**CSC 1100 – Problem Solving and Programming**

**Project 2 – rory lange**

**50 points – Due April 29, 11am**

**Late deadline is May 1, 11:59pm, but 20% off**

**a)** Save this document with your name and the project 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 again by *Part Pretenders* to enhance the great C++ console application you wrote for Labs 20 and 22. Expand the string array **parts** size from ten to 100 but remove the initializer list. The parts will now come from an input file. Create text file **CarParts.txt** and place at least fifteen car parts in array **parts**. The file contains no header line. Each detail line contains one car part. Each part name contains no spaces (you can use underscore to separate words within a part name). The part names should not be sorted within the file. Place the file in a folder where your development tool can locate it, or place the file in a data folder and use a path to the file. Create or modify the following functions:

**void addPart(string parts[], int &partCount)**

After a part is added to array **parts**, sort its data. This ensures that the data is always in alphabetical order.

**void deletePart(string parts[], int &partCount)**

Add logic to prompt the user for the index of the car part to delete. If the index is invalid (<0 or >= partCount), print an error message. If the index is valid, remove the part from array **parts**. See the Homework 6 key for code to do this. Change the message to indicate "part at index removed".

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

Use formatted output manipulators (setw, left/right) to print in one row the following:

● Index

● Part name

**int menuOption()**

Change the menu to:

Part Pretenders Menu

1 - List car parts

2 - Search car parts

3 - Add car part

4 - Delete car part

9 - Exit

Enter an option: ";

Ensure the correct function is called:

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

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

**3 – Add part** calls function addPart.

**4 – Delete part** calls function deletePart.

**int readParts(string parts[])**

This value function attempts to open the input file. If it doesn't open, return a negative number. If it does open, print a file-reading message, loop through the file and store each part read in array **parts**, print the number of lines read, and return the line count.

**int main()**

Call function readParts and test the returned value. If the file didn't open, print an error message. If the file opened, sort array parts, and continue to read and process menu options until the user enters the sentinel value of 9.

Define constants for the input file name, array size, and the column widths. Insure that your code adheres to the styles as defined in document **Source code styles** on Canvas. Provide a complete header comment and body comments. Arrange the programmer-defined functions in alphabetical order and place function **main** at the end. The output should look like:

Welcome to Part Pretenders

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

Reading lines from file 'CarParts.txt' ...

15 line(s) read from file 'CarParts.txt'.

Parts sorted.

Part Pretenders Menu

1 - List car parts

2 - Search car parts

3 - Add car part

4 - Delete car part

9 - Exit

Enter an option: 1

Part List

Index Part

0 Air\_bag

…

14 Transmission

Part count: 15

Part Pretenders Menu

1 - List car parts

2 - Search car parts

3 - Add car part

4 - Delete car part

9 - Exit

Enter an option: 2

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

'Gearshift' NOT found.

Part Pretenders Menu

1 - List car parts

2 - Search car parts

3 - Add car part

4 - Delete car part

9 - Exit

Enter an option: 3

Enter the part to add: Gearshift

Part 'Gearshift' added to list.

Parts sorted.

Part Pretenders Menu

1 - List car parts

2 - Search car parts

3 - Add car part

4 - Delete car part

9 - Exit

Enter an option: 1

Part List

Index Part

0 Air\_bag

..

9 Gearshift

…

12 Rear\_bumper

13 Side\_mirror

14 Tire

15 Transmission

Part count: 16

Part Pretenders Menu

1 - List car parts

2 - Search car parts

3 - Add car part

4 - Delete car part

9 - Exit

Enter an option: 4

Enter index of part to delete: 22

Error: invalid index of 22.

Part Pretenders Menu

1 - List car parts

2 - Search car parts

3 - Add car part

4 - Delete car part

9 - Exit

Enter an option: 4

Enter index of part to delete: 13

Part at index 13 deleted from list.

