CISP 41

Programming in C#

Project Evaluation Sheet

Student Name: ____________ Bryant Tunbutr ______________________ Project Number: _______4________

Project Name: _______BtunbutrProject4_____ Visual Studio Version:___2008________

Date Due: ________11/22/12_____________ Date Turned In: _____11/22/12_____

Above to be completed by student

Points (40 Possible)

Correctness/Efficiency:

Output is accurate	Show when not found	-1
Meets all requirements	Missing file input	-5
Provide appropriate user interface		___________
Logic is efficient		___________

Documentation/Coding Style:

Project can be open from the submitted zip file	___________
Folder is present and contains all necessary project files (no extra files)	___________
Use required coding template	___________
Use proper naming and spacing	___________
Submit all requested information	___________

Test Cases:

List all required test cases	___________
Provide output forms for important test cases	___________

Other issues: ___________

Extra Credit: ___________

Timeliness: -3

Project Score: 31/40

CISP 41 Programming in C#

Project specification

This software is intended to give the spark plug number that is equivalent to other brands.

It is designed to be run in Visual Studio 2008 using the C# coding.

It uses user input including the selected brand and number of the part.

It displays CCC brand number.

Used in the project are arrays, 2-dimensional arrays, methods, list boxes.

CISP 41 **Programming in C#**

Project status

Can continue to work on it until Thursday.

The project is not complete.

I tried for many hours to get the text file into an array of structure.

I was successful in loading the file into string arrays using this code

```
            //Load text file
            TextReader tr = new StreamReader("C:\\Users\\bryantt\\Desktop\\project 4 3 
test\\WindowsFormsApplication1\\bin\\Debug\\test.txt");

            //lines loaded
            int NumberOfLines = 5;

            //array for each line
            string[] ListLines = new string[NumberOfLines];

            //Read the number of lines and put them in the array
            for (int i = 1; i < NumberOfLines; i++)
            {
                ListLines[i] = tr.ReadLine();
            }

            //Make array for each line
            string[] ccc = ListLines[1].Split(new Char[] { ',' });
            string[] brandsA = ListLines[2].Split(new Char[] { ',' });
            string[] brandsB = ListLines[3].Split(new Char[] { ',' });
            string[] brandsC = ListLines[4].Split(new Char[] { ',' });

            // close the stream
            tr.Close();
```

You don't really need 2-D array, but you can convert these 4 arrays to 2-D array.

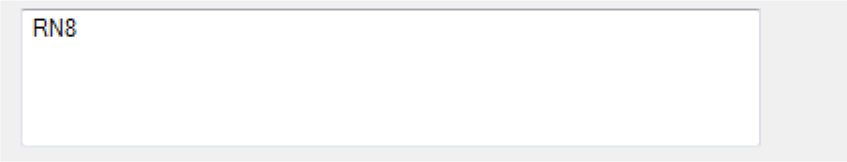
However I was never able to get these string arrays to convert into a 2-dimensional array structure.

I tried parsing methods, I tried integer and character and string conversion without success. I also tried making using loops to create 2-dimensional arrays without success.

I also tried just using these arrays but was unable to do so successfully

I do know that these arrays worked in exporting the data from the text file because when used code like the following, I got successful results

