IF100 – Spring 2019-2020
Take-Home Exam #4
– Simulating the Elections –
Due May 13th, Wednesday, 23:55 (Sharp Deadline)

Introduction

The aim of this take-home exam is to practice on functions. You are given a main program that uses some functions and for the program to work correctly you should implement those functions. Therefore, you cannot finish this take-home exam without using functions.

For this Take-Home Exam, we share a colab file with you that only contains the main part of the assignment. Please do not modify the last code cell, instead use the previous code cell(s) to implement your own functions. Note that a different main program will be used for grading purposes. If you modify the one given with the take-home exam documentation, then your grade will be affected with respect to the severity of the modification carried out. Therefore, do **not** modify the given main program. However, you still need to upload the entire solution while submitting your take-home exam. That is, your submission should include the function implementations, and also the main program that is provided together with the take-home exam documentation.

You will write a Python program that simulates the local elections in Turkey.

Description

In this take-home exam, you will be simulating a local election system. There will be exactly 3 different political parties. When the program starts, it will first prompt the names of these political parties. A party name may contain any kind of character, except for spaces (" "); however, it should start with a letter that belongs to the English alphabet. If a party name is entered incorrectly, then your program should force the user to re-enter this party name again until the correct format is satisfied by the user. One other important point is that the party names must be unique (case insensitively, means that "*partyA*" will be evaluated as the same with "*PARTyA*"). In other words, if the user enters a party name which already was entered before, then your program should ask for another party name.

- `getPartyName()` is the function to be implemented to get a valid party name as input from the user. This function takes two parameters: (i) a string value, indicating which party name is being asked (i.e. "first", "second", or "third"), and (ii) a list value, holding the names of the valid parties taken as input until so far so that the uniqueness of the party names can be validated within this function. In this function, you should be

getting party input from the user, then check its correctness and uniqueness as well. As long as the input is not correct nor unique, then you should be asking for another party name until you get a correct and unique input. After you get this correct and unique input, do not forget to add this input into the list of parties.

After all the 3 party names are correctly entered by the user, your program should display a menu of 4 different options, which are further explained below. The user of your program should select one of these options ("1", "2", "3" or "4") as each option will make the user have different processes, which are also further explained below. The program should not terminate until the user enters "4" as the menu option. If the user enters an input other than "1", "2", "3" or "4", your program should display an error message and ask for a new option input.

- `getMenuOption()` is the function to be implemented to display the menu and get a valid option from the user. This function does not take any parameters, but it should return the option selected by the user (as a string) so that the processing can continue correctly.

Storing the Dataset

With the use of the option menu, your program will get the results for the given parties in different cities. This data has to be stored somewhere in a compact way so that your program can process it correctly for all the options of the menu. `citiesAndResults` is the variable in which you will store this data. The type of this variable is a list, but the type of the items of this list, i.e. in what way your program will store this data, purely depends on your algorithm. This is the tricky part of the take-home exam, you should think hard on the algorithm before anything else.

Menu Options

1. Enter results for a new city: If the user selects the option "1" from the menu, then your program will get the upcoming results for a new city. For this purpose, your program will first prompt a city name. A city name can only have letters (both of the lowercase and uppercase letters of English alphabet), therefore you need to do an input check on the city name. If a city name is entered incorrectly, your program should enforce the user to re-enter a valid city name until the correct format is satisfied. Moreover, the user cannot enter a city name which has already been entered before. In such a case, your program should give an error message and return back to displaying the menu.

   If the user enters a new valid city name, then your program should ask for the election results for this specific city. These results should be entered in the format of `party1Score-party2Score-party3Score` such as

`30-15.45-45`. Hence, your program should get 3 numbers (there may be real numbers), each separated by a single dash character ("-"). You may assume that the user will enter this input in a correct format. When the required information is obtained from the user, your program should return to the main menu.