Part Pretenders Menu

1 - List car parts

2 - Search car parts

3 - Add car part

4 - Delete car part

9 - Exit

Enter an option: 1

Part List

Index Part

0 Air\_bag

…

12 Rear\_bumper

13 Tire

14 Transmission

Part count: 15

Part Pretenders Menu

1 - List car parts

2 - Search car parts

3 - Add car part

4 - Delete car part

9 - Exit

Enter an option: 9

End of Part Pretenders

Run the program using the following menu option order:

1

2

3

1

3

1

4

1

4

1

2

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

//

// Title:      Part Pretenders, v2

// Course:     CSC 1100

// Project:    2

// Author:     Rory Lange

// Date:       4/27/21

// Description:

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

//   reads parts from a .txt file and puts them in a list

//   the app then proceeds to do all the functions in the

//   menu.

//

//   Part Pretenders Menu

//   1 - List car parts

//   2 - Sort car parts

//   3 - Search car parts

//   4 - Add car part

//   5 - Delete car part

//   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 ARRAY\_SIZE = 100;

const string FileName = "CarParts.txt";

const int W = 15;

const int W1 = 20;

//function stubs

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

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

// addPart

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

void addPart(string parts[], int &partCount)

{

  // Declare variables

  string part;

  // Test whether room to add part

  if (partCount == ARRAY\_SIZE)

    cout << "No room to add part at this time." << endl;

  else

  {

    // Prompt for and get part

    cout << "Enter the part to add: ";

    cin >> part;

    // Add part

    parts[partCount] = part;

    partCount = partCount + 1;

    cout << "Part '" << part << "' added to list."

      << endl;

    sortParts(parts, ARRAY\_SIZE); //sort after part is added

  }

}

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

// binarySearchParts

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

int binarySearchParts(

  string parts[], int arraySize, string key)

{

  // Declare variables

  int min = 0;

  int index;

  int max = arraySize - 1;

  // Loop to find key

  while (min <= max)

  {

    // Get midpoint of array

    index = (min + max) / 2;

    // Test if key found

    if (parts[index] == key)  // Guess is right on

      return index;

    else if (parts[index] < key)  // Guess is too low

      min = index + 1;

    else

      max = index - 1;

  }

  // Return index

  return -1;

}

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

// deletePart

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

void deletePart(string parts[], int &partCount)

{

  int index;

  string temp;

  cout << "Enter index of part to delete: ";

  cin >> index;

  // Test whether room to add part

  if (index < 0 || index >= partCount)

    cout << "Error: invalid index" << endl;

  else

  {

    // Delete part

    cout << "Part at index " << index << " removed." << endl;

    while (index < partCount) {

      temp = parts[index+1];

      parts[index] = temp;

      index++;

    }

    partCount = partCount - 1;

  }

}

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

// listParts

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

void listParts(string parts[], int arraySize)

{

  // Loop to list parts

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

  cout << left << setw(W) << "Index" << left << setw(W1) << "Part" << endl;

  for (int i = 0; i < arraySize; i++)

    cout << left << setw(W) << i << left << setw(W1) << parts[i] << endl;

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

}

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

// menuOption

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

int menuOption()

{

  // Declare variables

  int option;

  // Show menu and get option

  cout << "\nPart Pretenders Menu" << endl;

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

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

  cout << "3 - Add car part" << endl;

  cout << "4 - Delete car part" << endl;

  cout << "9 - Exit" << endl;

  cout << "\nEnter an option: ";

  cin >> option;

  return option;

}

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

// readParts

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

int readParts(string parts[]) {

  ifstream inputFile;

  string part;

  int count = 0;

  inputFile.open(FileName); //open file

  if (!inputFile.is\_open()) //check if its good

    return -1;

  else {

    cout << "reading lines from file '" << FileName << "'" << endl;

    while (inputFile.good()) { //add parts to file

      getline(inputFile, part);

      parts[count] = part;

      count++; //iterate count

    }

    cout << "read " << count << " parts from file '" << FileName << "'" << endl;

    inputFile.close(); //close file

    sortParts(parts, ARRAY\_SIZE); //sort parts after reading from .txt

    return count;

