CPSC 410

Project 1

Motivation: Topics covered by this project;

- Vectors (C++ container for holding data)
- Vector sorting
- Structs
- File I/O and streams
- cpp and h files
- code organization
- A little on pointers

Overview

You are given a file that has an unknown number of rows, and 3 columns of integers. Assume there are no malformed rows (ie != 3 columns). Looks like the following.

```
1,10,51
2,5,4
3,3,55
4,100,53
5,15,21
6,0,41
7,7,11
8, 1,  19
```

This file can have any name, but I will call it testdata.txt for the purposes of this document. The first column is process_number, second is start_time and third is cpu_time.

I have provided a header file which describes the API that you are to code to

Requirements

1. Please create and complete utilities.cpp.
2. Please create and complete main.cpp. It should have your main() function. Its purpose is to verify all the functions from utilities.cpp.
3. Use utilities.h to link main.cpp and utilities.cpp. DO NOT MODIFY UTILITIES.H it's a contract between you and my test harness
4. Please remove superfluous imports.

In the interest of a minimal API, notice I've exposed a minimal interface in utilities.h. Clients of utilities.h do not need to see what kind of datastructure is holding

structures, so it isn't referenced anywhere in utilities.h. This means the underlying datastructure can be changed in future releases without affecting existing clients. Also , if a function does not need to be called outside of utilities.cpp then it isn't exposed in utilities.h!
Remember that header files should have ONLY enough includes so that any file that includes them will compile cleanly with no errors. Other necessary includes should go in the cpp file.

Documentation and Testing

Make sure you comment each function and the program as a whole. Test your program extensively (this means writing test code for your utilities.cpp implementation). I will use my own version of TestData.txt and Main.cpp and utilities.h to test your project.

To Turn In

Utilities.cpp only

Grading

40% File input working, including tokenizing lines
60% Your .cpp file

APIs that may help

std::string
std::vector
stringstream
std::getline
ifstream and ofstream
std::sort