLine ("这{0}个数的和是:{1}",iMax,sum);

average = (float)sum / iMax;

Console.WriteLine ("这{0}个数的平均值是:{1}",iMax,average);

}

}

求阶乘

using System;

class Test

{

public static void Main()

{

Console.WriteLine ("你想算几的阶乘:");

int x=int.Parse (Console.ReadLine ());

long y=1;

while(true)

{

y\*=x;

x--;

if(x==0)

{

break;

}

}

Console.WriteLine (y);

}

}

求平均成绩

using System;

class Score

{

public void score(int sequence\_number)

{

int sum=0;

int[] student=new int [5];

for(int i=0;i<=4;i++)

{

Console.WriteLine ("请输入第{0}个学生的第{1}门功课:",sequence\_number,i+1);

student[i]=int.Parse (Console.ReadLine ());

}

for(int index=0;index!=student.Length;index++)

{

sum+=student[index];

}

Console.WriteLine ("该生的平均成绩为"+sum/5);

}

}

class Average

{

public static void Main()

{

for(int i=1;i<=5;i++)

{

Score s=new Score ();

s.score (i);

}

}

}

求平均成绩使用二维数组

using System;

class AverageofScore

{

public static void Main()

{

int[,] student\_score=new int [4,5];

int temp;

for(int i=0;i<=3;i++)

{

temp=0;

for(int j=0;j<=4;j++)

{

Console.WriteLine ("请输入第{0}个学生的第{1}门功课:",i+1,j+1);

student\_score[i,j]=int.Parse (Console.ReadLine ());

temp+=student\_score[i,j];

}

Console.WriteLine ("This student's average of score is"+temp/5);

}

}

}

求三阶行列式的值

using System;

class Test

{

public static void Main()

{

Console.WriteLine ("该程序将求出三阶行列式的值:");

int[,] A=new int [3,3];

for(int i=0;i<3;i++)

{

for(int j=0;j<3;j++)

{

Console.WriteLine ("请输入第{0}行第{1}列的值:",i+1,j+1);

A[i,j]=int.Parse (Console.ReadLine ());

}

}

Console.WriteLine ("你输入的行列式为:");

for(int i=0;i<3;i++)

{

for(int j=0;j<3;j++)

{

Console.Write(A[i,j]+"\t");

}

Console.WriteLine ();

}

int result=A[0,0]\*A[1,1]\*A[2,2]+A[0,1]\*A[1,2]\*A[2,0]+A[1,0]\*A[2,1]\*A[0,2]-A[2,0]\*A[1,1]\*A[0,2]-A[1,0]\*A[0,1]\*A[2,2]-A[2,1]\*A[1,2]\*A[0,0];

Console.WriteLine ("该三行列式的值是:"+result);

}

}

求素数

using System;

namespace Other\_Features

{

public class Prime

{

public int number;

public Prime(int n)

{

if(isPrime(n))

this.number=n;

else

throw new Exception (n+" is not a prime.");

}

public static Prime operator ++ (Prime orig)

{

bool succeeded =false;

while(!succeeded)

succeeded=isPrime(++orig.number );

return orig;

}

public static bool isPrime(int number)

{

int max=(int)(number/2+1);

for(int i=2;i<max;++i)

if(number%i==0)

return false;

return true;

}

public static void Main()

{

Prime p=new Prime (1);

for(int i=0;i<100;++i)

{

Console.WriteLine (p.number);

++p;

}

}

}

}

求素数最简单

using System;

class Test

{

public static void Main()

{

int i;

Console.WriteLine (2);

Console.WriteLine (3);

Console.WriteLine (5);

Console.WriteLine (7);

for(i=2;i<=1000;i++)

{

if((i%2!=0)&&(i%3!=0)&&(i%5!=0)&&(i%7!=0))

{

Console.WriteLine (i+"\t");

}

}

}

}

求一系列数的和

using System;

class Class1

{

public static int x;

public static int y;

public static long temp=0;

public static long sum=0;

public static void Main()