  }

}

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

// searchParts

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

void searchParts(string parts[], int arraySize)

{

  // Declare variables

  string key;

  int index;

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

  cin >> key;

  index = binarySearchParts(parts, arraySize, key);

  if (index != -1)

    cout << "'" << key << "' found at index "

    << index << "." << endl;

  else

    cout << "'" << key << "' NOT found." << endl;

}

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

// sortParts

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

void sortParts(string parts[], int arraySize)

{

  // Declare variables

  string value;

  int spot;

  // Loop to test each value

  for (int i = 1; i < arraySize; i++)

  {

    // Loop to find spot to place value

    value = parts[i];

    if (value == ""); //if spot is empty then skip

    else { //spot isnt empty keep going

      spot = i - 1;

      while (spot >= 0 && parts[spot] > value)

      {

        parts[spot + 1] = parts[spot];

        spot = spot - 1;

      }

      // Place value in spot

      parts[spot + 1] = value;

    }

  }

  // Print sorted message

  cout << endl << "Parts sorted." << endl;

}

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

// main

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

int main()

{

  // Declare variables

  string parts[ARRAY\_SIZE];

  int partCount = 0;

  int option;

  // Show application header

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

  cout << "--------------------------" << endl;

  partCount = readParts(parts);

  if (partCount == -1)

    cout << "Error: could not open file." << endl;

  else {

    // Loop to process options

    option = menuOption();

    while (option != 9)

    {

      // Handle option

      switch (option)

      {

        // List parts

      case 1:

        listParts(parts, partCount);

        break;

        // Sort parts

      // case 2:

      //   sortParts(parts, partCount);

      //   break;

        // Search parts

      case 2:

        searchParts(parts, partCount);

        break;

        // Add part

      case 3:

        addPart(parts, partCount);

        break;

        // Delete part

      case 4:

        deletePart(parts, partCount);

        break;

        // Handle invalid option

      default:

        cout << endl << "Error: unknown option of '" << option

          << "'." << endl;

      }

      // Get next option

      option = menuOption();

    }

  }

  // Show application close

  cout << "\nEnd of Part Pretenders" << endl;

