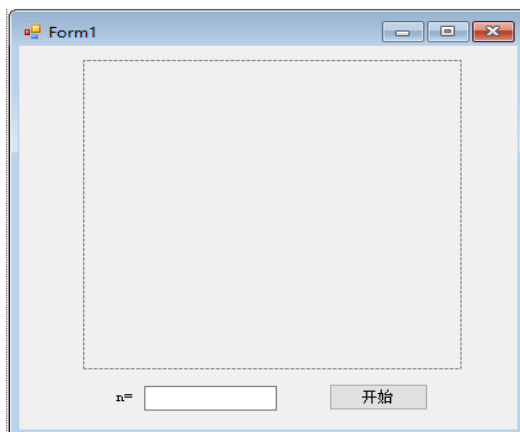


1.实验目的

c#图形界面编程;

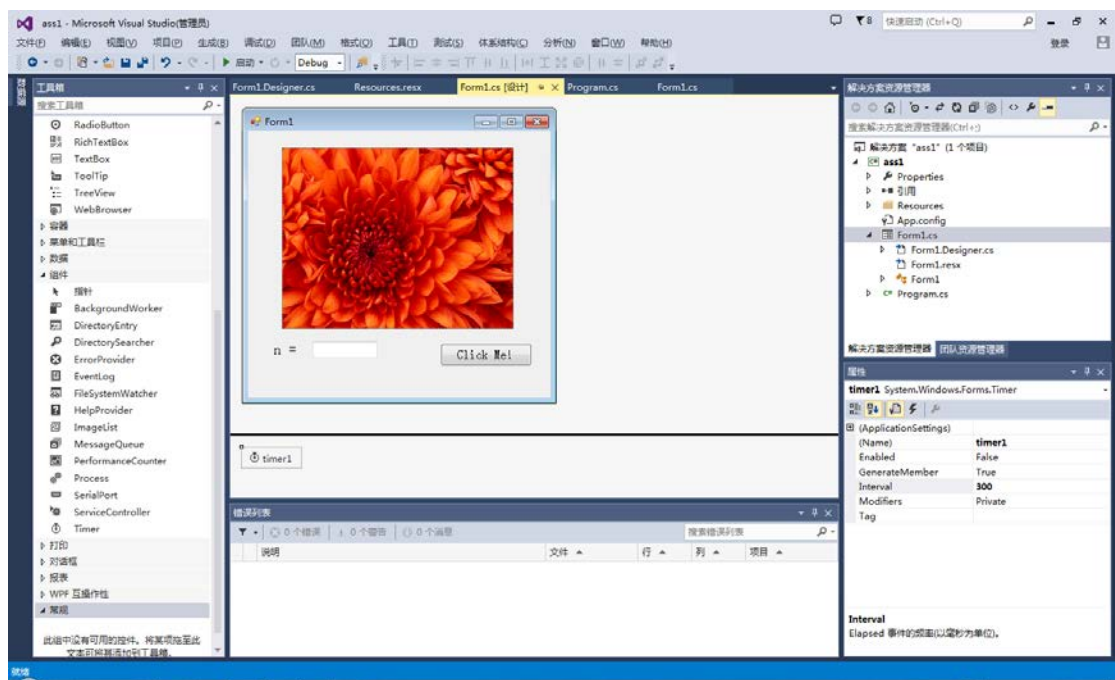
2.实验练习题

- 1: 在图片框中循环显示 n 张图片, 每隔 5 秒显示一张图片。(如下图)
图片有 5 张, 数字 $n \leq 5$ 。填入 n , 点开始, 程序开始运行。



第一步:

建立 windows 窗体程序, 放入控件 pictureBox1, label1, textBox1, button1, 如图所示。修改各控件的 Size 属性, 修改 label1, button1 的 Font, Text 属性, 修改 textBox1 的 Font 属性。修改 pictureBox1 的 SizeMode 为 StretchImage 以完全显示图像。放入控件 timer1 并修改其 Interval 属性



第二步:

添加 **timer1** 与 **button1** 的响应函数，并修改控件主要代码如下：

```
// Form1.cs
using System;
using System.Drawing;
using System.Windows.Forms;

namespace ass1
{
    public partial class Form1 : Form
    {
        private Bitmap[] array;
        private int n, cur;
        public Form1()
        {
            InitializeComponent();
            array = new Bitmap[5];
            array[0] = Properties.Resources.Chrysanthemum;
            array[2] = Properties.Resources.Desert;
            array[3] = Properties.Resources.Hydrangeas;
            array[4] = Properties.Resources.Jellyfish;
            array[1] = Properties.Resources.Koala;
            n = 5; cur = 0;
            timer1.Start();
        }

        private Bitmap NextPic()
        {
            return array[cur == n - 1 ? cur = 0 : ++cur];
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (timer1.Enabled == true)
                timer1.Stop();
            else
                timer1.Start();
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            this.pictureBox1.Image = NextPic();
        }

        private void textBox1_TextChanged(object sender, EventArgs e) {
            try {
```

