# COMP2401 - Assignment #6

## (Due: **Thursday, December 10th, 2020 @ 6pm**)

In this assignment, you will practice reading and writing information to/from both text & binary files.

We will be writing code that reads in a file containing some worldwide data from the Covid-19 Pandemic. The statistics were extracted from here: https://www.worldometers.info/coronavirus/ on May 28, 2020. The data set contains statistics on Covid-19 cases from 215 countries, including the number of cases, the number of deaths, the number of recovered, population, etc… Follow the steps below to complete the assignment.


## Part 1

Download the file called **pandemic.tsv**. This is a text file with "tab-separated values". That means, a tab character separates the values on each line. It contains a header line, followed by **215** lines of country statistics. Write a program called **tsvToBin.c** that will read in this file, extract some data from it and then re-write it to 3 new files called **ongoing.bin**, **resolved.bin** and **incomplete.bin**. Here is what you need to do:

1. Make sure that you open the .tsv file for reading and close it when you are done. You must also check for errors in opening the file and use **printf** to indicate if an error has occurred.

2. You must open the binary files for writing, and close them when you are done. You must check for errors as well in case the file cannot be created. Each time you run the program, the binary files must be overwritten.

3. The program must read in all the lines of the .tsv file until none remain. There are **215** countries in the file but you MUST NOT hard-code this number anywhere. It must be computed by reading through the entire file once to determine how many lines are in there. Thus, your program should work properly regardless of the number of lines in the file.

4. The binary files will contain statistic information in the same order as the .tsv file, however each of the .tsv file lines of data will be written to only one of the binary files. You will divide up the countries into the three files based on the statistics. Countries that have ongoing cases (i.e., ongoing means that there are still cases that have not yet resulted in recovery nor death) will be written to the **ongoing.bin** file. Countries in which all cases have been resolved (i.e., #cases = #deaths + #recovered) will be written to the **resolved.bin** file. Any countries with missing "important" data (i.e., total cases, total deaths, total recovered or population) will be written to the **incomplete.bin** file. Each binary file must begin with an **unsigned short int** indicating how many countries have been written to that file. As a helpful hint, you can re-write a number to the file after it has been written … if you need to.

5. You MUST NOT create any array to hold all of the data that you read in. You may, however, determine the maximum number of characters in a country name and the maximum number of characters in a line … and define appropriate constants … if that helps.

6. You will need to read in one piece of data at a time. Each piece of data is separated by a tab character. If you want to use **fscanf()**, you can make use of the following format string which will read everything up to (but not including) a tab character: `"%[^\t]s"`. And the following will read everything up to (but not including) a newline character: `"%[^\n]s"`.

7. You must read in the first line … which is a header line, but then it will be discarded. After that, each line read in represents the statistics (shown here on the right) for a country. Only the data in RED should be written to the binary files, following these rules:

#
**Country / Other**
**Total Cases**
New Cases
**Total Deaths**
New Deaths
**Total Recovered**
Active Cases
Serious / Critical
Total Cases/ 1M pop
Deaths/1M pop
Total Tests
Tests/1M pop
**Population**

   - The **Country** should be written to the binary files by first writing a byte indicating the number of characters in the country name, followed by the exact number of characters needed (i.e., no extra spacing nor padding) for the name. For example, if “Canada” is the country title, then a 6 should be written, followed by the 6 letters in “Canada”. The NULL-terminating character should NOT be written.

   - The **Total Cases**, **Total Deaths**, **Total Recovered** and **Population** should all be written as **unsigned ints**. For the **resolved.bin** file, in order to keep the file size smaller, you will NOT write the **Total Cases,** since this is redundant and can be computed later as **Total Deaths** + **Total Recovered**.

   - When reading the values from the text file, you will notice that all numbers above 999 will have a comma in them as well as quotes (e.g., "23,718") but the numbers less than 1,000 will not have quotes. You will need to handle the reading/processing of these numbers properly. You will likely want to write a function that takes a number string (i.e., either “`999`” or “`"23,178"`”) and then returns the proper integer that it represents. Be aware that a couple of the numbers may have N/A as their value or may be blank, indicating that the data was not available. Depending on how you read in the data, the blank could appear as an empty string, a string with a tab, a string with a '\r' or '\n' character, etc… For invalid or missing data, you should set the value to -1. However, for some countries, the **Total Deaths** and/or **Total Recovered** is blank because there have not been any occurrences as of yet. In this situation, you should set the value to 0, not -1. You can check if it should be zero by comparing the active cases (read in from the original file) with the total cases. If all cases are active, then there should be zero deaths and zero recovered. If all cases have recovered, then there should be zero deaths. If all cases have died, then there should be zero recovered. In these situations, we must not assume that the data is incomplete, but just that the blanks are zero.

   - Finally, for the **ongoing.bin** file only, a single **char** should be written as ‘**L**’, ‘**M**’ or ‘**H**’ indicating whether the virus has had a **low**, **medium** or **high** impact on the population. If the total number of cases is less than or equal to **0.05%** of the population, then this was a **low** impact. If it was greater than or equal to **0.30%** of the population, then this was a **high** impact. Otherwise it was a **medium** impact.