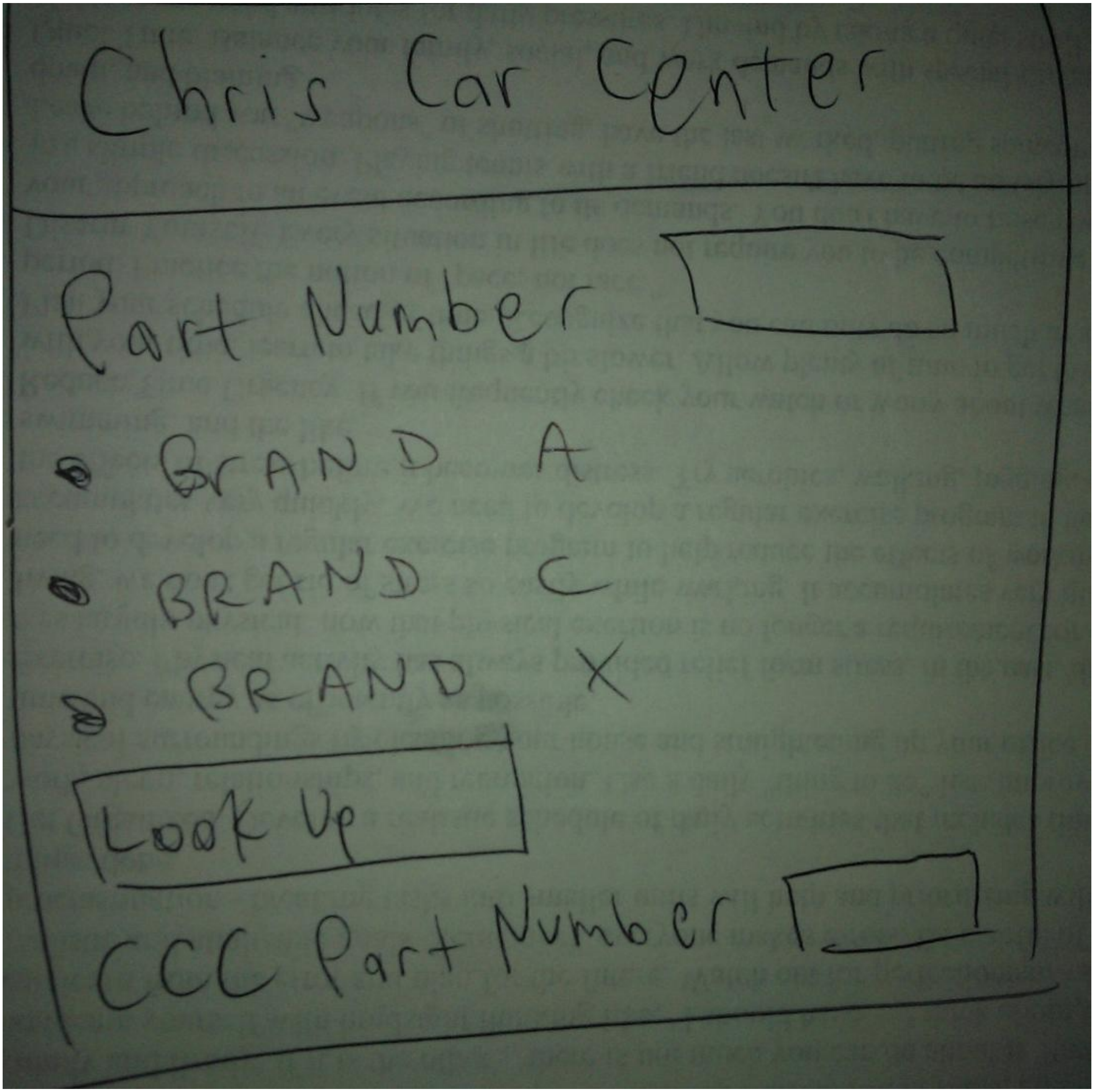
```
            textBox1.Text = brandsB[3].ToString();
```



```
RN8
```

I feel that the project is 95% complete, I am just missing one step of converting these string arrays into the 2-dimensional arrays that I used in the finalized project

Sketch of user interface



CISP 41 Programming in C#

Objects and Properties Plan for __ BTunbutrProject 4_________Form

Object	Property	Setting
Look Up Button	Name Text	**Method**
Look Up	Name Text	**Button Click**
If loop	Text	**Method**
label	Name Text	**Part Number**
Combo text box	Text	**Drop down**
BTunbutrProject4	Name Text	**Form**
brandsString	Text	**array**
cccString	Text	**array**

CISP 41 **Programming in C#**

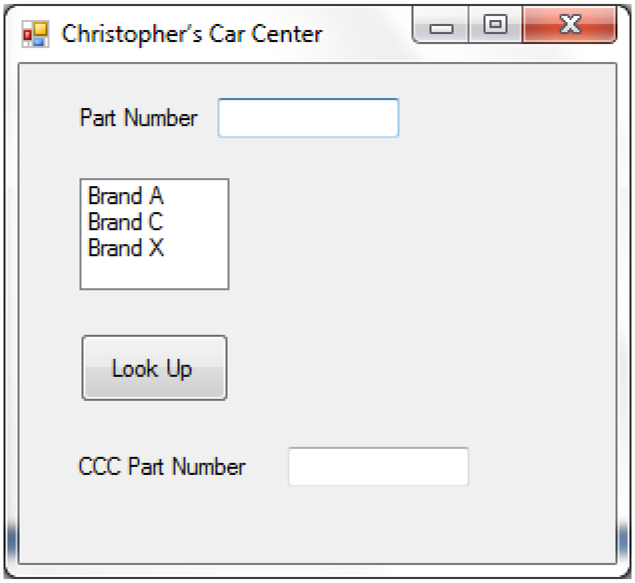
Event Plan for ___ BTunbutrProject 4________Form

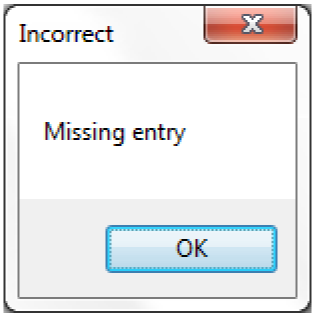
Object	**Event**	**Action - Pseudocode**
Look Up Button	Click	Call Look Up Button_Click. Display the part number. Run methods with arrays
Part number	Click	Use in method to match
List box	Click	Select array
ccc number	Click	Display method results

CISP 41 Programming in C#

Test cases and captured screens

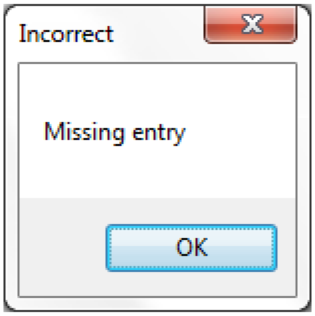
Test case #1
Nothing selected




Incorrect

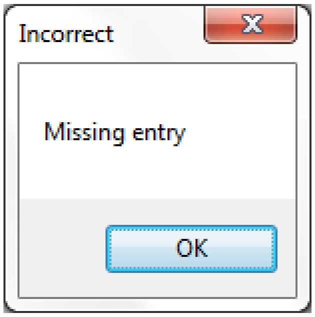
Missing entry

Test case #2a