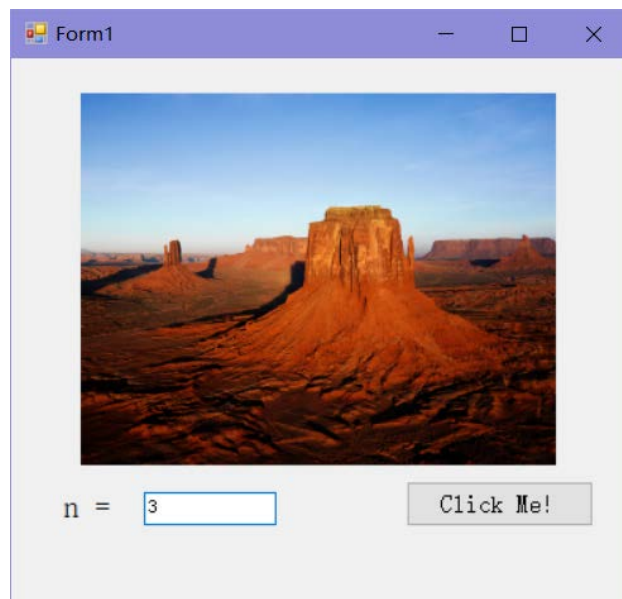
```

        int tmp = Convert.ToInt32(this.textBox1.Text);
        this.n = tmp >= 0 && tmp <= 5 ? tmp : n;
        this.cur = 0;
    } catch (FormatException) {
        // then don't change n
        this.cur = 0;
    }
}
}
}

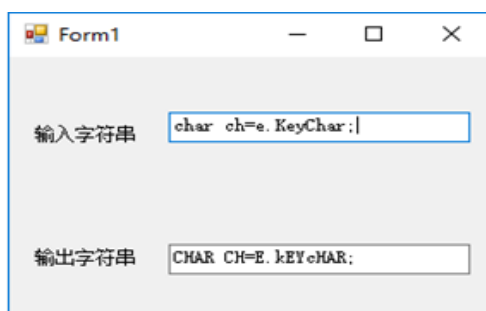
```

达到功能：1. 当文本框内容发生变化时，图像循环的数量发生变化。2. 当按钮被按下时，图像循环开始或停止。

运行截图：



2: 键盘事件处理程序。在文本框 1 中输入字符，同时在文本框 2 中显示其密码字符。加密规则：大小写英文字母互换，其它字符不变。（如下图）



第一步：

建立 Windows 窗体程序，添加两个 Label，两个 TextBox，布局到合适的位置，并调节控件及窗体尺寸；调节字体为微软雅黑，小四；设置 Label 的 AutoSize 为 true。

第二步：

为 Form1 类添加响应 textBox1 文字改变的函数如下：

```

private void textBox1_TextChanged(object sender, EventArgs e) {

```

```

char[] cstr = textBox1.Text.ToCharArray();
for (int i = 0; i < cstr.Length; i++) {
    if (cstr[i] >= 'a' && cstr[i] <= 'z')
        cstr[i] = (char)(cstr[i] - 'a' + 'A');
    else if (cstr[i] >= 'A' && cstr[i] <= 'Z')
        cstr[i] = (char)(cstr[i] - 'A' + 'a');
}
textBox2.Text = new String(cstr);
}

```

运行结果：



3: 设计数字式计时器, 初始界面如图, 点击开始则计时开始。要求: 小时、分、秒能正确转换。

第一步:

建立 WinForm 窗体, 添加两个 Label, 修改 label1 文字、字体、位置。修改 label2 文字、字体为小三、TextAlign 为 MiddleCenter, 背景为青绿色, BorderStyle 为 Fixed3D。

第二步:

在 Form1 类中添加私有字段 time 用于存储从开始到现在的总秒数。在构造函数中初始化为 0。

添加 Timer, 修改 Interval 为 1000。在 Form1 类中添加对于 timer1 Tick 的响应函数如下:

```
private void timer1_Tick(object sender, EventArgs e) {
    time++;
    int hour = time / 3600;
    int minute = (time % 3600) / 60;
    int second = time % 60;
    String str = string.Format("{0:D2}:{1:D2}:{2:D2}", hour, minute, second);
    label2.Text = str;
}
```

第三步:

在 Form1 类中添加对于 button1 点击事件的响应函数, 以控制 timer1 的开启与关闭:

```
private void button1_Click(object sender, EventArgs e) {
    if (timer1.Enabled == false)
        timer1.Start();
    else
        timer1.Stop();
}
```

运行结果:

