**CSC 1101 – Problem Solving and Programming Laboratory**

**Lab 20 – rory lange**

**25 points – Due April 12, 11pm**

**a)** Save this document with your name and the lab assignment number somewhere in the file name.

**b)** Type/paste your answers into the document.

**c)** Submit the following documents to the Canvas assignment link where you downloaded this document:

✓ This document.

✓ Your .cpp files renamed to .txt.

Submit the documents separately, not as one .zip file.

You've been hired by *Part Pretenders* to write a C++ console application that manages car parts. Declare string array **parts** of size eight. Use an initializer list to store eight car parts in the array. Use one word per part. Don't store the parts in alphabetical order. Create the following functions:

**int binarySearchParts(string parts[], int arraySize, string key)**

This value function takes the parts array, its size, and a key, and does a binary search on the array. If the key is found, it returns its index in the array. If the key is not found, it returns -1.

**void listParts(string parts[], int arraySize)**

This void function prints a list header, loops through the **parts** array and prints each part, one per line, and prints the number of parts.

**void searchParts(string parts[], int arraySize)**

This void function prompts for and gets from the user a part. If you initialized the parts array correctly, none of the part names contains spaces. It then calls function **binarySearchParts** and tests the result. If the part name was found, it prints a "part found at index N" message. Otherwise, it prints a "part not found" message.

**void sortParts(string parts[], int arraySize)**

This void function sorts the **parts** array. You may use a selection or insertion sort. It then prints a "data sorted" message when the sort completes.

**int menuOption()**

This value function presents the following menu to the user:

Part Pretenders Menu

1 - List car parts

2 - Sort car parts

3 - Search car parts

9 – Exit

Enter an option:

It then prompts for and gets from the user an option, and returns it. Here are the option descriptions:

**1 – List car parts** calls function listParts.

**2 – Sort car parts** calls function sortParts.

**3 – Search car parts** calls function searchParts.

Declare a global constant for the array size. Continue to read and process menu options until the user enters the sentinel value of 9. The output should look like:

Welcome to Part Pretenders

--------------------------

Part Pretenders Menu

1 - List car parts

2 - Sort car parts

3 - Search car parts

9 - Exit

Enter an option: 1

Part List

…

Part count: 8

Part Pretenders Menu

1 - List car parts

2 - Sort car parts

3 - Search car parts

9 - Exit

Enter an option: 2

Parts sorted.

Part Pretenders Menu

1 - List car parts

2 - Sort car parts

3 - Search car parts

9 - Exit

Enter an option: 3

Enter a part to search for (no spaces): Tire

'Tire' found at index 5.

Part Pretenders Menu

1 - List car parts

2 - Sort car parts

3 - Search car parts

9 - Exit

Enter an option: 9

End of Part Pretenders

Run the program using the following menu option order:

1

2

1

3 with part in list

3 with part not in list

//==========================================================

//

// Title:      Part Pretenders

// Course:     CSC 1101

// Lab Number: 20-1

// Author:     rory lange

// Date:       4/8/21

// Description:

//   This C++ console application manages part names.  It

// declares string array parts of size eight, and stores

// eight parts in it.  It presents the following menu to the

// user:

//

//   Part Pretenders Menu

//   1 - List car parts

//   2 - Sort car parts

//   3 - Search car parts

//   9 � Exit

//

//   Enter an option :

//

// It continues to read and process menu options until the

// user enters the sentinel value of 9.

//

//==========================================================

#include <cstdlib> // For several general-purpose functions

#include <fstream> // For file handling

#include <iomanip> // For formatted output

#include <iostream> // For cin, cout, and system

#include <string> // For string data type

using namespace std; // So "std::cout" may be abbreviated to "cout"

//==========================================================

// Globals

//==========================================================

const int ArrSize = 8;

string parts[] = {"transmission", "engine", "electronics", "doors", "windows", "wheels", "steering", "brakes"};

//==========================================================

// binarySearchParts

//==========================================================

int binarySearchParts(string parts[], int arraySize, string key) {

    //sortParts(parts, arraySize);

    int min = 0;

    int index;

    int max = arraySize - 1;

    while (min <= max) {

        //midpoint

        index = (min + max) / 2;

        if (parts[index] == key)

            return index;

        else if (parts[index] < key)

            min = index + 1;

        else

            max = index - 1;

    }

    return -1;

}

//==========================================================

// menuOption

//==========================================================

int menuOption() {

    int input;

    cout << "Part Pretenders Menu" << endl;

    cout << "1 - List car parts" << endl;

    cout << "2 - Sort car parts" << endl;

    cout << "3 - Search car parts" << endl;

    cout << "9 - Exit" << endl << endl;

    cout << "Enter an option: ";

    cin >> input;

    return input;

}

//==========================================================

// listParts

//==========================================================

void listParts(string parts[], int arraySize) {

    cout << "Part List" << endl << endl;

    for (int i = 0; i < arraySize; i++) {

        cout << parts[i] << endl;

    }

    cout << endl << "Part count: " << arraySize << endl << endl;

}

//==========================================================

// searchParts

//==========================================================

void searchParts(string parts[], int arraySize) {

    string input;

    int index = 0;

    cout << "Enter a part to search for (no spaces): ";

    cin >> input;

    index = binarySearchParts(parts, 8, input);

    if (index == -1)

        cout << "'" << input << "'" << " was not found int the array." << endl << endl;

    else

        cout << "'" << input << "'" << " was found at index " << index << "." << endl << endl;

}

//==========================================================

// sortParts

//==========================================================

void sortParts(string parts[], int arraySize) {

    string temp;

    int j;

    string key;

    for (int i = 1; i < arraySize; i++) {

        j = i - 1;

        key = parts[i];

        while (j >= 0 && parts[j] > key) {

            parts[j + 1] = parts[j];

            j = j - 1;

        }

        parts[j + 1] = key;

    }

}

//==========================================================

// main

//==========================================================

int main() {

    //header

    cout << "Welcome to Part Pretenders" << endl << endl;

    //variables

    int input = menuOption();

    //sentinel

    while (input != 9) {

        switch(input) {

            case 1:

                listParts(parts, ArrSize);

                break;

            case 2:

                sortParts(parts, ArrSize);

                break;

            case 3:

                searchParts(parts, ArrSize);

                break;

        }

        input = menuOption();

    }

    //exit

    cout << "END OF PART PRETENDERS" << endl;

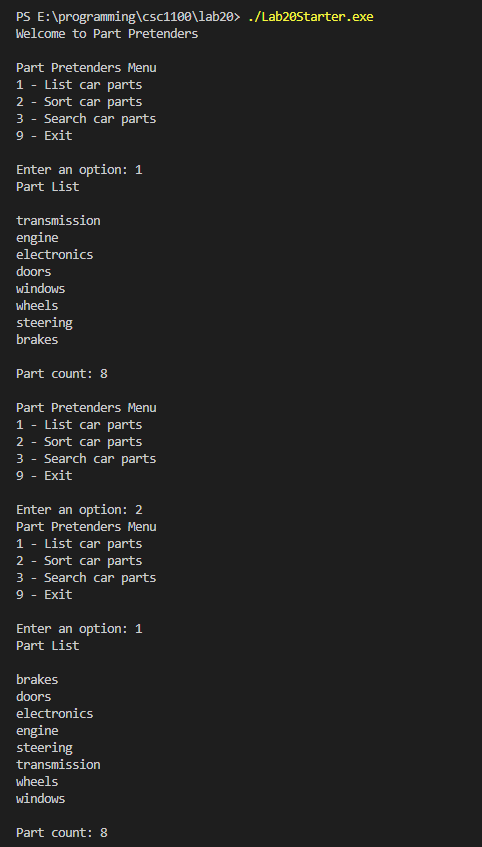
}

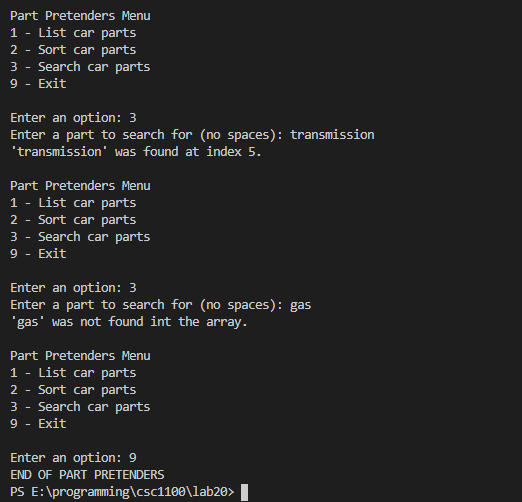
**If possible, format your code like this:**

**Font “Courier New”**

**Font size “9”s**

**Bold**





**\* Copying-and-pasting C++ code to a Word document**

**macOS**

1) From within the C++ program, press **command-A** and press **command-C**.

2) From within the Word document, press **command-V**.

**Windows**

1) From within the C++ program, press **CTRL-A** and press **CTRL-C**.

2) From within the Word document, press **CTRL-V**.

**\*\* Copying-and-pasting C++ console application output to a Word document**

**macOS**

1) From the C++ console, press **shift-command-4-space**.

2) From within the Word document, **command-V**.

**Windows**

1) From the C++ console, press **ALT-PrintScreen**.

2) From within the Word document, press **CTRL-V**.