Only a part number is entered


Incorrect

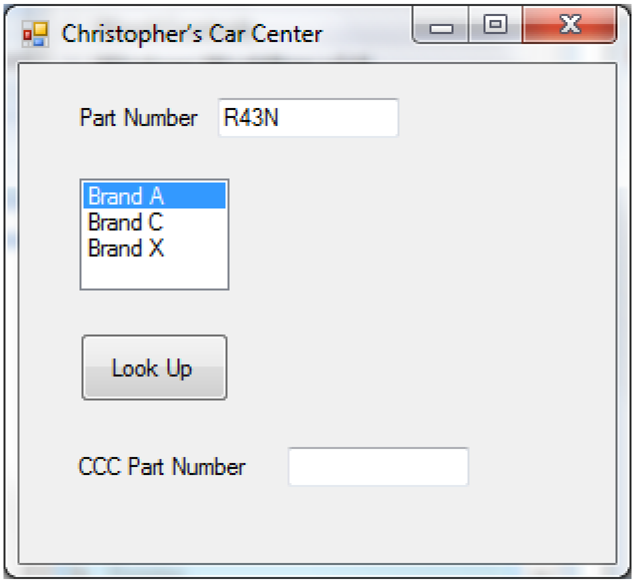
Missing entry

Test case #2b
Only a brand is selected


Incorrect

Missing entry

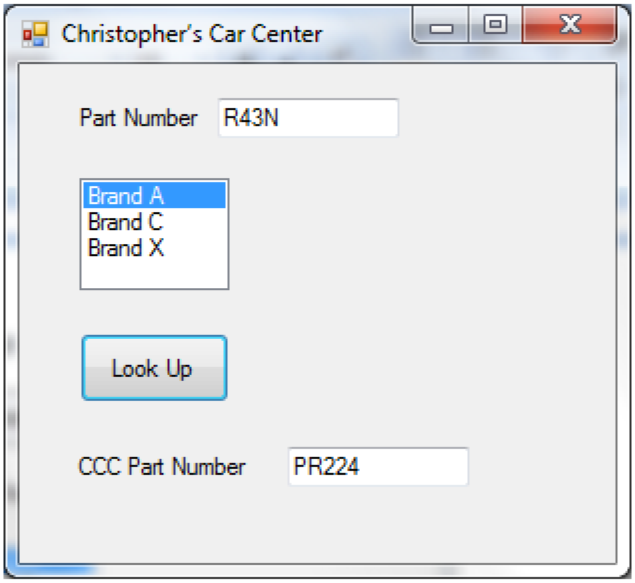
Test case #3



Part Number R43N

Brand A
Brand C
Brand X

CCC Part Number



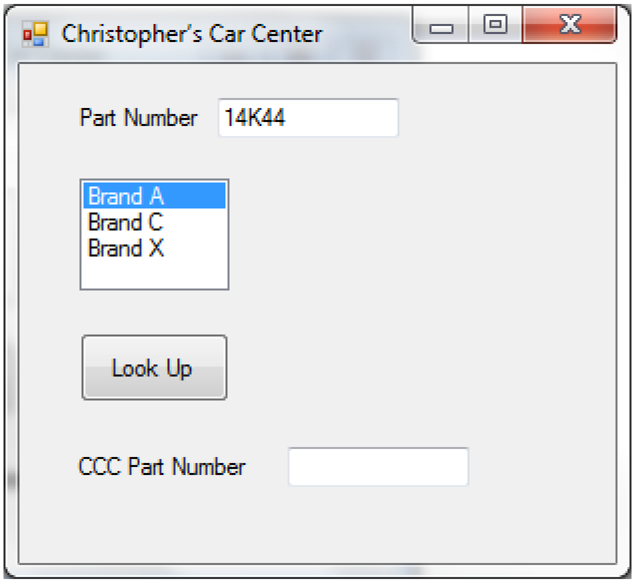
Part Number R43N

Brand A
Brand C
Brand X

CCC Part Number PR224

Test case #4

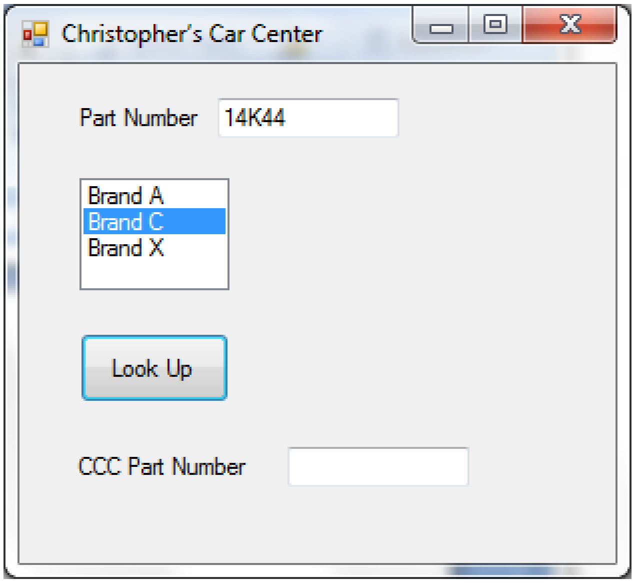
Part #14K44 with each of the brands selected