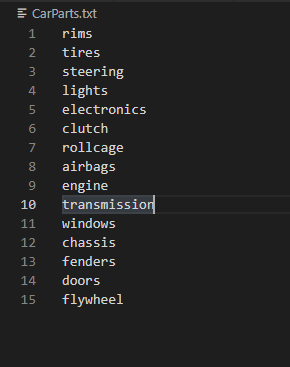
}

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

**Font “Courier New”**

**Font size “9”**

**Bold**

**

***Windows PowerShell***

***Copyright (C) Microsoft Corporation. All rights reserved.***

***Try the new cross-platform PowerShell https://aka.ms/pscore6***

***PS C:\Users\rorys\OneDrive - Wayne State University\CSC1100-C++\project2> ./main.exe***

***Welcome to Part Pretenders***

***--------------------------***

***reading lines from file 'CarParts.txt'***

***read 15 parts from file 'CarParts.txt'***

***Parts sorted.***

***Part Pretenders Menu***

***1 - List car parts***

***2 - Search car parts***

***3 - Add car part***

***4 - Delete car part***

***9 - Exit***

***Enter an option: 1***

***Part List***

***Index Part***

***0 airbags***

***1 chassis***

***2 clutch***

***3 doors***

***4 electronics***

***5 engine***

***6 fenders***

***7 flywheel***

***8 lights***

***9 rims***

***10 rollcage***

***11 steering***

***12 tires***

***13 transmission***

***14 windows***

***Part count: 15***

***Part Pretenders Menu***

***1 - List car parts***

***2 - Search car parts***

***3 - Add car part***

***4 - Delete car part***

***9 - Exit***

***Enter an option: 2***

***Enter a part to search for (no spaces): electronics***

***'electronics' found at index 4.***

***Part Pretenders Menu***

***1 - List car parts***

***2 - Search car parts***

***3 - Add car part***

***4 - Delete car part***

***9 - Exit***

***Enter an option: 3***

***Enter the part to add: exhaust***

***Part 'exhaust' added to list.***

***Parts sorted.***

***Part Pretenders Menu***

***1 - List car parts***

***2 - Search car parts***

***3 - Add car part***

***4 - Delete car part***

***9 - Exit***

***Enter an option: 1***

***Part List***

***Index Part***

***0 airbags***

***1 chassis***

***2 clutch***

***3 doors***

***4 electronics***

***5 engine***

***6 exhaust***

***7 fenders***

***8 flywheel***

***9 lights***

***10 rims***

***11 rollcage***

***12 steering***

***13 tires***

***14 transmission***

***15 windows***

***Part count: 16***

***Part Pretenders Menu***

***1 - List car parts***

***2 - Search car parts***

***3 - Add car part***

***4 - Delete car part***

***9 - Exit***

***Enter an option: 3***

***Enter the part to add: blinkers***

***Part 'blinkers' added to list.***

***Parts sorted.***

***Part Pretenders Menu***

***1 - List car parts***

***2 - Search car parts***

***3 - Add car part***

***4 - Delete car part***

***9 - Exit***

***Enter an option: 1***

***Part List***

***Index Part***

***0 airbags***

***1 blinkers***

***2 chassis***

***3 clutch***

***4 doors***

***5 electronics***

***6 engine***

***7 exhaust***

***8 fenders***

***9 flywheel***

***10 lights***

***11 rims***

***12 rollcage***

***13 steering***

***14 tires***

***15 transmission***

***16 windows***

***Part count: 17***

***Part Pretenders Menu***

***1 - List car parts***

***2 - Search car parts***

***3 - Add car part***

***4 - Delete car part***

***9 - Exit***

***Enter an option: 4***

***Enter index of part to delete: 5***

***Part at index 5 removed.***

***Part Pretenders Menu***

***1 - List car parts***

***2 - Search car parts***

***3 - Add car part***

***4 - Delete car part***

***9 - Exit***

***Enter an option: 1***

***Part List***

***Index Part***

***0 airbags***

***1 blinkers***

***2 chassis***

***3 clutch***

***4 doors***

***5 engine***

***6 exhaust***

***7 fenders***

***8 flywheel***

***9 lights***

***10 rims***

***11 rollcage***

***12 steering***

***13 tires***

***14 transmission***

***15 windows***

***Part count: 16***

***Part Pretenders Menu***

***1 - List car parts***

***2 - Search car parts***

***3 - Add car part***

***4 - Delete car part***

***9 - Exit***

***Enter an option: 4***

***Enter index of part to delete: 7***

***Part at index 7 removed.***

***Part Pretenders Menu***

***1 - List car parts***

***2 - Search car parts***

***3 - Add car part***

***4 - Delete car part***

***9 - Exit***

***Enter an option: 1***

***Part List***

***Index Part***

***0 airbags***

***1 blinkers***

***2 chassis***

***3 clutch***

***4 doors***

***5 engine***

***6 exhaust***

***7 flywheel***

***8 lights***

***9 rims***

***10 rollcage***

***11 steering***

***12 tires***

***13 transmission***

***14 windows***

***Part count: 15***

***Part Pretenders Menu***

***1 - List car parts***

***2 - Search car parts***

***3 - Add car part***

***4 - Delete car part***

***9 - Exit***

***Enter an option: 2***

***Enter a part to search for (no spaces): transmission***

***'transmission' found at index 13.***

***Part Pretenders Menu***

***1 - List car parts***

***2 - Search car parts***

***3 - Add car part***

***4 - Delete car part***

***9 - Exit***

***Enter an option: 9***

***End of Part Pretenders***

***PS C:\Users\rorys\OneDrive - Wayne State University\CSC1100-C++\project2>***

**\* 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**.