CCPROG2 MP SPECIFICATIONS PART 1 [15 points] サルバドール・フロランテ
AY2020 Term 2

GENERAL INSTRUCTIONS

1. **Honesty Policy and Honor Code apply. Cheating of any form is NOT an acceptable behavior. It is a major offense punishable with a final grade of 0.0.**
2. Make sure that you have read and understood the problem requirements. You will be presented with several challenge problems. Solve the challenges by formulating your algorithms, and encoding your solutions using the given skeleton codes.
3. Comply with the specifications and restrictions stated in MP Specs document, and inside the comments in the skeleton codes. Non-compliance will result into deductions.
4. You are NOT allowed to use pre-defined library functions that we did not discuss in class (unless specified otherwise).
5. Make sure that there are no warnings, no syntax errors and no logical errors in your solutions. **Subject your solution to exhaustive testing to ensure that it is logically correct.** The following scoring and deductions system will be applied.
    - 10/10 points will be awarded for a compliant and logically correct solution.
    - A solution with logical error/s will be awarded 0 to at most 6/10 points. Do NOT expect a point for "effort". The score is based solely on the correctness of your solution -- not on how many hours and how much effort you put into it.
    - Each unique compiler warning will result into a deduction of one point.
    - Non-compliance will result into a deduction ranging from one point to five points.
    - A syntax error will result into a score of 0/10 for the associated challenge.
6. You will be required to submit C source files, header files, and text files via Canvas before a specified deadline. Late submission beyond the deadline without a supporting, verifiable and reasonable excuse will not be accepted.
7. To identify your submission, you will need to indicate your group number as part of the filename.
8. ALL questions about the MP should be POSTed in the Canvas Discussion Q&A on Machine Problem. Please do NOT email me – except when your reason is a personal one -- for example, you cannot submit your solution due to sickness.

I. INTRODUCTION

Big Data and *Data Science* are recent buzz words in multiple disciplines including the sciences and engineering. For this Machine Problem (MP), we will develop C programs that will answer real-life questions based on real-life COVID-19 related data.

Concepts covered in CCPROG2, i.e., arrays, string, structures, text and binary file processing will be applied. The MP is a learning activity meant to assess if you can:

- perform data gathering
- design and implement your own data structure for representing, storing, accessing and processing COVID-19 data
- design and implement your own algorithms for the challenge problems in the MP specification
- specify your test cases
- test and debug programs
- properly document and articulate your solution to the MP

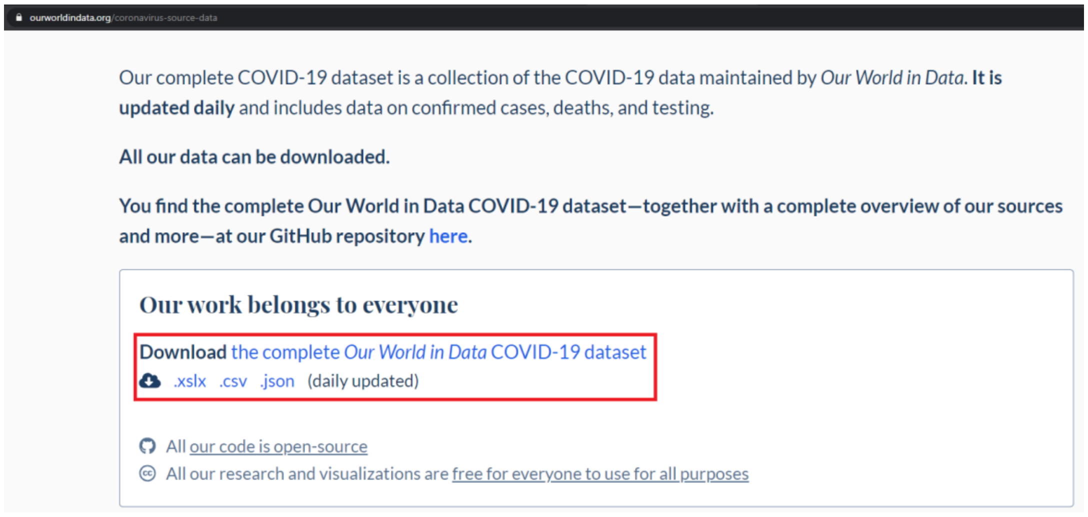
II. CHALLENGES

Challenge #0: Obtain COVID-19 Historical Data (CHD) **[5 POINTS]**

As a preliminary task, we need to obtain country based **COVID-19 Historical Data** (CHD). Fortunately, this is an easy clerical task that can be done in less than a day since the data that we need are downloadable for free from "Our World in Data" website https://ourworldindata.org/coronavirus-source-data.

Do the following:

1. Go to the website specified above. Click on the link indicating .xlsx to download the Covid-19 Data Set in Excel xlsx file format – refer to Figure 1. Note that the data set is updated daily.
2. Open the downloaded Excel file and examine the contents. There are LOTS of rows and columns of data. We will only use a subset of the data values, more specifically:
    a. column C: location
    b. column D: date
    c. column F: new_cases
    d. column I: new_deaths
    e. column AS: population
    f. column BF: life_expectancy


ourworldindata.org/coronavirus-source-data

Our complete COVID-19 dataset is a collection of the COVID-19 data maintained by *Our World in Data*. **It is updated daily** and includes data on confirmed cases, deaths, and testing.

