

COMP2710: Homework 4

Points Possible: 20

Due: **Tuesday, Sept. 18th, 2012 at 11:55 pm on Canvas**

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 Homework 3)
- 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. Your program should define a function that is called with the two input-file streams and the output-file stream as three arguments.

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 “Xiao Qin” with your name. Your program must first run a list of unit test cases and print unit testing results. Next, your program loads two input files and merges the numbers in the two files.

```
*** Unit Test Cases ***
Unit Test Case 1: Function Name - show function name here.
    Case 1.1: explain test case 1.1 here (e.g., input arguments).
    Case 1.1 passed.
    Case 1.2: explain test case 1.2 here (e.g., input arguments).
    Case 1.2 passed.
    ...
Press any key to continue...

Unit Test Case 2: Function Name - show function name here.

    Case 2.1: explain test case 2.1 here (e.g., input arguments).
    Case 2.1 passed.
    Case 2.2: explain test case 2.2 here (e.g., input arguments).
    Case 2.2 passed.
    ...
Press any key to continue...

Add more test cases below
Press any key to continue...
```

```

*** Welcome to Xiao Qin's sorting program ***
Enter the first input file name: input1.txt
The list of 4 numbers in file input1.txt is:
3
7
9
12

Enter the second input file name: input2.txt
The list of 6 numbers in file input2.txt is:
4
6
8
11
16
23

The sorted list of 10 numbers is: 3 4 6 7 8 9 11 12 16 23
Enter the output file name: output3.txt
*** Please check the new file - output3.txt ***
*** Goodbye. ***

```

Your program must follow the above user interface, which includes unit testing results.

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)

Unit Testing:

Unit testing is a way of determining if an individual function or class works. You need to isolate a single function or class and test only that function or class. For each function in this homework, you need to check normal cases and boundary cases.

Examples for tested values:

- string – empty string, medium length, very long
- Array – empty array, first element, last element
- Int – zero, mid-value, high-value

You must implement a unit test driver for each function implemented in your program. You may need to use assert() to develop your unit test drivers if tested results are predictable.

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. (1 point) 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. (1 point) Your source code file should be named as "<username>_hw4.cpp" (for example, mine would read "xzq0001_hw4.cpp").
3. (1 point) Your program must test if files have been correctly opened.
4. (1 point) You must close files after using them.
5. (2 points) Your program should read and display numbers stored in the two input files.
6. (4 points) You program should sort and display the numbers.
7. (2 points) You program should correctly write the sorted list to the third file.
8. (2 points) You must define at least three functions.
9. (3 points) You must implement test drivers for your functions.
10. (1 point) You must reduce the number of global variables and data
11. (1 point) Usability of your program (e.g., user interface)
12. (1 point) Readability of your source code.

Note: You will lose **at least 10 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 short program in C++. Compile and run it using the g++ compiler on a Linux box (either in Shop 3, computer labs in Shelby, your home Linux machine, a Linux box on a virtual machine, or using an emulator like Cygwin).

Sample Code 1:

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

```
//Sample code for Homework 4
#include <fstream>
#include <iostream>
using namespace std;

const int MAX_SIZE = 100;
```

```

//Input: (1) Array storing data retrieved from the file (i.e.,
istream)
//      (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 "<username>_hw4.cpp" through the Canvas system.

Late Submission Penalty:

- Twenty percent (20%) penalty per day for late submission. For example, an assignment submitted after the deadline but up to 1 day (24 hours) late can achieve a maximum of 80% of points allocated for the assignment. An assignment submitted after the deadline but up to 2 days (48 hours) late can achieve a maximum of 60% of points allocated for the assignment.
- Assignment submitted more than 3 days (72 hours) after the deadline will not be graded.

Rebuttal period:

- You will be given a period of 72 hours to read and respond to the comments and grades of your homework or project assignment. 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.