{

Console.WriteLine ("请输入所要求这一系列数的和的个数:");

x=int.Parse (Console.ReadLine ());

Console.WriteLine ("请输入这一系列数的基数:");

y=int.Parse (Console.ReadLine ());

for(int i=1;i<=x;i++)

{

int temp=y;

for(int j=1;j<i;j++)

{

temp=temp\*10+y;

}

sum+=temp;

}

Console.WriteLine (sum);

Console.Read();

}

} 求一系列整数的和

using System;

using System.Threading;

class My302

{

static void Main()

{

Console.WriteLine("Please Input the Number:");

int all = int.Parse(Console.ReadLine());

int sum=0,temp=0;

for (int i=1;i<=all;i++)

{

Console.Write("Please Input the NO."+i+" value:");

temp=int.Parse(Console.ReadLine());

sum+=temp;

}

Thread.Sleep(1000);

Console.WriteLine("The all is "+sum);

Thread.Sleep(5000);

}

}

求质数

void main()

{

int n=2,i=2;

while(n<100)

{

for(int j=2;j<=sqrt(i);j++)

{

if(i%j)

continue;

else

break;

}

if(j>sqrt(i)) cout<<i<<endl;

i++;

n++;

}

}

三个数排序

using System;

public class Test

{

public static void Main()

{

int x;

int y;

int z;

Console.WriteLine ("请输入三个数:");

x=int.Parse (Console.ReadLine ());

y=int.Parse (Console.ReadLine ());

z=int.Parse (Console.ReadLine ());

if(x>y)

{

if(y>z)

{Console.WriteLine("{0},{1},{2}",x,y,z);}

else if(y<z)

{Console.WriteLine("{0},{1},{2}",x,z,y);}

else if(x<z)

{Console.WriteLine("{0},{1},{2}",z,x,y);}

}

else

if(x<y)

{

if(x>z)

{Console.WriteLine("{0},{1},{2}",y,x,z);}

else if(x<z)

{Console.WriteLine("{0},{1},{2}",y,z,x);}

else if(y<z)

{Console.WriteLine("{0},{1},{2}",z,y,x);}

}

}

}

三个数最大数

using System;

public class Test

{

public static void Main()

{

int x;

int y;

int z;

Console.WriteLine ("请输入三个数:");

x=int.Parse (Console.ReadLine ());

y=int.Parse (Console.ReadLine ());

z=int.Parse (Console.ReadLine ());

if(x>y)

{

if(x>z)

{

Console.WriteLine (x);

}

else

Console.WriteLine (z);

}

else

{

if(y>z)

{

Console.WriteLine (y);

}

else

{

Console.WriteLine (z);

}

}

}

}

身份验证

using System;

class validate

{

public static void Main()

{

string str\_wrench="ilovemyfamily";

string str\_next="";

Console.WriteLine ("请输入你的姓名:");

Console.ReadLine ();

do

{

Console.WriteLine ("请输入你的密码:");

string str\_passwd=Console.ReadLine ();

if(str\_passwd!=str\_wrench)

{

Console.WriteLine ("密码不正确,是否继续?Y/N");

}

str\_next=Console.ReadLine ();

}while(str\_next=="y"||str\_next=="Y");

}

}

十进制转换成二进制

using System;

class ggg

{

static void Main()

{

Console.WriteLine("请你输入十进制，我为你转换成二进制，谢谢使用。");

int a,b,d,f;

int c=0;

a=int.Parse(Console.ReadLine());

b=a;

do

{

b/=2;

c=++c;

}

while(b!=1);

d=c+1;

f=c;

int[] A=new int[d];

for(int i=c;i>0;i--)

{

A[i]=(int)a%2;

a/=2;

}

A[0]=1;

for(int j=0;j<=f;j++)

{

Console.Write(A[j]);

}

Console.WriteLine("\n"+"欢迎 使用，谢谢退出！");

System.Threading.Thread.Sleep(3000);

}

}

实时获取CPU使用率

using System;

using System.Diagnostics;

using System.Threading;

public class CpuLoadInfo

{

// auxiliary print methods

private static void Say(string txt)