Part Number 14K44

Brand A

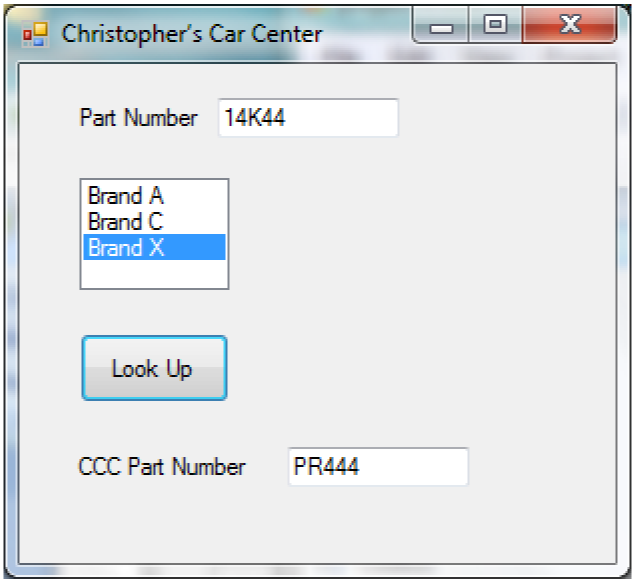
CCC Part Number



Part Number 14K44

Brand C

CCC Part Number



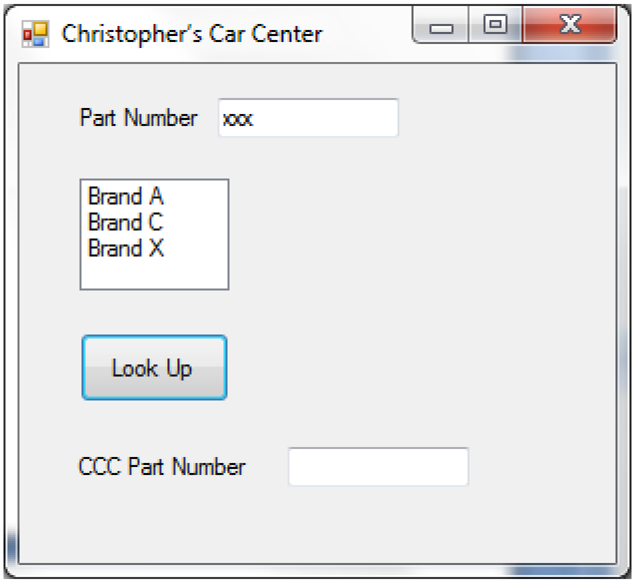
Part Number 14K44

Brand X

CCC Part Number PR444

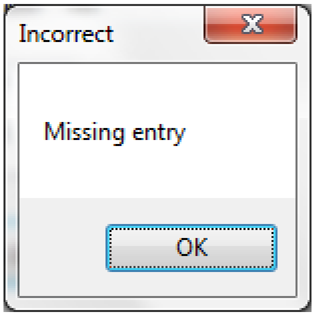
Test case #5

A non existing part number is entered



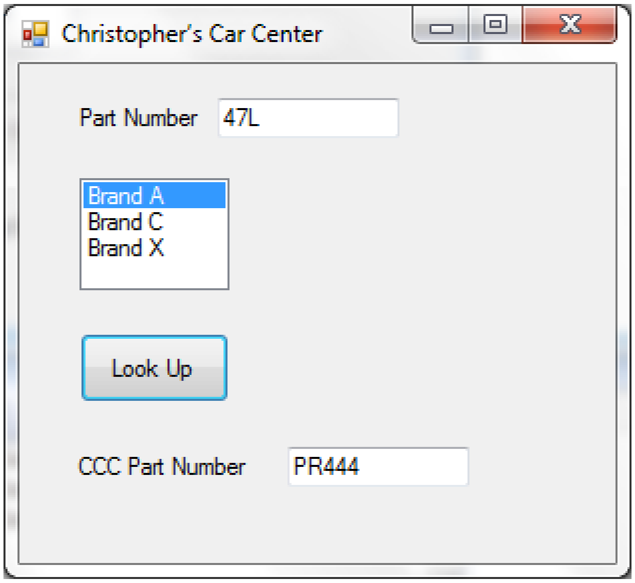
Part Number xxx

CCC Part Number



Missing entry

Test case #6

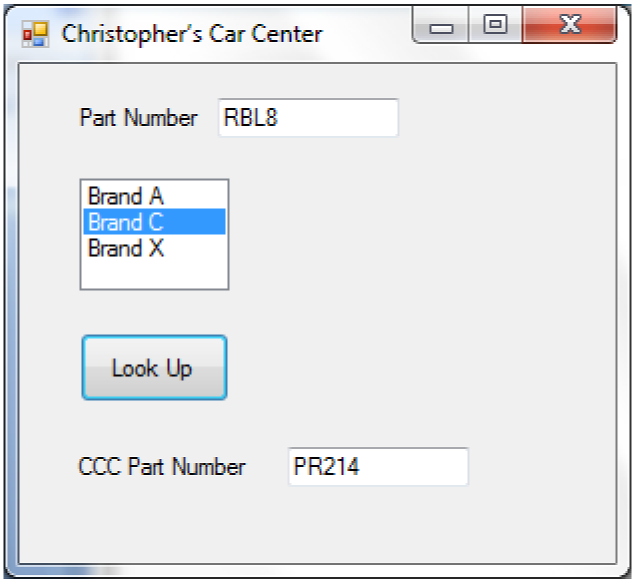


Part Number 47L

Brand A

CCC Part Number PR444

Test case #7

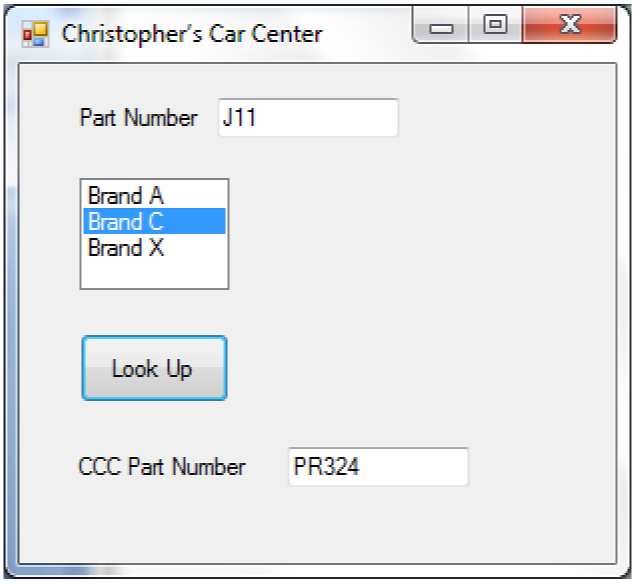


Part Number RBL8

Brand A
Brand C
Brand X

CCC Part Number PR214

Test case #8



Part Number J11

Brand A
Brand C
Brand X

CCC Part Number PR324

Source code

```
/*
* Project: BtunbutrProject4
* Programmer: Bryant Tunbutr
* Date: Nov 22 2012
* Description: Uses array of structures to match part number with brand number
 * I certify that the code below is my own work.
*/

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;
using System.IO;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        //Create arrays, 2 dimensional for brands
        string[,] brandsString = new string[9, 3];
        string[] cccString = new string[9];

        public Form1()
        {
            InitializeComponent();
        }

        private void lookUpButton_Click(object sender, EventArgs e)
        {
            //Organize by having each brand have same second integer,
            //i.e. Brand A has second integer of 0

            //Brand A
            brandsString[0, 0] = "MR43T";
            brandsString[1, 0] = "R43";
            brandsString[2, 0] = "R43N";
            brandsString[3, 0] = "R46N";
            brandsString[4, 0] = "R46TS";
            brandsString[5, 0] = "R46TX";
            brandsString[6, 0] = "S46";
            brandsString[7, 0] = "SR46E";
            brandsString[8, 0] = "47L";

            //Brand C
            brandsString[0, 1] = "RBL8";