   - None of the other data should be written to the files.

Here are the sizes of the files (in bytes) that you MUST have if everything is done properly:

- **pandemic.tsv** **16,124** bytes (has data for 215 countries)
- **ongoing.bin** **4,929** bytes (has data for 189 countries)
- **resolved.bin** **549** bytes (has data for 22 countries)
- **incomplete.bin** **109** bytes (has data for 4 countries)

You can use **ls -l** to confirm the file sizes. If you cannot get the correct file sizes, then something is wrong. However, do not get hung up on this if you cannot get it to be the correct sizes. Just move on to the next part … you can always come back to this later and figure it out. It is not worth a lot of marks to ensure the correct sizes.

# Part 2

Write a program called **pandemicStats.c** while reads in the three binary files that you just created and prints out the information as discussed below. You must open and close the files properly and handle any errors accordingly. The following should be displayed in the order here:

- The number of countries in the **ongoing.bin** file followed by a nicely-formatted printout of the data in the file in the format shown here. Note that in order for numbers to be displayed with commas, you need to include `<locale.h>` in your file and also call `setlocale(LC_NUMERIC, "");` once in your program.

| Country | Cases | Deaths | Recovered | Population | Impact |
|---|---|---|---|---|---|
| Afghanistan | 13,036 | 235 | 1,209 | 38,838,960 | L |
| Albania | 1,076 | 33 | 823 | 2,878,078 | L |
| Algeria | 8,997 | 630 | 5,277 | 43,772,270 | L |
| Andorra | 763 | 51 | 681 | 77,254 | H |
| Angola | 71 | 4 | 18 | 32,757,041 | L |
| Antigua and Barbuda | 25 | 3 | 19 | 97,853 | L |
| Argentina | 13,933 | 501 | 4,617 | 45,156,510 | L |
| Armenia | 8,216 | 113 | 3,287 | 2,962,740 | M |
| ... | ... | ... | ... | ... | ... |

- The number of countries in the **resolved.bin** file followed by a nicely-formatted printout of the data in the file in the format shown here. Note that the Cases is not read in from the file … it must be computed.

| Country | Cases | Deaths | Recovered | Population |
|---|---|---|---|---|
| Anguilla | 3 | 0 | 3 | 14,990 |
| Belize | 18 | 2 | 16 | 396,910 |
| British Virgin Islands | 8 | 1 | 7 | 30,212 |
| Caribbean Netherlands | 6 | 0 | 6 | 26,200 |
| Dominica | 16 | 0 | 16 | 71,970 |
| ... | ... | ... | ... | ... |

- The number of countries in the **incomplete.bin** file followed by a nicely-formatted printout of the data in the file in the format shown here. Note that any missing data should appear as ---.

| Country | Cases | Deaths | Recovered | Population |
|---|---|---|---|---|
| Diamond Princess | 712 | 13 | 651 | --- |
| MS Zaandam | 9 | 2 | 0 | --- |
| Netherlands | 45,950 | 5,903 | --- | 17,132,422 |
| UK | 269,127 | 37,837 | --- | 67,852,992 |
| ... | ... | ... | ... | ... |

- The following statistics (Note that for the stats below, you will be ignoring stats from all countries that are in the **incomplete.bin** file):

  (1) The total world population for which we have complete statistics. Should be **7,666,356,914**.

  (2) The total number of cases worldwide. Should be **5,536,801**. Also show what percent of the population this represents. Should be **0.07%**.

  (3) The total number of deaths worldwide. Should be **316,050**. Also show what percent of the cases this represents .. which is the death rate. Should be **5.71%**.

  (4) The total number of recovered worldwide. Should be **2,538,669**. Also show what percent of the cases this represents .. which is the recovery rate. Should be **45.85%**.

  (5) The total number of ongoing cases worldwide. Should be **2,682,082**. Also show what percent of the cases this represents. Should be **48.44%**.

- A list of the countries with ongoing cases, sorted by the number of cases (highest first). You MUST use the **qsort()** function in C to do this. You should display the number of cases and country name, nicely formatted as follows:

```
  1,756,161   USA
    418,608   Brazil
    379,051   Russia
    283,849   Spain
    231,732   Italy
    182,913   France
    182,313   Germany
    165,348   India
    160,979   Turkey
    143,849   Iran
    135,905   Peru
     88,468   Canada
```

---

## IMPORTANT SUBMISSION INSTRUCTIONS:

Submit all of your **c source code** files as a single **tar** file containing:

1. A **Readme** text file containing
   - your name and studentNumber
   - a list of source files submitted
   - any specific instructions for compiling and/or running your code
2. All of your **.c source** files and all other files needed for testing/running your programs.
3. Any output files required, if there are any.

The code **MUST** compile and run on the course VM.

- If your internet connection at home is down or does not work, we will not accept this as a reason for handing in an assignment late ... so make sure to submit the assignment WELL BEFORE it is due !

- You WILL lose marks on this assignment if any of your files are missing. So, make sure that you hand in the correct files and version of your assignment. You will also lose marks if your code is not **written neatly with proper indentation and containing a reasonable number of comments.** See course notes for examples of what is proper indentation, writing style and reasonable commenting).