- ➢ `getNewResults()` is the function to be implemented to handle getting new election results for a new city. This function takes a single parameter, a list value storing the entire dataset of the results. After getting the results for a particular city, do not forget to modify the list that you store all the information.

2. Change results of a specific city: If the user selects the option "2" from the menu, then your program will change the election results for a specific city that the user has already entered before. To manage that, your program will first prompt a city name. As in the case with option "1", a city name can only have letters (both of the lowercase and uppercase letters of English alphabet). If a city name is entered incorrectly, your program should enforce the user to re-enter a valid city name until the correct format is satisfied. Once a valid city name is entered by the user, your program should then check whether the user has entered the results of this city before or not. Obviously, you cannot change the result of a city if you don't know any information about it. In this case, your program will return back to the main menu with an appropriate message and without doing any modification. Otherwise, i.e. if the results exist for the given city, then your program should ask for the new results for the given city. Again you may assume that the new result information will be entered correctly by the user. Consequently, your program will update the election results and return back to the main menu again.

- ➢ `changeResults()` is the function to be implemented to handle changing the election results of a city, for which some data is entered before. This function takes a single parameter, a list value storing the entire dataset of the results. In case of a valid change, do not forget to update the list that you store all the information.

3. Show current results: If the user selects the option "3" from the menu, then your program will display the current election results for each city as follows: In every line, your program should display the information of one specific city, for which the data is entered before, in the format given below.

```
cityName party1Score party2Score party3Score X is/are leading
```

Here, `X` represents the name(s) of the leader party for this city. If there is

more than one party having the same highest score, then your program should display all of them in the order of the parties entered by the user. Additionally, all the city names and party names appearing in the output should be in lowercase, except for the first letter.

- ➢ `showResults()` is the function to be implemented to handle displaying the current election results. This function takes two parameters: (i) a list value storing the entire dataset of the results, and (ii) another list value storing the names of the parties.

4. Exit: If the user selects the option "4" from the menu, then your program will terminate with an appropriate message.

Please note that all of the names inputted during the program should be case-insensitive; meaning that "Izmir" and "izMir" and "izmir" are equal.

In the previous part of the document, we mentioned that it is up to you how to keep `citiesAndResults` variable. Let us give you an example at this stage of the document. For instance, if two (2) cities were entered with their results, then `citiesAndResults` would appear as follows:

```
[["Istanbul", 35, 34, 31], ["Izmir", 10, 60.5, 29.5]]
```

As you can see, `citiesAndResults` becomes a list of lists, and each sublist indicates a result for a particular city. In each sublist, you can find the city name, the score for the first political party for that city, the score for the second political party for that city, and the score for the third political party for that city respectively. As we said, this is just an example of how you can store the information. You may prefer to keep all the information in `citiesAndResults` in a different way like given below:

```
[[35, 34, 31, "Istanbul"], [10, 60.5, 29.5, "Izmir"]] or

["Istanbul_35_34_31", "Izmir_10_60.5_29.5"] or

["Istanbul_35_34_31;Izmir_10_60.5_29.5"]
```

You may prefer a completely different way to store your data (it depends on your algorithm).

In this take-home exam, you are given a main program, which aims at solving the above-described problem. This main program uses some functions, which are not provided. Your aim is to implement these functions, so that the program can work correctly.

You should not change the cell that contains the main program in the provided file. We will use another main program for grading purposes.

Therefore, any change on the given template main program may (and possibly will) result in grade reduction.

The inputs of the program and their order are explained above. It is extremely important to follow this order with the same format since we automatically process your programs. Also, prompts of the input statements to be used have to be exactly the same as the prompts of the "Sample Runs". ***Thus, your work will be graded as 0 unless the order is entirely correct***.

You **must** implement **five** functions to make the given main program work. These functions are explained in this section together with the inputs and outputs. You must have those functions in your program but if you want, you are also welcome to use some other additional functions.

You may check the Sample Run given below for further information.

