CCPROG2 Open Notes, Group Exercise on Arrays of Structures. [Each group can have at most 4 members only]

--This is a group activity. Thus only 1 member need to upload the C source code. Make sure that the content of the file includes the names of the contributing members. Include these names too in the comment of the submission/upload. Prior to uploading, make sure that your team has already Joined in appropriate groups (under People) in Canvas.

1. Define the data structure to represent an array of at most 20 candidates where each contains the name, birthday, position, party, a list of at most 10 bills passed, the number of bills passed, and a rating. Note the following as you define the data structure to represent the model:

   a.) name further consists of the first name, last name, and middle initial
   b.) birthday further consists of the numeric month, day, and year
   c.) each bill passed is a structure that contains the bill and the date it was passed
   c.) rating is further defined to contain the percentage of confidence/possible votes from a survey, the organization who administered the survey, and the date of the survey.
   d.) For data types or lengths of strings not explicitly indicated, please make your own intelligent judgement.

For the succeeding requirements, these are all based on the declaration you defined in #1. Also note that the parameter passed to the function should only be the pertinent information (i.e., do not pass the entire candidate, if only the name is to be modified) unless otherwise explicitly stated.

2. Create a function getName() that will get the name of the candidate.

3. Create a function getDate() that will get the date information.

4. Create the function getInput() to now call function to get input from the user. Note that part of the solution in this function is to call the functions getName() to get the name of the candidate and getDate() to get the birthday and the date the bill was passed.

5. Create a function displayDate() that will display the date information in the format of <Month in word form> <day>, <year>. For example: March 3, 2018

6. Create a function display() that will display all information of 1 candidate. Whenever appropriate, call displayDate().

7. Create a function displayByParty() that will accept as parameter the array of candidates and the string party and will display all candidates (and their corresponding information) who belongs to the same party. This function should call function display() as part of the solution. Allow the user to press a key (of your choice, like Enter) to display the next candidate that is of the same party.

8. Create a function swap() that will accept addresses of 2 candidate structures as parameter. After the function, these addresses will contain the updated values, i.e., the data in both structures will be swapped.

9. Create a function sortByRating() that will rearrange the contents of the array of candidates from the one with the highest rating to the lowest rating. This function will not perform any display (i.e., no printf()). Part of the solution to this function is to call function swap(). Hint: You may use the algorithm to sort an array of floating point values that we discussed before.

10. Create a function sortAlphabetical() that will rearrange the contents of the array of candidates based on the last name of the candidate. This function will not perform any display (i.e., no printf()). Part of the solution to this function is to call function swap().

11. Create the main program such that the user is allowed to navigate through a series of menu options until he chooses to exit. The menu options are:

   a . Add Candidate Info : This option will ask 1 candidate information from the user.

   b. Display All Candidates : This option will display all candidates (and their respective information) in alphabetical order. Thus, in this option, the function sortAlphabetical() should be called. The function display() should also be called as part of the solution. Note that the user is to be asked to press a key (of your choice, like N or Enter) before the information of the next candidate may be shown.

   c. Display By Rating: In this option, the function sortByRating() should first be called. After which, part of the solution should call function display(). The user is also asked to press a key before the next candidate information will be displayed.

   d. Display By Party: Here, the user is asked to input the party to search for. Then, the function displayByParty() is called.

   e. Exit : If this option is chosen, the program then terminates properly. Remember that exit() should not be called.