CCPROG2 MP SPECIFICATIONS PART 3 [40 points] *サルバドール・フロランテ*
AY2020 Term 2

COVID-19 Historical Data: (some simple) Analysis

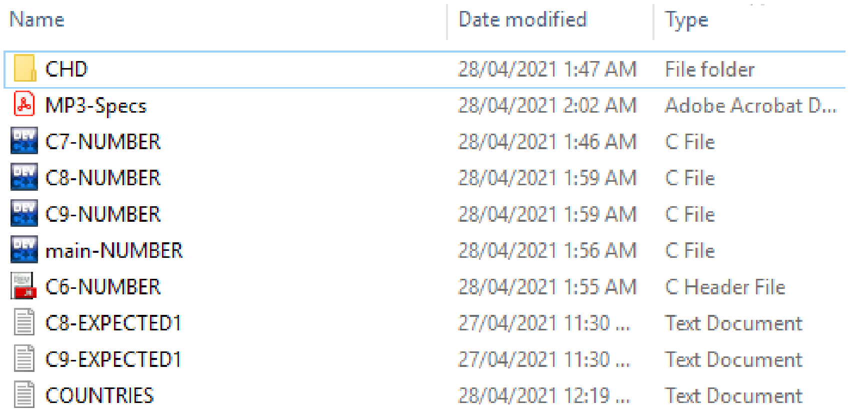
- Honesty Policy and Honor Code apply. Cheating of any form is NOT an acceptable behavior. It is a major offense punishable with a final grade of 0.0.
- You are NOT allowed to use library functions which were not discussed in class (unless specified explicitly otherwise).
- Refer to the accompanying skeleton file(s) that you need to complete.

Objective: apply and demonstrate your knowledge on text file processing.

Preliminary Requirements:

1. Open https://docs.google.com/spreadsheets/d/10tFErkDaip0gvLRpr5Iy3JWVYnl-MxhwhL_y-pn6weo/edit?usp=sharing
2. Copy the data in columns A (CONTINENT)and B (COUNTRY) from rows 2 to 174 into a file named `CONTINENT-COUNTRY.TXT`. Verify the following:
    a. there is no empty cell in the CONTINENT column
    b. the cell content is in capital letters
    c. a dash connects NORTH with AMERICA, and SOUTH with AMERICA.

    If there is a problem with a cell, email and pressure the student-in-charge of the cell to have it fixed.
3. Ensure that the file `CONTINENT-COUNTRY.TXT` containing the correct data is stored in the CHD folder. Recall that the CHD folder contains the text files where COVID data for a particular country are encoded.
4. Make sure that your MP3 header and C source files are in the same folder where you have CHD as shown in the following example screenshot.



Name	Date modified	Type
CHD	28/04/2021 1:47 AM	File folder
MP3-Specs	28/04/2021 2:02 AM	Adobe Acrobat D...
C7-NUMBER	28/04/2021 1:46 AM	C File
C8-NUMBER	28/04/2021 1:59 AM	C File
C9-NUMBER	28/04/2021 1:59 AM	C File
main-NUMBER	28/04/2021 1:56 AM	C File
C6-NUMBER	28/04/2021 1:55 AM	C Header File
C8-EXPECTED1	27/04/2021 11:30 ...	Text Document
C9-EXPECTED1	27/04/2021 11:30 ...	Text Document
COUNTRIES	28/04/2021 12:19 ...	Text Document

Challenge #6 (C6): Header File for the Succeeding Challenges C7 to C9. **[5 POINTS]**
Using **C6-NUMBER.h** (header file) as skeleton do the following:

1. Edit the line that contains `#define GROUP_NUMBER "88"` as specified in the header file instructions.
2. Add the following:
    - contents of your C2 header file – you may edit the copied contents as you deem necessary.
    - macro definitions
    - structure type declarations
    - typedef declarations (aliases for strings and structures)
    - function prototypes for all functions that you defined in order accomplish C7 to C9.

Edit/modify the contents of C6-NUMBER.h to satisfy the requirements of C7 to C9.

Challenge #7 (C7): Reading COVID-19 Historical Data From a Text File. [10 POINTS]
Using **C7-NUMBER.c** (C source file) as skeleton, define the function

```
int Read_COVID_Data(char * param_country, _____ ptrData)
```

which will read all the contents of one text file (the source file) containing a country's COVID-19 historical data. The 1st parameter is the country's name, and the 2nd parameter is a pointer to a structure variable whose data type was specified (by you) in Challenge #2. The structure variable will contain information about the country (i.e., name, population, life span and the COVID-19 historical daily data) after executing the function. You must fill in the blank with the appropriate parameter data type.

C7 is essentially the same as C3 except that you will no longer use input redirection. Instead, your program should read the data via **`fscanf()`** from a text file based on the value of **`param_country`**. For example, when **`param_country`** is "PHILIPPINES", the corresponding text file to be read will be "PHILIPPINES.TXT" located in the CHD folder.

In case there is no text file corresponding to **`param_country`** (for example: there is no text file for GABON), the function should output an error message indicating "No data for <param_country>." to the **`stderr`** device BUT should not terminate the function via **`exit()`**. Do not forget to encode a period after <param_country>.

The function should return a 1 if data were read successfully, otherwise it should return a 0.

Challenge #8 (C8): Statistics for a List of Locations – Sorted Alphabetically by Country Name. [10 POINTS]
Write a function that will compute the answer to the following question:

Q: What are the statistics (population, number of cases, deaths) for each country listed in <param_input_filename>?