{

Console.WriteLine(txt);

}

// auxiliary print methods

private static void Say()

{

Say("");

}

// The main method. Command line arguments are ignored.

public static void Main()

{

Say("$Id: CpuLoadInfo.cs,v 1.2 2002/08/17 17:45:48 rz65 Exp $");

Say();

Say("Attempt to create a PerformanceCounter instance:");

Say("Category name = " + CategoryName);

Say("Counter name = " + CounterName);

Say("Instance name = " + InstanceName);

PerformanceCounter pc = new PerformanceCounter(CategoryName,CounterName,InstanceName);

Say("Performance counter was created.");

Say("Property CounterType: " + pc.CounterType);

Say();

Say("Property CounterHelp: " + pc.CounterHelp);

Say();

Say("Entering measurement loop.");

while(true)

{

Thread.Sleep(1000); // wait for 1 second

float cpuLoad = pc.NextValue();

Say("CPU load = " + cpuLoad + " %.");

}

}

// constants used to select the performance counter.

private const string CategoryName = "Processor";

private const string CounterName = "% Processor Time";

private const string InstanceName = "\_Total";

}

实现关机-危险勿试

using System;

using System.Runtime.InteropServices;

class shoutdown{

[StructLayout(LayoutKind.Sequential, Pack=1)]

internal struct TokPriv1Luid

{

public int Count;

public long Luid;

public int Attr;

}

[DllImport("kernel32.dll", ExactSpelling=true)]

internal static extern IntPtr GetCurrentProcess();

[DllImport("advapi32.dll", ExactSpelling=true, SetLastError=true)]

internal static extern bool OpenProcessToken(IntPtr h, int acc, ref IntPtr phtok);

[DllImport("advapi32.dll", SetLastError=true)]

internal static extern bool LookupPrivilegeValue(string host, string name, ref long pluid);

[DllImport("advapi32.dll", ExactSpelling=true, SetLastError=true)]

internal static extern bool AdjustTokenPrivileges(IntPtr htok, bool disall,

ref TokPriv1Luid newst, int len, IntPtr prev, IntPtr relen);

[DllImport("user32.dll", ExactSpelling=true, SetLastError=true)]

internal static extern bool ExitWindowsEx(int flg, int rea);

internal const int SE\_PRIVILEGE\_ENABLED = 0x00000002;

internal const int TOKEN\_QUERY = 0x00000008;

internal const int TOKEN\_ADJUST\_PRIVILEGES = 0x00000020;

internal const string SE\_SHUTDOWN\_NAME = "SeShutdownPrivilege";

internal const int EWX\_LOGOFF = 0x00000000;

internal const int EWX\_SHUTDOWN = 0x00000001;

internal const int EWX\_REBOOT = 0x00000002;

internal const int EWX\_FORCE = 0x00000004;

internal const int EWX\_POWEROFF = 0x00000008;

internal const int EWX\_FORCEIFHUNG = 0x00000010;

private static void DoExitWin(int flg)

{

bool ok;

TokPriv1Luid tp;

IntPtr hproc = GetCurrentProcess();

IntPtr htok = IntPtr.Zero;

ok = OpenProcessToken(hproc, TOKEN\_ADJUST\_PRIVILEGES|TOKEN\_QUERY, ref htok);

tp.Count = 1;

tp.Luid = 0;

tp.Attr = SE\_PRIVILEGE\_ENABLED;

ok = LookupPrivilegeValue(null, SE\_SHUTDOWN\_NAME, ref tp.Luid);

ok = AdjustTokenPrivileges(htok, false, ref tp, 0, IntPtr.Zero, IntPtr.Zero);

ok = ExitWindowsEx(flg, 0);

}

public static void Main()

{

Console.WriteLine("正在关闭计算机……");

// 修改 EWX\_SHUTDOWN 或者 EWX\_LOGOFF, EWX\_REBOOT等实现不同得功能。

// 在XP下可以看到帮助信息，以得到不同得参数

// SHUTDOWN /?

