COMP2710: Project 3

Points Possible: 100
Deadline: **11:59pm Friday**, **March 24th (Central Time)**
Note: You do not need to submit hard copies.

Goals:

- To learn streams and file I/O
- To learn how to use tools for stream I/O
- To use arrays to group data elements
- To design and implement functions (Note: This topic was covered in Project 2)
- To perform unit testing
- To design a simple algorithm

Write a program that merges the numbers in two files and writes all the numbers into a third file. Your program takes input from two different files and writes its output to a third file. Each input file contains a list of numbers of type int in sorted order from the smallest to the largest. After the program is run, the output file will contain all the numbers in the two input files in one longer list in sorted order from smallest to largest.

You must provide the following user interface. The user input is depicted as **Bold**, but you do not need to display user input in bold. Please replace "Li" with your name. Your program loads two input files and merges the numbers in the two files.

```
*** Welcome to Li's sorting program ***
Enter the first input file name: input1.txt
The list of 6 numbers in file input1.txt is:
4
13
14
17
23
89

Enter the second input file name: input2.txt
The list of 5 numbers in file input2.txt is:
3
7
9
14
15
```

```
The sorted list of 11 numbers is: 3 4 7 9 13 14 14 15 17 23 89
Enter the output file name: output.txt
*** Please check the new file - output.txt ***
*** Goodbye. ***
```

Your program must follow the above user interface.

Design Issues:

Please do NOT intend to implement this project using a single main() function.
You need **at least three functions** to implement this project. The suggested functions are:

1. `array_size <- readfile(inputArray[], instream)`
2. `outputArray_size <- sort(inputArray1[], inputArray1_size, inputArray2[], inputArray2_size, outputArray[])`
3. `void <- writefile(outputArray[], outputArray_size)`

Integration Testing:

Integration testing (a.k.a., Integration and Testing) is the phase in software testing in which individual software modules are combined and tested as a group. You may use the attached two input files to test the correctness of your program as an entire system.

Requirements:

1. (5 points) Use comments to provide a heading at the top of your code containing your name, Auburn UserID, filename, and how to compile your code. Also describe any help or sources that you used (as per the syllabus).
2. (5 points) Your source code file should be named as "**project3_LastName_UserID.cpp**". (e.g., project3_Pham_tmp0038.cpp) **Note**: You will not lose any point if Canvas automatically changes your file name (e.g., project3_LastName_UserID-2.cpp) due to resubmissions.
3. (5 points) Your program must test if files have been **correctly opened**.
4. (5 points) You must **close files** after using them.
5. (10 points) Your program should **read and display numbers** stored in the two input files.
6. (20 points) Your program should **correctly sort and display the numbers**. You should also construct your own lists including **negative numbers and zeros for testing**.
7. (15 points) Your program should **correctly write the sorted list** to the third file.
8. (10 points) You must define **at least three functions**.
9. (5 points) You must reduce the number of **global variables** to **less than 3**.
10. (10 points) Usability of your program (e.g., user interface)
11. (10 points) Readability of your source code.

Note: You will lose **at least 40 points** if there are compilation errors or warning messages when the TA compiles your source code. You will lose points if you: do not use the specific program file name, or do not have a comment on each function in your program you hand in.

Programming Environment:

Write a program in C++. Compile and run it using AU server (no matter what kind of text editor you use, please make sure your code could run on AU server, the only test bed we accept is AU server).

Sample Code 1:

The following code shows you how to use arrays and their sizes as input parameters of functions.

```
//Sample code for Project 3
#include <fstream>
#include <iostream>
using namespace std;

const int MAX_SIZE = 100;
```

```
//Input: (1) Array storing data retrieved from the file (i.e.,
instream)
//       (2) input file stream object
//Output: Size of array. Note: you need to use this parameter to
control the array size.
int readfile(int inputArray[], ifstream& instream);

int main( )
{
    ifstream inStream1;

    int iArray1[MAX_SIZE];
    int iArray1_size;
    int iArray2[MAX_SIZE];
    int iArray2_size;


    inStream1.open("input1.txt");

    iArray1_size = readfile(inputAry, inStreamFirst);

    inStreamFirst.close( );

    return 0;
}

int readfile(int inputArray[], ifstream& inStream){
    int index;

    inStream >> inputArray[index];
    while (! inStream.eof()) {
        cout << inputArray[index] << endl;
        index++;
        inStream >> inputArray[index];
    }

    return index;
}
```

Sample Code 2:

The following code shows you (1) how to retrieve a file name from your keyboard, (2) how to open a file, and (3) how to read data from the file.

```
#include <fstream>
#include <iostream>
#include <cstdlib>  //for exit()
using namespace std;
```

```
int main()
{
    ifstream inStream;
    int data;

    cout << "file name:";
    cin >> filename;
    cout << "entered filename is:" << filename << endl;

    // Pass the file name as an array of chars to open()
    // inStream.open(filename);
    inStream.open((char*)filename.c_str());

    if (inStream.fail()) {
        cout << "Input file opening failed." << endl;
        exit(1);
    }

    inStream >> data;
    while (!inStream.eof()) {
        cout << data << endl;
        inStream >> data;
    }
    inStream.close();

    return 0;
}
```

Deliverables:

- Submit your source code file named as "project3_LastName_UserID.cpp" through the Canvas system.

Late Submission Penalty:

- Late submissions are not accepted and will results in a **ZERO** without valid excuses, in which case you should talk to Dr. Li to explain your situation.
- GTA/Instructor will not accept any late submission caused by internet latency.

Rebuttal period:

- You will be given a period of 3 business days to read and respond to the comments and grades of your homework or project assignments. The TA may use this opportunity to address any concern and question you have. The TA also may ask for additional information from you regarding your homework or project.

Academic Integrity (Honesty):

- Students will be expected to understand and follow Academic Honesty policies in place by the university. All work is to be done individually. Students should NOT share any project code or even detailed algorithm information with each other. Your programming code is exclusive to you. If found, all students will receive ZERO, including student(s) who provide source(s).