Sample Runs

Below, we provide some sample runs of the program that you will develop. The *italic* and **bold** phrases are inputs taken from the user. You have to display the required information in the same order and with the same words and characters as below.

Sample Run 1

```
Welcome to this virtual local elections Python program.

Please enter the name of the first party: 1party
Invalid party name. Please enter again: party 1
Invalid party name. Please enter again: *party
Invalid party name. Please enter again: party1
Please enter the name of the second party: +p2
Invalid party name. Please enter again: PARTY1
Party1 already exists. Please enter again: PartY1
Party1 already exists. Please enter again: party2
Please enter the name of the third party: 100if
Invalid party name. Please enter again: party2
Party2 already exists. Please enter again: PARTY2
Party2 already exists. Please enter again: party 2
Invalid party name. Please enter again: p

Menu:
1. Enter results for a new city.
2. Change results of a specific city.
3. Show current results.
```

```
4. Exit.

Select one of the options (1-4): 3
No information available yet...

Menu:
1. Enter results for a new city.
2. Change results of a specific city.
3. Show current results.
4. Exit.

Select one of the options (1-4): 5
You have entered an invalid input for menu selection. Please enter again:
abc
You have entered an invalid input for menu selection. Please enter again: 0
You have entered an invalid input for menu selection. Please enter again:
-10
You have entered an invalid input for menu selection. Please enter again: 4
**********
Terminating, good bye dear citizen!!! Do not forget to vote in elections :)
```

Sample Run 2

```
Welcome to this virtual local elections Python program.

Please enter the name of the first party: p1
Please enter the name of the second party: p2
Please enter the name of the third party: p3

Menu:
1. Enter results for a new city.
2. Change results of a specific city.
3. Show current results.
4. Exit.

Select one of the options (1-4): 1
Please enter the name of the city: ist34
Invalid city name. Please enter again: 34ist
Invalid city name. Please enter again: ist anbul
Invalid city name. Please enter again: istanbul
Please enter results for Istanbul: 10-20-70

Menu:
1. Enter results for a new city.
2. Change results of a specific city.
3. Show current results.
4. Exit.

Select one of the options (1-4): 1
Please enter the name of the city: ISTANBUL
```

You cannot enter Istanbul since this city has already been entered. If you want you can update the results of this city by using the 2nd menu option.

Menu:
1. Enter results for a new city.
2. Change results of a specific city.
3. Show current results.
4. Exit.

Select one of the options (1-4): ***1***
Please enter the name of the city: ***ist***
Please enter results for Ist: ***60-30-10***

Menu:
1. Enter results for a new city.
2. Change results of a specific city.
3. Show current results.
4. Exit.

Select one of the options (1-4): ***1***
Please enter the name of the city: ***izmir***
Please enter results for Izmir: ***60-30-10***

Menu:
1. Enter results for a new city.
2. Change results of a specific city.
3. Show current results.
4. Exit.

Select one of the options (1-4): ***3***
Istanbul 10.0 20.0 70.0 --> P3 is leading
Ist 60.0 30.0 10.0 --> P1 is leading
Izmir 60.0 30.0 10.0 --> P1 is leading

Menu:
1. Enter results for a new city.
2. Change results of a specific city.
3. Show current results.
4. Exit.

Select one of the options (1-4): ***1***
Please enter the name of the city: ***ankara***
Please enter results for Ankara: ***40-20-30***

Menu:
1. Enter results for a new city.
2. Change results of a specific city.
3. Show current results.
4. Exit.

```
Select one of the options (1-4): 3
Istanbul 10.0 20.0 70.0 --> P3 is leading
Ist 60.0 30.0 10.0 --> P1 is leading
Izmir 60.0 30.0 10.0 --> P1 is leading
Ankara 40.0 20.0 30.0 --> P1 is leading

Menu:
1. Enter results for a new city.
2. Change results of a specific city.
3. Show current results.
4. Exit.

Select one of the options (1-4): 4
**********
Terminating, good bye dear citizen!!! Do not forget to vote in elections :)
```