DoExitWin(EWX\_SHUTDOWN);

}

}

实现一个数的N次方

using System;

class My

{

static void Main()

{

int a,b=1;

int n;

Console.WriteLine("请输入所要求这个数的次方数:");

n=int.Parse(Console.ReadLine());//n是次方

Console.WriteLine("请输入所要计算的这个数:");

a=int.Parse(Console.ReadLine());//a是那个数

for(int i=1;i<=n;i++)

{

b=b\*a;

}

Console.WriteLine(a+"的"+n+"次方是："+b);

}

}

输出素数

using System;

class Prime

{

static void Main()

{

int sum = 1;

Console.Write(2 + " ");

for(int m=3;m<=100;m+=2)

{

bool a = true;

if(m%2==0)

{

a = false;

}

else

{

for(int i=3;i<m/2;i+=2)

{

if(m%i==0)

{

a = false;

break;

}

}

}

if(a==true)

{

sum++;

Console.Write(m.ToString()+" ");

}

}

Console.WriteLine();

Console.WriteLine("total:" + sum);

}

}

输出随机数

using System;

namespace StaticConstructor

{

class RandomNumberGenerator

{

private static Random randomNumber;

static RandomNumberGenerator()

{

randomNumber=new Random ();

}

public int Next()

{

return randomNumber.Next ();

}

}

class Class1

{

static void Main(string[] args)

{

RandomNumberGenerator g1=new RandomNumberGenerator ();

for(int i=0;i<10;i++)

{

Console.WriteLine (g1.Next ());

}

}

}

}

输出图形

using System;

class My301

{

static void Main()

{

for(int i=1;i<=6;i++)//控制行数 ，输出图形上半部分

{

for(int k=5;k>=i;k--)

{

Console.Write(" ");

}

for (int x=1;x<2\*i;x++)

{

Console.Write("\*");

}

Console.WriteLine();

}

for(int i=1;i<=5;i++)//控制行数 输出图形下半部分

{

for(int k=1;k<=i;k++)

{

Console.Write(" ");

}

for (int x=11;x>2\*i;x--)

{

Console.Write("\*");

}

Console.WriteLine();

}

System.Threading.Thread.Sleep(5000);

}

}

宿舍值日

using System;

public class OnDuty

{

public static void Main()

{

int today;

do

{

Console.WriteLine ("Please input what day it is today:\n\n1 2 3 4 5 6 7 will be ok.and -1 is quit");

today=int.Parse (Console.ReadLine ());

switch(today)

{

case 1:

Console.WriteLine ("\nA is on duty today.He must make our room cleanly.");

break;

case 2:

Console.WriteLine ("\nB is on duty today.He msut make our room cleanly.");

break;

case 3:

Console.WriteLine ("\nC is on duty today.He msut make our room cleanly.");

break;

case 4:

Console.WriteLine ("\nD is on duty today.He must make our room cleanly.");

break;

case 5:

Console.WriteLine ("\nE is on duty today.He msut make our room cleanly.");

break;

case 6:

Console.WriteLine ("\nF is on duty today.He must make our room cleanly.");

break;

case 7:

Console.WriteLine ("\nG is on duty today.He must make our room cleanly.");

break;

case -1:

break;

default:

Console.WriteLine ("Invalid input.Please retry:");

break;

}

}while(today!=-1);

}

}

验证概率

using System;

namespace luntan

{

class Class1

{

static void Main(string[] args)

{

int x=0;

int[] Sum1=new int[1000];

for(int i=1;i<10;i++)

{

for(int j=1;j<10;j++)

{

if(i!=j)

{

for(int k=1;k<10;k++)

{

if(k!=i&&k!=j)

{

Sum1[x]=100\*i+10\*j+k;

x++;

}

}

}

}

}

string String1,String2,String3;

string[] Char=new string[9];

for(int i=0;i<x;i++)

{

for(int j=i+1;j<x;j++)

{

for(int k=j+1;k<x;k++)

