Lab 2

Due Date: Sep 11, 2019
Total Points: 15 points

The purpose of this lab is to review working with arrays and vectors.

Program 1

Write four functions to perform the following tasks:

a. ***ReadData***: this function opens a file "arrays.txt" and reads its contents into 3 integer arrays named A, B and C. Each array holds 10 integers. The input file contains 3 lines; each line holds 10 integer values. So the data in the first line is used to initialize array A. The data in the second line is used to initialize array B. And the data in the third line is used to initialize array C.

   You may assume that "arrays.txt" has the following integers values:

```
10 9 7 5 4 4 4 3 2 1
0 1 3 4 5 6 7 8 9 10
0 1 3 4 5 7 10 11 12 13
```

b. ***PrintArray:*** this function prints the elements of an array of any size.

c. ***ReverseArray:*** this function reverses an array of any size <u>in place</u>.

d. ***LongestSequence:*** this function finds and returns the length of the longest consecutive sequence of numbers in an array of any size. You may assume that the array is sorted in ascending order. For example.
   Longest sequence of {1,2,3,4,4,4,5,7,9,10} is 5. Note that repeated numbers in a sequence are skipped.
   Longest sequence of {0,1,3,4,5,6,7,8,9,10} is 8.
   Longest sequence of {0,1,3,4,5,7,10,11,12,13} is 4.

Write a main program that:

- Declares 3 arrays A, B, and C
- Calls the function *readData* to initialize these arrays
- Calls the function *printArray* three times to print the elements of arrays A, B, and C
- Calls the function *reverseArray* and sends A as an argument
- Calls the function *printArray* to print the elements of array A
- Calls the function *longestSequence* three times to find and return the longest sequence in all three arrays. The results should be printed in main

Note: Any parameter that is not supposed to be changed inside a function must be declared as a constant parameter.

Program 2

Write a program that creates an empty vector of strings called V. Vector V grows and shrinks as the user processes commands from a data file called “vectorData.txt”.

Each line in the transaction file contains a command and the corresponding data. For example, you may have the following information in your file:

```
Insert          Hello       4
Delete                      5
Print
```

The transaction file can only include three types of commands: Insert, Delete, and Print.

- Insert command inserts a string value in the vector at a specific position. So the Insert command comes with two more information, the string you need to insert and the position it should be inserted at. For example, the first line indicates that the word “Hello” should be inserted in V[4]. You should check if this insert is possible. It is possible if the position you are attempting to insert the element is a positive number not beyond the size of the vector.

- Delete command deletes the element at the specified position. So Delete comes with one more information that indicates which element (index) should be deleted. For example, The second line means V[5] should be removed. Again this operation should only be allowed if the index is positive and not beyond the current size of the vector.

- Print command prints the contents of the vector on the screen.

You may test your program with the following data file:

```
Delete                  -1
Insert      Students    0
Insert      Welcome     0
Insert      Back        1
Insert      Ready       5
Insert      Get         -1
Insert      To          6
Insert      Program     3
Insert      Very        3
Insert      Good        4
Delete                  1
Insert      Job         4
Delete                  7
Insert      CS211       1
Delete                  6
Print
```

Sample Output:

```
Welcome CS211 Students Very Good Job
```

Note: Each command must be implemented in a separate function.