CS410C FALL 2019 DUE: 12/9/2019

PROGRAMMING ASSIGNMENT 10P Submit code: 10**P**

Purpose

The purpose of this assignment is to have you work a little bit with everything that you've learnt so far

Scenario

Earthquakes occur all over the world all the time. Though most of the earthquakes are never felt by us, they do get recorded in seismic laboratories all over the world. Seismic data analysts are interested to gather various statistics about these earthquakes and interpret them in some meaningful way to predict future earthquakes. Let us say there's an organization that wants to gather and analyze seismic data for certain regions in the U.S and Canada for a 31-day period in October 2014. Your job as a programmer would be to take in all the data that's given to you and come up with some specific statistics.

Method

To implement this program you will use several appropriate data structures based on what's needed. There is seismic data from 6 different regions coming from 6 input files one for each region, they are all the same format and need to be stored in arrays of structures. A structure needs to be created to hold all the fields for each record in the dataset.

Data

A sample record contains the following data as indicated below:

Year	Month	Day	Time (UTC)	Latitude	Longitude	Magnitude	Depth	Location
2014	10	28	06:10:35	37.041	97.930	5.0	21.3	4-mi-SW-of-Bluff-City,KS

Data Structures

You must define a single type of structure that can hold all of the data fields indicated above for one record. Pay attention to the data type for each item and use appropriate declarations for each data item within the structure. *To store all the data from one data file you will need one array of structures – so if we had just one data file this would be sufficient*.

But we have multiple data files giving us data for multiple regions, therefore you need to create a 2-D array of structures to hold the entire data set. The maximum size of the 2-D array should be able to hold all the data from the largest data file present in your data set. Each row of the 2-D array pertains to data from one region (or one data file) and each cell (column) is capable of holding one record (represented by one structure).

You will also need an array of structures to store the file names and their sizes (a parallel array corresponding to the actual data stored in your first 2-D array of structures). Other than these data structures, you also need to create other data structures as appropriate based on the requirements.

Requirements

1. **[20 points]** The program should initially read all the data and store them in the appropriate structures
2. **[20 points]** A summary of the data in each data file should be printed programmatically to an output file which includes the following pieces of information:

a. Summary heading indicates range of Latitude and Longitude for the data set
b. Total number of earthquakes recorded for the period
c. Range of depth that earthquakes were recorded
d. Range of magnitude of earthquakes
e. Highest recorded magnitude with details of date, time and location

The output file is created and written to programmatically (not redirection of standard output using “>”). The output data file is needed only for this item (2).

3. **[10 points]** The program should display a table that indicates day by day earthquake totals for all the datasets together. In other words a 2D array must be created of size 6x31 (or 6x32 if that’s easier) whose cells contain the total number of earthquakes per day per region.
4. **[10 points]** The program should display a table that indicates totals for magnitude ranges for each region. In other words a 2D array must be created of size 6x10 (if the highest magnitude for any region is 9). You can truncate magnitudes to the nearest integer just for the purposes of this table.
5. **[10 points]** The program should display a table that indicates totals for depth ranges for each region. In other words a 2D array must be created of size 6xA (look at sample output file for ranges to set your value for A for number of columns). You can truncate depths to the nearest integer just for the purposes of this table.
6. **[10 points]** The program should display a table that indicates earthquake totals for each latitude/longitude possible for all 6 regions combined. This will require you to define a 2D array of the appropriate size. Since a lot of the grid area for this table will be empty (since several latitude/longitude combinations will have no data) you can choose to display just the portion of the 2D array (table) that contains data for the regions that we are working with.
7. **[Extra credit 15 points]** Finally we want to try and do some predictions based on some factors. We want to do this prediction only for one specific region. I will give you some factors that will enable you to make this prediction for the region which falls into a specific geographic zone of latitude/longitude. The prediction will be to simply indicate a number between 0 and 10.0 with 10.0 being the highest indicator that a major earthquake is likely to occur in the next 50 years. For the Alaska region here are the factors provided to make your prediction:

A. Ground Uplift and Tilting (25%)
B. Foreshocks (25%)
C. Water level in wells (15%)
D. Unusual Radio waves (25%)
E. Strange animal behavior (10%)
F. Prior earthquake in the area (neighbors)

The table below gives you what factors are present for the range of latitude and longitudes shown. For all cells that fall within a particular range, the corresponding factors will apply. Adding all the factors that apply to a particular cell you will initially come up with an initial prediction number between 0 & 1. Next you apply factor F (above) by averaging the total number of earthquakes that have occurred in the 4 neighboring cells and if this number is greater than 0, multiply it by the initial prediction number to get the actual prediction for that cell. If the initial prediction number is 0, factor F becomes the actual prediction number. If the factor F produces a result of 0, the initial prediction number will become the actual prediction number. If the prediction number happens to be greater than 10, set it to 10.

	Lat 58-59	Lat 60-61	Lat 62-63	Lat 64-65	Lat 66-67
Long 140-145	A	A, B	A,C	C	C
Long 146-150	A,C	B	C	B,C,D	D
Long 151-155	A,B,C	A,B,C,D	C	D	B
Long 156-160	E	E		B,D	C,D

Long 161-165		B,C	C,D	B,C,D,E	
Long 166-170		D			D

The program should display a table that indicates a prediction number in each cell of the 2D array which represents the region in latitude/longitude. You need to make one prediction only (just one final array representing the predictions). You can choose to display just the portion of the 2D array (table) that contains data for the Alaska region.

8. **[20 points]** Your program must be modular and as many functions that are needed to make coding less repetitive, the better. Feel free to create any other data structures for items (3) through (6) to make the code more efficient and organized. Your grade for this assignment will also be based on how well you have accomplished all of this.

All tables that are displayed for items (3) through (7) above must have clear row and column headings for ease of data analysis.

Input

The input data is coming from several data files. There is one data file called data.txt supplied via command line arguments (via argc, argv) that actually contains all the other file names. All input values will come from the individual data files that are listed in data.txt. So you basically have to run the program as: a.out data.txt

The program will open and read the filenames from data.txt (one filename per line in data.txt) and in turn open up each file to process data.

Your program should have the ability to accept multiple arguments on the command line (like a.out data1.txt data2.txt for example).

Output

The output will first summarize the data as indicated above. The Summary data needs to be written to an output file called “summary”. The rest of the output is displayed on standard output. For items 4, 5, 6 I am truncaing the values to integers so if you want your values to match with mine discard the fractional values (for example magnitude 4.1 and 4.7 will both be truncated to 4.0).

Other Details

- The program must be split into multiple files:
    - The main must be the only function in one code file (.c)
    - All the other functions should be in a second code file (.c)
    - The other functions should be forward declared (prototype declarations) in a header file (.h) and included in both code files
    - The definition for constants should all be done in the header file
    - **Be sure to submit all three files!**

- As always test your program carefully and follow good programming style.

- You must follow all the coding style rules as specified in our “coding guidelines”. In particular:
    - You must put your name enclosed in a comment box at the top, and keep any other comments that are already there.

    - Keep lines to the point of making your code easily readable. It is a good idea to make your comment box lines 66 characters long and use this as a guide.
    - You must use good names for any variables you create (a full word that describes what it is there for).
- Details that you do not follow are penalized after other scored items are added up, so even if you got a 100 for the functionality of your program, you can still get a lower score because you did not follow all the other requirements for the assignment.

Submission

Submit this assignment with the code 10P followed by the names of all 3 files:
For example: `submit 10P 10P.c 10Pfunctions.c 10P.h`