{

if(Sum1[k]==(Sum1[i]+Sum1[j]))

{

String1=Sum1[k].ToString();

String2=Sum1[i].ToString();

String3=Sum1[j].ToString();

for(int y=0;y<3;y++)

Char[y]=String2.Substring(y,1);

for(int y=3;y<6;y++)

Char[y]=String3.Substring(y-3,1);

for(int y=6;y<9;y++)

Char[y]=String1.Substring(y-6,1);

int flag=0;

for(int y=0;y<9;y++)

{

for(int z=y+1;z<9;z++)

{

if(Char[y]==Char[z])

flag=1;

}

}

if(flag==0)

Console.WriteLine("{0}+{1}={2}",String2,String3,String1);

}

}

}

}

}

}

}

一到一百之间的素数

using System;

class Prime

{

static void Main()

{

int sum = 1;

Console.Write(2 + " ");

for(int m=3;m<=100;m+=2)

{

bool a = true;

if(m%2==0)

{

a = false;

}

else

{

for(int i=3;i<m/2;i+=2)

{

if(m%i==0)

{

a = false;

break;

}

}

}

if(a==true)

{

sum++;

Console.Write(m.ToString()+" ");

}

}

Console.WriteLine();

Console.WriteLine("total:" + sum);

}

}

以二进制读取文本文件

using System;

using System.IO;

public class FileApp

{

public static void Main()

{

// 在当前目录创建一个文件myfile.txt，对该文件具有读写权限

FileStream fsMyfile = new FileStream("myfile.txt" , FileMode.Create, FileAccess.ReadWrite);

// 创建一个数据流写入器，和打开的文件关联

StreamWriter swMyfile = new StreamWriter(fsMyfile);

// 以文本方式写一个文件

swMyfile.WriteLine("Hello, World");

swMyfile.WriteLine("abcdefghijklmnopqrstuvwxyz");

swMyfile.WriteLine("ABCDEFGHIJKLMNOPQRSTUVWXYZ");

swMyfile.WriteLine("0123456789");

// 冲刷数据(把数据真正写到文件中去)

// 注释该句试试看，程序将报错

swMyfile.Flush();

// 以文本方式读文件

// 创建一个数据流读入器，和打开的文件关联

StreamReader srMyfile= new StreamReader(fsMyfile);

// 把文件指针重新定位到文件的开始

srMyfile.BaseStream.Seek(0, SeekOrigin.Begin);

// 打印提示信息

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*以文本方式读文件\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

// 打印文件文本内容

string s1;

while((s1 = srMyfile.ReadLine())!=null)

{

Console.WriteLine(s1);

}

Console.WriteLine();

// 以文本方式读文件结束

// 以二进制方式读文件

// 创建一个二进制数据流读入器，和打开的文件关联

BinaryReader brMyfile= new BinaryReader (fsMyfile);

// 把文件指针重新定位到文件的开始

brMyfile.BaseStream.Seek(0, SeekOrigin.Begin);

// 打印提示信息

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*以二进制方式读文件\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

// 打印文件文本内容

Byte b1;

while(brMyfile.PeekChar()>-1)

{

b1=brMyfile.ReadByte();

// 13为"\n"，表示回车；10为"\r"，表示换行

if(b1 != 13 && b1 != 10)

{

Console.Write("{0}",b1.ToString());

Console.Write(".");

}

else

{

Console.WriteLine();

}

}

Console.WriteLine("\n");

// 以二进制方式读文件结束

// 关闭以上new的各个对象

brMyfile.Close();

srMyfile.Close();

fsMyfile.Close();

// 读取文件属性

// 打印提示信息

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*读取文件属性\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

FileInfo fiMyfile=new FileInfo("myfile.txt");

Console.WriteLine("文件名 : {0}",fiMyfile.Name);

Console.WriteLine("文件名(含路径) : {0}",fiMyfile.FullName);

Console.WriteLine("文件大小(bytes) : {0}",fiMyfile.Length);

Console.WriteLine("文件创建时间 : {0}",fiMyfile.CreationTime);

}

}