

In this question, you will explore basic data analysis in Python using a Government of Canada dataset. Governments in Canada (federal, provincial, and municipal) are working together to make data accessible to citizens as a means of increasing government transparency and providing citizens with the opportunity to work with real government data. Here is a [link](#) to the Government of Canada's Open Data portal, where you can explore the different [datasets](#) and see how Canadians have used the data to create and make their own discoveries.

Specific to this question, you will be using an open dataset provided by **Elections Canada**. Here is a [link](#) to the page where the dataset is located. You will be using Table 11. Download the **dataset (.csv format)** first and format it as per the following:

- There are 13 columns in the dataset from where you will remove the 13th column. Also remove the French heading part from each of the column names.
- The remaining 12 columns would be:
  - Province Name
  - Electoral District Name
  - Electoral District Number
  - Population
  - Electors
  - Polling Stations
  - Valid Ballots
  - Percentage of Valid Ballots
  - Rejected Ballots
  - Percentage of Rejected Ballots
  - Total Ballots Cast
  - Percentage of Voter Turnout

### Your task to represent data and develop functionality:

1. Your **python** program should read in the dataset from the CSV file and store the data in a **list of dictionaries**. There will be one dictionary for each electoral district (row of the dataset) and each of the dictionaries will have a corresponding key for each of the 12 pieces of data (column of the dataset). Note that, although this dataset appears to be sorted by electoral district number, it is actually not. You will notice further down in the dataset many electoral district numbers that are out of place.
2. Write a function called **displayInfo** that prints the following information given an electoral district number. The output should be reader friendly.
  - The number of polling stations
  - The number of valid ballots
  - The total ballots cast
  - Percentage of voter turnout
3. Write a function called **uniqueDistricts** that accepts a province name as its parameter and returns a list of the names of the electoral districts in that province. For example, a call to **uniqueDistricts("PrinceEdwardIsland")** would return a list containing the following strings: ["Cardigan", "Charlottetown", "Egmont", "Malpeque"].
4. Write two functions called **findMax** and **findMin** that returns the maximum and minimum value for the supplied key value among any of ("Electors", "Population", "Polling stations", "Valid ballots", "Rejected ballots", "Total ballots"). For example, a call to **findMax("Electors")** would return the maximum number of electors across Canada and the associated Electoral district name. Your function should not use the built-in max and min functions
5. Write a function called **totalVotes** that accepts the list of dictionaries as its parameter and returns a list of dictionaries which consists of the total number of ballots cast in every Canadian province and territory. This means that the list that is returned from the **totalVotes** function will contain 13 dictionaries with two keys (the province/territory name and the total number of ballots cast in the province/territory). To calculate the total number of votes, please use the values of the total ballots cast attribute.
6. Write a function called **selectionSort** that accepts the list of dictionaries, and a key, and then sorts the supplied list of dictionaries in situ (in-place) into increasing order based on the specified key. For example, if the insertion sort function is called with the "Electors" key as its parameter, the dictionaries would be sorted into increasing order by Electors.
7. Write a function called **binarySearch** that searches for an electoral district based on its electoral district number and returns the percentage of voter turnout in that district. This function assumes that the list is already sorted by electoral district number. Your

function should use binary search. You are welcome to use the provided binary search code as a guide - you will need to edit the code to properly search a list of dictionaries. Remember, the dataset is not initially sorted.

8. Create a simple menu that gives the user to choose a menu option and execute the corresponding functionality. Once executed, repeatedly ask the user to choose the menu option further. It should look like the following:
  1. Display information for an electoral district
  2. Show unique districts by the given province
  - .....
  - .....

Now, use all of your functions in a main program called **a3\_q1.py** (don't forget to import your functions file). You should have this main program create the list of dictionaries based on the dataset you preprocessed.

## To Submit:

Submit the following files as a **SINGLE zip file** named **a3\_999999999.zip**:

1. **functions3.py**: A file containing all of your functions.
2. **a3\_q1.py**: the main program for the question
3. **electionsData.txt**: a copy of the dataset file you downloaded and preprocessed.
4. **testing.txt**: test the program by selecting all the menu options and cut-paste the output in a text file.

***NOTE: Updates on output requirements will be posted shortly for compatibility with the autograder.***