COMP2710 – Frequently Asked Questions
Homework 4

1. What I was wondering was if we needed to write our own algorithms for this or if we should use existing C++ libraries.

   Answer: Please do not use any library to implement your sorting function in this assignment.

2. I'm not very familiar with the C++ but it looks like in the algorithms library there is a min heap class you can implement. So for example would it be a viable solution to use the min heap to combine and sort the numbers for us or should we write our own?

   Answer: The algorithm is very simple because the two arrays are sorted ones. You do **NOT** need the min heap class to implement your sorting function.

3. What kind of input can I expect? Should I program for a specific input format, a list with one number on each line in each file, no blank lines. Or can I not assume this much, should I try to program for several possible input formats, i.e., separated by spaces, comma separated (CSV), etc.

   Answer: Please use two simple text files, each of which has a list with one number on each line. No blank lines please.

4. Here is the error message,

```
***0003_hw4.cpp: In function ‘int main()’:
***0003_hw4.cpp:44:15: error: no matching function for call to
‘std::basic_ifstream<char>::open(std::string&)’
```

   It says something is wrong on line 44.

```
============================================================
  30 int main()
  31 {
  32
  33 int iArray1[MAX_SIZE];
  34 int iArray1_size = 0;
  35 int iArray2[MAX_SIZE];
  36 int iArray2_size = 0;
  37 string developer("Brian Ponder");
  38 string inFile;
  39 ifstream in;
  40
  41 cout << "***Welcome to " << developer << "\'s sorting program ***"
  42          << "Enter the first input file name: ";
  43 cin  >> inFile;
  44 in.open(inFile);
  45 while(in >> iArray1[i]){
  46 iArray1_size++;
  47 i++;
  48
  49 }
============================================================
```

   Answer: The line 44 is "in.open(inFile);"

Below is the definition of ifstream::open
public member function

```
void open ( const char * filename, ios_base::openmode mode =
ios_base::in );
```

As we can see here, the first parameter is a pointer of char* type, but you passed in the inFile of string type.

You need a type convention; please refer to string::c_str(...);

5. In the sample output, you have it reading as something like:

   The list of 4 numbers in file input1.txt is:
   3
   7
   9
   12

   However, in going through the included example source code, the numbers are output in the readFile() function while the function itself returns the number of numbers. For example, 4, in the above output. What this means is that in the part that says "The list of __ numbers in file input1.txt is:" is saying the number 4 before running the function and knowing that the number is 4. Is this a mistake or should I alter the example code such that the file is read in entirety to know how many numbers it includes before outputing the individual numbers one by one?

   Answer: To address this issue, your program needs to load the data into an array first. Then, the program calculates and prints the number of elements stored in the array.

6. When I sort outputarray, do i need to delete duplicate number?

   Answer: No, you do not need to delete the duplicate numbers.

7. Do we need to test the readFile function and the writeFile function as well? if yes, how?

   Answer: It is an optimal task to test the readFile and writeFile functions. You must test the sort (a.k.a., merge) function in your program.

8. Right now my main function calls the readFile function which loads the file information into an array and returns the length. How am I supposed to pass that array too the sort function if it is not returned to the main?

   Answer: Sample code is given below:

```
……
 inputArray1_size = readfile(inputArray1, inStream1);
 inputArray2_size = readfile(inputArray2, inStream2);
……

// sort and print the merged files
outputArray_size = sort(inputArray1, inputArray1_size, inputArray2,
                        inputArray2_size, outputArray);
```

9. What should the MAX_SIZE of each array be?

   Answer: You may set MAX_SIZE to 1000. Since this value is a constant, it can be easily configure in your source code.

10. I have completed the assignment and have it running perfectly on my own computer. I have been using terminal and the program has been able to retrieve the file inputs. However, on the linux server I am unable to retrieve them. Where do I need to place the files so that my program can reach them?

    Answer: You will need to use winscp (windows) or filezilla (mac OS) to transfer the input files from your local machine to a remote linux machine.

11. Is our test driver supposed to be a collection of methods within the homework file or is it supposed to be a separate file?

    Answer: All the test drivers are included in the same single source code file.

12. In my unit testing for readFile(), I use the two given input files (input1.txt and input2.txt). I also use two files that I created myself: input0.txt has 1 value, and input3.txt has 50 values. I used these to test readFile() for large and small input files.
    How should I submit my project if I used these files? Should I include the input files in my upload, or leave a comment next to the test code describing the input file parameters, or something else?

    Answer: Please submit your assignment with your extra input files. You also should leave a comment next to the test code describing your input files.

13. Do we need to handle a situation where the files passed in by the user combine to be larger than the allocated size of the arrays? Ex: const int DEFAULT = 1000, and the 2 files combine to be larger than this.

    Answer: It is an optional feature in your program. If you have time, I encourage you to implement this feature.

14. My test cases currently use separate files that have no numbers, 1000 numbers, and 15 numbers as tests. However, since the canvas page only takes .cpp files how am I supposed to test the functions without including extra files to test?

    Answer: There is no file extension restriction on Canvas. You may compress multiple files in a single file; then, you are allowed to submit a compressed file named either hw4.tar or hw4.zip.