            brandsString[1, 1] = "RJ6";
            brandsString[2, 1] = "RN4";
            brandsString[3, 1] = "RN8";
            brandsString[4, 1] = "RBL17Y";
            brandsString[5, 1] = "RBL12-6";
            brandsString[6, 1] = "J11";
            brandsString[7, 1] = "XEJ8";
            brandsString[8, 1] = "H12";

            //Brand X
            brandsString[0, 2] = "14K22";
            brandsString[1, 2] = "14K24";
            brandsString[2, 2] = "14K30";
            brandsString[3, 2] = "14K32";
            brandsString[4, 2] = "14K33";
            brandsString[5, 2] = "14K35";
            brandsString[6, 2] = "14K38";
            brandsString[7, 2] = "14K40";
            brandsString[8, 2] = "14K44";

            //CCC NUMBER
            cccString[0] = "PR214";
            cccString[1] = "PR223";
            cccString[2] = "PR224";
            cccString[3] = "PR246";
            cccString[4] = "PR247";
            cccString[5] = "PR248";
            cccString[6] = "PR324";
            cccString[7] = "PR326";
            cccString[8] = "PR444";

            int brandInteger = brandListBox.SelectedIndex;
            int lengthInteger = 9;

            //make sure user has selected and entered data
            if (partNumberTextBox.Text != "" && brandInteger != -1)

                for (int indexInteger = 0; indexInteger < lengthInteger; indexInteger++)
                {
                        //match part number with brand number.
                        if (partNumberTextBox.Text == brandsString[indexInteger, 
brandInteger])
                        {
                                //display result.
                                cccPartNumberTextBox.Text = cccString[indexInteger];
                        }
                }

            //check for exeption.
            else
            {
                MessageBox.Show("Missing entry", "Incorrect");
            }
        }

    }
}
```

Should indicate that part not found
if applicable.