CCPROG2 MP SPECIFICATIONS PART 2 [35 points]
AY2020 Term 2

サルバド-ル・フロランテ

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'll need to complete.

MP1 was concerned with the use of arrays and strings. For MP2 you will need to apply and demonstrate your knowledge of **struct** data type.

Challenge #2: Representing COVID-19 Historical Data (CHD) Using Structures. [5 POINTS]
You used two separate arrays, i.e., a 1D array of strings, and a 2D integer array to represent the COVID-19 daily data in the previous challenge. For Challenge #2, you will use instead a better representation, i.e., an array of structures.

Accomplish the following series of steps. Encode your solution using the accompanying **`C2-NUMBER.h`** (header file).

1. Define a macro name for an integer value to be used as the maximum 1D array size of COVID-19 historical data. You must make a logical decision on what should be the appropriate array size to use.
2. Declare a structure data type (not a variable!) with 3 members that can be used to represent ONE row of daily data. Recall that a row of daily COVID-19 data contains the following values in sequence:
    a. date (string) - you are required to use the provided alias **`StrDate`** as the data type of an entity to be used for storing a date value as a string.
    b. new_cases (integer)
    c. new_deaths (integer)
3. Declare a **`typedef`** alias for the structure data type in the previous step.
4. Declare a 2nd structure data type (not variable!) with 5 members that can be used to represent data/statistics for one country. The structure members should be specified in the sequence given below:
    a. country/location name (note: the longest country name in the CHD folder is "SAINT-VINCENT-AND-THE-GRENADINES")
    b. population (use long integer as data type)
    c. life expectancy (float data type)
    d. a 1D array of structures based on what you accomplished in steps 2 and 3 above. The array size should be what you specified in step 1.
    e. the number of daily data that are actually stored in the array. Note that this number differs for each country. It is also less than or equal to the macro value in step 1. For example: for PHILIPPINES.TXT this value is 417, for SAMOA.TXT this value is 124.
5. Declare a **`typedef`** alias for the structure data type in the previous step.

You may use any name (structure type name, member name, alias name) that you want – just make sure that the names are suggestive of their use (i.e., self-documenting) and do not conflict with other pre-defined and existing names. Include inline comments for internal documentation.

Your header file should NOT contain any unnecessary declarations.

Test your header file thoroughly on Challenge #3 first before you submit it.

Challenge #3 (C3): Reading/Storing COVID-19 Historical Data. [10 POINTS]
Using **C3-NUMBER.c** (C source file) as skeleton, define the function

```
void Input_COVID_Data( ____ param_country, _____ ptrData )
```

which will read using **scanf()** all the contents of one text file (the source file) containing a country's COVID-19 historical data via input redirection. You must supply the appropriate parameter data type.

The 1st parameter is the country's name, and the 2nd parameter is a pointer to a structure variable whose data type was specified in Challenge #2. The structure variable contains information about the country (i.e., name, population, life span and the COVID-19 historical daily data) after executing the function.

Challenge #4 (C4): Binary Search Applied on COVID-19 Historical Data. [10 POINTS]
Write a function that will compute the answer to the following question:

Q: Is there COVID-19 data for <param_date>?

Examples: Is there COVID-19 data for "2020-03-21"?
Is there COVID-19 data for "2019-08-08"?

Using **C4-NUMBER.c** (C source file) as skeleton, define the function

```
int COVID_Search( _______param_date, ____ Data )
```

which will perform a **binary search** to determine if there is an entry corresponding to **param_date** in **Data**. If such an entry exists, the function should return the corresponding array index; otherwise, it should return a -1. Assume that **param_date** is always formatted as "YYYY-MM-DD".

Challenge #5 (C5): Monthly Summary of COVID-19 Historical Data. [10 POINTS]
Write a function that will compute the answer to the following question:

Q: What are the monthly COVID-19 statistics for <param_country>?

Example: What are the COVID-19 monthly statistics for "PHILIPPINES".

Using **C5-NUMBER.c** (C source file) as skeleton, define the function

```
void COVID_Monthly_Statistics( _______ param_country, ____ Data )
```

which will produce via **printf()** a summary of the monthly statistics based on the available data (in the text file) for a given country. Assume at this stage that the **param_country** value is the same as the country member value in parameter **Data**.

Make sure to produce an output that complies with the following format:

```
<country name>
<population>
<life expectancy with 2 digits after the decimal point>
one blank line
<Month as a 3-letter word> <start day> to <end day>, <year>  <total monthly cases> <total monthly deaths>
                                                          :
                                                          :
                                                          :
<Month as a 3-letter word> <start day> to <end day>, <year>  <total monthly cases> <total monthly deaths>
```

Using PHILIPPINES.TXT as an example, the expected result should be:

```
PHILIPPINES
109581085
71.23

Jan 30 to 31, 2020         1          0
Feb  1 to 29, 2020         2          1
Mar  1 to 31, 2020      2081         87
Apr  1 to 30, 2020      6177        472
May  1 to 31, 2020      9314        378
Jun  1 to 30, 2020     18876        306
Jul  1 to 31, 2020     54843        753
Aug  1 to 31, 2020    122587       1519
Sep  1 to 30, 2020     87430       1907
Oct  1 to 31, 2020     66650       1659
Nov  1 to 30, 2020     48517       1154
Dec  1 to 31, 2020     41139        826
Jan  1 to 31, 2021     49798       1501
Feb  1 to 28, 2021     49080       1511
Mar  1 to 21, 2021     85413        646
```

Try to format the output as neatly as possible as shown in the example above.

HOW TO RUN AND TEST YOUR SOLUTION:
First, you must edit the accompanying source file **MP2-NUMBER.c** to include the C2 header file and the C3, C4 and C5 source files. Fill-up also the missing codes designated as blanks/underscores – see comments with the words TO DO. Thereafter, compile **MP2-NUMBER.c** to produce the corresponding exe file. Run the exe file with I/O redirection as shown in the example below:

```
C:\CCPROG2 > MP2-08 < CHD\PHILIPPINES.TXT > RESULT-PHILIPPINES.TXT
```

where MP2-08.exe is the group's executable file, PHILIPPINES.TXT is the source file, and RESULT-PHILIPPINES.TXT is the destination file. The < (less than symbol) is for input redirection, and the > (greater than symbol) is for output redirection. Replace 08 with your own group number.

The expected output file RESULT-PHILIPPINES.TXT accompanies this specifications document for your reference. Run your own exe file. Your result should be the same as the expected output file. Otherwise, there is a high probability that there is a logical error in your solution, or you did not comply with one or more instructions/specifications.

Test your solution EXHAUSTIVELY by subjecting it to other input data for other countries in the CHD folder. Verify the output using other tools (for example, you can use Excel to check the monthly statistics in Challenge #5).

DELIVERABLES: For Challenge #2 - submit only the header file **C2-NUMBER.h**. For Challenge #3 to #5 - submit a ZIPped file named **MP2-NUMBER.zip**. Make sure to rename the files with your own group NUMBER. When unzipped, the extracted files should be:

1. C2-NUMBER.h
2. C3-NUMBER.c
3. C4-NUMBER.c
4. C5-NUMBER.c
5. MP2-NUMBER.c
6. Redirected output text file named **RESULT-COUNTRY.TXT** using as input data one of the countries that you chose in **C0**. Replace COUNTRY with the actual country name.

More challenges in Part 3 (on file processing)…