Sample Run 3

```
Welcome to this virtual local elections Python program.

Please enter the name of the first party: inanc
Please enter the name of the second party: ada
Please enter the name of the third party: duygu

Menu:
1. Enter results for a new city.
2. Change results of a specific city.
3. Show current results.
4. Exit.

Select one of the options (1-4): 1
Please enter the name of the city: bursa
Please enter results for Bursa: 22.5-30.5-33.5

Menu:
1. Enter results for a new city.
2. Change results of a specific city.
3. Show current results.
4. Exit.

Select one of the options (1-4): 1
Please enter the name of the city: cANAkkale
Please enter results for Canakkale: 40.5-20.8-35.2

Menu:
1. Enter results for a new city.
2. Change results of a specific city.
3. Show current results.
4. Exit.
```

Select one of the options (1-4): ***1***
Please enter the name of the city: ***AdaNA***
Please enter results for Adana: ***44.5-55.5-0***

Menu:
1. Enter results for a new city.
2. Change results of a specific city.
3. Show current results.
4. Exit.

Select one of the options (1-4): ***3***
Bursa 22.5 30.5 33.5 --> Duygu is leading
Canakkale 40.5 20.8 35.2 --> Inanc is leading
Adana 44.5 55.5 0.0 --> Ada is leading

Menu:
1. Enter results for a new city.
2. Change results of a specific city.
3. Show current results.
4. Exit.

Select one of the options (1-4): ***2***
Please enter the name of the city: ***13bur***
Invalid city name. Please enter again: ***bur***
You cannot change the result of Bur since no result has been entered for this city yet.

Menu:
1. Enter results for a new city.
2. Change results of a specific city.
3. Show current results.
4. Exit.

Select one of the options (1-4): ***2***
Please enter the name of the city: ***Ada***
You cannot change the result of Ada since no result has been entered for this city yet.

Menu:
1. Enter results for a new city.
2. Change results of a specific city.
3. Show current results.
4. Exit.

Select one of the options (1-4): ***2***
Please enter the name of the city: ***canakKALE***
Please enter results for Canakkale: ***33.3-33.3-33.3***

Menu:
1. Enter results for a new city.

```
2. Change results of a specific city.
3. Show current results.
4. Exit.

Select one of the options (1-4): 2
Please enter the name of the city: BurSA
Please enter results for Bursa: 31-31-30

Menu:
1. Enter results for a new city.
2. Change results of a specific city.
3. Show current results.
4. Exit.

Select one of the options (1-4): 3
Bursa 31.0 31.0 30.0 --> Inanc Ada are leading
Canakkale 33.3 33.3 33.3 --> Inanc Ada Duygu are leading
Adana 44.5 55.5 0.0 --> Ada is leading

Menu:
1. Enter results for a new city.
2. Change results of a specific city.
3. Show current results.
4. Exit.

Select one of the options (1-4): 4
**********
Terminating, good bye dear citizen!!! Do not forget to vote in elections :)
```

How to get help?

You can use GradeChecker (http://sky.sabanciuniv.edu:8080/GradeChecker/) to check your expected grade. Just a reminder, you will see a character ¶ which refers to a newline in your expected output.

What and where to submit?