Using **C8-NUMBER.c** (C source file) as skeleton, define the function

```
int Stats_C8(char * param_output_filename, char * param_input_filename)
```

which will compute the required statistics for each country listed in the input text file named as **`param_input_filename`**. Assume that there is at least one country/location listed in the input file if it exists.

The function should write SIX columns of values listed below into an output text file named **`param_output_filename`**. The output data should be in alphabetical order by country name.

1. country name
2. population
3. total number of cases
4. percentage of the number of cases with respect to the population
5. total number of deaths
6. percentage of the number of cases with respect to the population

The percentage values should be printed as a floating point with six digits after the decimal point. You are free to decide on how much space you would like to put in between columns.

In case there is no text file corresponding to **`param_input_filename`**, the function should output an error message indicating "<param_input_file> file not found." to the **`stderr`** device, BUT should not terminate the function via **`exit()`**. Do not forget to encode a period after the word "found".

The function should return 1 if data were processed successfully, otherwise it should return 0.

HARD REQUIREMENTS: The function should

- call **`Read_COVID_Data()`**
- use selection sort algorithm.

Example #1: the file corresponding to the 2nd parameter exists.

```
int result;
result = Stats_C8("C8-EXPECTED1.TXT", "COUNTRIES.TXT");
```

View and study the contents of the accompanying example files "**COUNTRIES.TXT**" and "**C8-EXPECTED1.TXT**" to help you understand the requirements for this challenge. The value of result is 1 in the example above.

Example #2: the file corresponding to the 2nd parameter DOES not exist.

```
int result;
result = Stats_C8("DUMMY.TXT", "WALA-ITO.TXT");
```

The function outputs the error message "WALA-ITO.TXT not found." in the **stderr** device, and returns 0. Note also that the file "**DUMMY.TXT**" should not be created, and therefore should not be seen if we issue a DIR command.

Challenge #9 (C9): Statistics Per Continent. **[NOTE: this challenge is worth 15 POINTS!]**
Write a function that will compute the answer to the following question:

Q: What are the statistics (population, number of cases, deaths) for continent based on the countries listed in <param_input_filename>?

This challenge is essentially the same as C8 but instead of a per country statistics you will need to compute the statistics per continent.

Using **C9-NUMBER.c** (C source file) as skeleton, define the function

```
int Stats_C9(char *param_output_filename, char *param_input_filename)
```

which will compute the required statistics for the continents corresponding to each country listed in the input text file named as **param_input_filename**. Assume that there is at least one country/location listed in the input file if it exists.

The function should write SIX columns of values listed below into an output text file named **param_output_filename**. The output data should be in alphabetical order by continent name.

1. continent name
2. population
3. total number of cases
4. percentage of the number of cases with respect to the population
5. total number of deaths
6. percentage of the number of cases with respect to the population

In case there is no text file corresponding to **param_input_filename**, the function should output an error message indicating "<param_input_file> file not found." to the **stderr** device, BUT should not terminate the function via **exit()**. Do not forget to encode a period after the word "found".

The function should return 1 if data were processed successfully, otherwise it should return 0.

HARD REQUIREMENTS: The function should

- call **Read_COVID_Data()**
- use binary search algorithm to determine the continent corresponding to a given country. HINT: recall that the text file **CONTINENT-COUNTRY.TXT** contains data about continents and countries. The country names are already listed alphabetically in that text file so there is no need to sort them.

Example #1: the file corresponding to the 2nd parameter exists.

```
        int result;
        result = Stats_C9("C9-EXPECTED1.TXT", "COUNTRIES.TXT");
```

View and study the contents of the accompanying example files "**COUNTRIES.TXT**" and "**C9-EXPECTED1.TXT**" to help you understand the requirements for this challenge. The value of result is 1 in the example above.

Example #2: the file corresponding to the 2nd parameter DOES not exist.

```
        int result;
        result = Stats_C9("DUMMY.TXT", "WALA-ITO.TXT");
```

The function outputs the error message "WALA-ITO.TXT not found." in the **stderr** device, and returns 0. Note also that the file "**DUMMY.TXT**" should not be created, and therefore should not be seen if we issue a DIR command.

HOW TO RUN AND TEST YOUR SOLUTION:
Read, understand and follow the instructions given in the accompanying **main-NUMBER.c** file.

DELIVERABLES: submit a zip file named **NUMBER.zip**. There should be exactly TEN extracted files when it is unzipped.

1. **C6-NUMBER.h**
2. **C7-NUMBER.c**
3. **C8-NUMBER.c**
4. **C9-NUMBER.c**
5. **main-NUMBER.c**
6. **COUNTRIES-NUMBER.TXT** – Create this text file. It should contain a minimum of 25 to a maximum of 50 countries (names) of your own choice for testing your C8 and C9 solutions. Choose countries such that all continents, and countries with different first letters are represented.
7. **C8-NUMBER-OUTPUT1.TXT** – C8 output text file using the accompanying **COUNTRIES.TXT** as input file
8. **C8-NUMBER-OUTPUT2.TXT** – C8 output text file using your own **COUNTRIES-NUMBER.TXT** as input file
9. **C9-NUMBER-OUTPUT1.TXT** – C9 output text file using the accompanying **COUNTRIES.TXT** as input file
10. **C9-NUMBER-OUTPUT2.TXT** – C9 output text file using your own **COUNTRIES-NUMBER.TXT** as input file

Do NOT include other files – non-compliance with this instruction will result to a 0.5 point deduction for every unnecessary file.
Make sure to replace NUMBER with your own group number.

Questions? Post them in our Canvas Q&A thread on the MP.

--- The End ---