All our data can be downloaded.

You find the complete Our World in Data COVID-19 dataset—together with a complete overview of our sources and more—at our GitHub repository here.

Our work belongs to everyone

Download the complete *Our World in Data* COVID-19 dataset
.xslx .csv .json (daily updated)

All our code is open-source
All our research and visualizations are free for everyone to use for all purposes

Figure 1. Screenshot from "Our World in Data" https://ourworldindata.org/coronavirus-source-data

3. For the two countries that you chose (recall https://docs.google.com/spreadsheets/d/10tFErkDaip0gvLRpr5Iy3JWVYnl-MxhwhL_y-pn6weo/edit?usp=sharing):
   a. In case there are cells that are blank in new_case and new_deaths columns, fill them up with a value of zero. For example, if you chose Afghanistan, you'll need to fill up the cells in column I, rows 2 to 28 with 0.
   b. Format the date as **YYYY-MM-DD** (this format is an important consideration in Challenge #1).
   c. Once you are sure that there are no more blank cells, and that the dates are already formatted as required, copy the values for date, new_cases, new_deaths, population and life_expectancy for the country that you chose, and store them into a text file. **Important Note: Do NOT copy any data beyond March 21, 2021.**
   d. Name the text file with the same name as the country. For example, data for the Philippines will be copied into a text file named as PHILIPPINES.TXT. For countries whose names are made up of more than one word, for example, Sri Lanka, connect the words with a hyphen, for example use SRI-LANKA.TXT. Ensure that there is no blank or space in the filename.
   e. Verify that the data values in the text file are correct (based on the original Excel file), and that the text file can be loaded and viewed in a text editor like Notepad.

4. Submit the two text files as your deliverable for this challenge before the indicated Canvas deadline.

For your reference, an example text file named PHILIPPINES.TXT accompanies this MP specifications. Inspect it so that you will have a clear idea on the expected text file contents.

Challenge #1: Representing and Storing the CHD Using Arrays. [10 POINTS]
For this task, you will apply your knowledge of 1D array, 2D array and string processing. Write a C program that will read all the contents of one text file (the source file) containing a country's CHD via input redirection. Do some processing (to comply with the requirements below) to produce another text file (the destination file) via output redirection.

Hard Requirements: Your program should
1. read via **`scanf()`** the date as a string, and the other data values as numbers
2. all date values must be stored in a 1D array of string, all new_case and new_deaths values in a 2D integer array
3. note that the population and life expectancy values do NOT change each day (unlike the new_cases and new_deaths numbers) so you do not need an array to store them; just use one variable each to store their values.

Do NOT use **`struct`** data type yet for this challenge.

The program should produce as output the following:

On the first row: population
On the second row: life expectancy - with 2 digits after the decimal point
On the third row: empty line

Starting from the fourth row: 5 columns of data indicating the following

1st column: date values in **YYYY/MM/DD** format; for example, March 8, 2021 is printed as 2021/03/08
2nd column: new_case values
3rd column: running total of the number of new cases (accumulated values)
4th column: new_deaths values
5th column: running total of the number of deaths (accumulated values)

It is up to you to decide how many white spaces you want as a separator between values within the same line of text.

For example, using data for the Philippines, the program's expected output is:

```
109581085
71.23

2020/01/30   1     1     0    0
2020/01/31   0     1     0    0
2020/02/01   0     1     0    0
2020/02/02   1     2     1    1
2020/02/03   0     2     0    1
  :
  :  other lines of output not shown due to space limitation
  :
2021/03/21   7738   663794   38   12968
```

Interpretation: the last line shows that on March 21, 2021 there were 7738 new cases, and that total number of COVID-19 cases as of said date is 663794. Also, on the same date, there were 38 new deaths, and that the total number of deaths as of the said date is 12968.

Complete the accompanying skeleton file named **`C1-NUMBER.c`**. Note that C1 means Challenge 1, and NUMBER should be replaced with a 2-digit number representing the group's number (see Canvas Groups). For example, if the group's number is 8, then the file should be named as C1-08.c. Note that there should be a leading zero for group numbers 1 to 9.

Compile, run and test your solution exhaustively. Do NOT forget to enable -Wall compiler directive for a strict compiler setting.

HOW TO RUN and TEST YOUR PROGRAM: You should run the exe file of your program in the command line with **input and output redirection**. For example:

```
C:\CCPROG2>C1-08 < PHILIPPINES.TXT > RESULT.TXT
```

where C1-08.exe is the group's executable file, PHILIPPINES.TXT is the source file, and RESULT.TXT is the destination file. The < (less than symbol) is for input redirection, and the > (greater than symbol) is for output redirection.

The program's output will be written in the text file RESULT.TXT.

DELIVERABLE: submit your source code **`C1-NUMBER.c`** via Canvas before the indicated deadline.

More challenges in Part 2…