```
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 Prog2
{
    public partial class RegForm : Form
    {
        public RegForm()
        {
            InitializeComponent();
        }

        // Precondition:  Find registration time button clicked
        // Postcondition: If last name entered, earliest registration
        //                date/time output. Otherwise, error message displayed
        private void findRegTimeBtn_Click(object sender, EventArgs e)
        {
            const string DAY1 = "March 29";  // 1st day of registration
            const string DAY2 = "March 30";  // 2nd day of registration
            const string DAY3 = "March 31";  // 3rd day of registration
            const string DAY4 = "April 3";   // 4th day of registration
            const string DAY5 = "April 4";   // 5th day of registration
            const string DAY6 = "April 5";   // 6th day of registration

            const string TIME1 = "8:30 AM";  // 1st time block
            const string TIME2 = "10:00 AM"; // 2nd time block
            const string TIME3 = "11:30 AM"; // 3rd time block
            const string TIME4 = "2:00 PM";  // 4th time block
            const string TIME5 = "4:00 PM";  // 5th time block

            // Low end of letter range for juniors and seniors
            char[] juniorSeniorLowLetters = { 'A', 'E', 'J', 'P', 'T' };
            // Times associated with range for juniors and seniors
            string[] juniorSeniorTimes = { TIME3, TIME4, TIME5, TIME1, TIME2 };

            // Low end of letter range for freshmen and sophomores
            char[] freshSophLowLetters = { 'A', 'C', 'E', 'G', 'J',
                                           'M', 'P', 'R', 'T', 'W' };
            // Times associated with range for freshmen and sophomores
            string[] freshSophTimes = { TIME5, TIME1, TIME2, TIME3, TIME4,
                                        TIME5, TIME1, TIME2, TIME3, TIME4 };

            char[] lowLetters; // Will refer to either juniorSeniorLowLetters or
freshSophLowLetters
            string[] times;    // Will refer to either juniorSeniorTimes or
freshSophTimes

            string lastNameStr;       // Entered last name
            char lastNameLetterCh;    // First letter of last name, as char
            string dateStr = "Error"; // Holds date of registration
            string timeStr = "Error"; // Holds time of registration
```

```
bool isUpperClass;        // Upperclass or not?
bool found;               // Range match found?
int index;                // Subscript of array for range match

lastNameStr = lastNameTxt.Text;
if (lastNameStr.Length > 0) // Empty string?
{
    lastNameLetterCh = lastNameStr[0];   // First char of last name
    lastNameLetterCh = char.ToUpper(lastNameLetterCh); // Ensure upper case

    if (char.IsLetter(lastNameLetterCh)) // Is it a letter?
    {
        isUpperClass = (seniorRBtn.Checked || juniorRBtn.Checked);

        // Juniors and Seniors share same schedule but different days
        if (isUpperClass)
        {
            if (seniorRBtn.Checked)
                dateStr = DAY1;
            else // Must be juniors
                dateStr = DAY2;

            // Assign search array vars
            lowLetters = juniorSeniorLowLetters;
            times = juniorSeniorTimes;
        }
        // Sophomores and Freshmen
        else // Must be soph/fresh
        {
            if (sophomoreRBtn.Checked)
            {
                // C-O on one day
                if ((lastNameLetterCh >= 'C') && // >= C and
                    (lastNameLetterCh <= 'O'))   // <= O
                    dateStr = DAY4;
                else // All other letters on previous day
                    dateStr = DAY3;
            }
            else // must be freshman
            {
                // C-O on one day
                if ((lastNameLetterCh >= 'C') && // >= C and
                    (lastNameLetterCh <= 'O'))   // <= O
                    dateStr = DAY6;
                else // All other letters on previous day
                    dateStr = DAY5;
            }

            // Assign search array vars
            lowLetters = freshSophLowLetters;
            times = freshSophTimes;
        }

        // Range match search
        found = false;
        index = lowLetters.Length - 1;   // Start from end
                                         // since lower limits
        while (index >= 0 && !found)
```

```
                    {
                        if (lastNameLetterCh >= lowLetters[index])
                            found = true;
                        else
                            --index;
                    }

                    if (found)
                        timeStr = times[index];

                    // Output results
                    dateTimeLbl.Text = dateStr + " at " + timeStr;
                }
                else // First char not a letter
                    MessageBox.Show("Make sure last name starts with a letter");
            }
            else // Empty textbox
                MessageBox.Show("Enter a last name!");
        }
    }
}
```