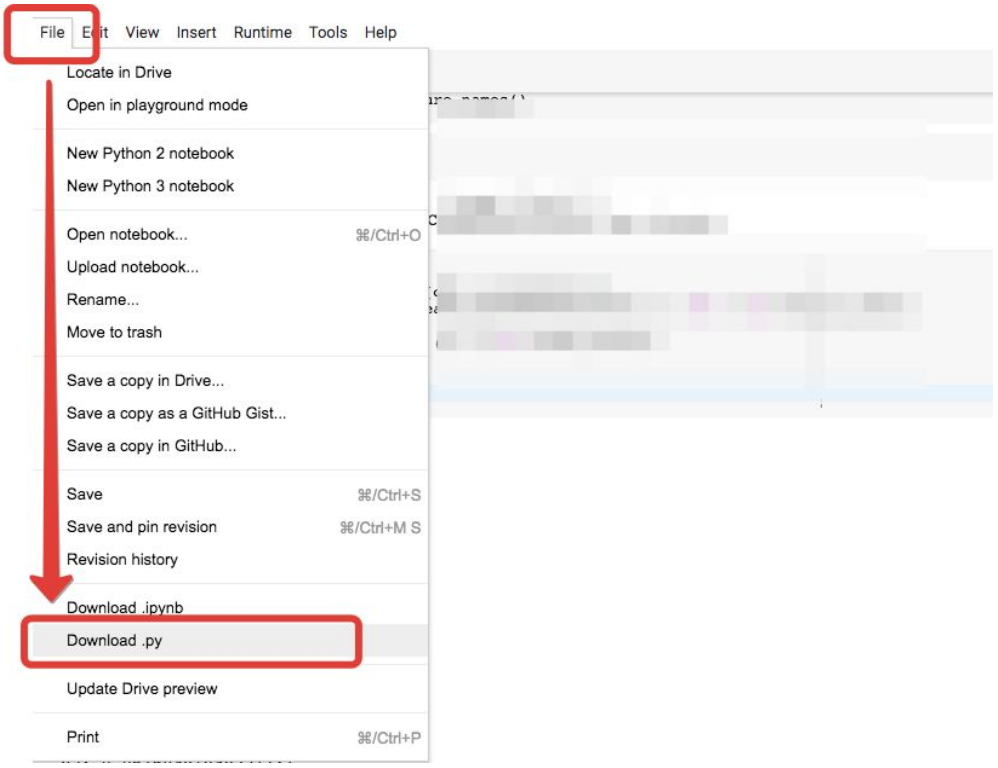
You should prepare (or at least test) your program using Python 3.x.x. We will use Python 3.x.x while testing your homework.

It'd be a good idea to write your name and lastname in the program (as a comment line of course). Do not use any Turkish characters anywhere in your code (not even in comment parts). If your name and last name is "İnanç Arın", and if you want to write it as comment; then you must type it as follows:

 # Inanc Arin

Submission guidelines are below. Since the grading process will be automatic, students are expected to strictly follow these guidelines. If you do not follow these guidelines, your grade will be 0.

- Download your code as *py* file with "File" -> "*Download .py*" as below:



- Name your *py* file that contains your program as follows:

  "**username_the4.py**"

  For example: if your SuCourse username is "**duygukaltop**", then the name of the *py* file should be: **duygukaltop_the4*.py*** (please only use lowercase letters).

- Please make sure that this file is the latest version of your homework program.

- Submit your work **through SUCourse only**! You can use the GradeChecker only to see if your program can produce the correct outputs both in the correct order and in the correct format. It will not be considered as the official submission. You must submit your work to SUCourse.

- If you would like to resubmit your work, you should first remove the existing file(s). This step is very important. If you do not delete the old file(s), we will receive both files and the old one may be graded.

General Homework Rules

- Successful submission is one of the requirements of the homework. If, for some reason, you cannot successfully submit your homework and we cannot grade it, your grade will be 0.
- There is NO late submission. You need to submit your homework before the deadline. Please be careful that SUCourse time and your computer time may have a 1-2 minutes differences. You need to take this time difference into consideration.
- Do NOT submit your homework via email or in hardcopy! SUCourse is the only way that you can submit your homework.
- If your code does not work because of a syntax error, then we cannot grade it; and thus, your grade will be 0.
- Please do submit your **own** work only. It is really easy to find out "similar" programs!
- Plagiarism will not be tolerated. Please check our plagiarism policy given in the syllabus of the course.

Good Luck!
Pınar Ön & Tunal Hamzaoğlu & IF100 Instructors
… AND… Deniz Naz Bayram