

IF100 – Spring 2021-2022
Take-Home Exam 3
Due May 12nd, 2022, Thursday, 23:59 (Sharp Deadline)

Introduction

The aim of this take-home exam is to practice on loops (for statements). The use of loops is due to the nature of the problem; that is, you cannot finish this take-home exam without using loop statements.

Description

Assume that we work in an earthquake observatory, and record every earthquake that has occurred all along throughout the country. Before or after every earthquake, the probability of the occurrence of foreshocks and aftershocks increases. Because faults are not straight, smooth boundaries, ground shaking often occurs both before and after the main quake. Foreshocks are the energy release and ground shaking before an earthquake, and aftershocks are the energy release and ground shaking after an earthquake. Aftershocks can be used as "echo sounders" to study the local structure of the earth. Seismologists and geologists can find the orientation of the fault plane, which helps enormously in characterizing the earthquake, and the stresses and strains within the Earth that caused it.

We are asked to write a Python program that will take from the user a database together with the name of the main earthquake and a radius (in km) from the center of the main earthquake, and display if there existed foreshocks and aftershocks within the 24 hours and within the target radius.

In this take-home exam, you will implement a Python program that will get three (3) inputs from the user:

- The first input to your program is the dataset. This information contains (i) names of the earthquakes (main, foreshock, aftershock doesn't matter), (2) time of the earthquakes (in unix timestamp format - check [this link](#) to get details about unix timestamp), and (3) latitude and longitude of the earthquakes (location of the earthquake on the Earth). The name of the earthquake is separated from the corresponding other information with a colon (:). Each corresponding information related with the earthquake is separated from one another by a comma (,), and the earthquakes are separated by a semicolon (;).

- The second input to your program is the name of the earthquake that your program will look for its foreshocks and aftershocks.
- The third input to your program is the maximum distance from the target earthquake (second input) that you want to search the foreshocks and aftershocks for (radius in km).

Your program will then do some calculations and print an appropriate output according to the conditions listed below:

- If the target earthquake is not in the database, then your program should print out an error message.
- If there are no foreshocks or aftershocks for the target earthquake, within the target range and within 24 hours, then your program should print out an informative message.
- If there are at least one foreshock(s) or aftershock(s) for the target earthquake, within the target range and within 24 hours, then your program should print out each foreshock/aftershock one by one with their name, latitude and longitude information in the same order as they exist in database.

You can find the details about the inputs and outputs in the following section.

Input, Process and Output

The inputs of the program and their order are explained below. 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.**

Your program will have multiple inputs and the specifications for these inputs are explained below.

- database: The name of the earthquake, time in unix timestamp format, latitude of the earthquake and longitude of the earthquake.

eqname1:timestamp1,latitude1,longitude1;eqname2:timestamp2,latitude2,longitude2;...;eqnameN:timestampN,latitudeN,longitudeN

You may assume that the database will be given correctly and there will be no colon (":"), comma (",") or semicolon (";") characters used in the earthquake names.

- You may assume that there will be only one colon (":") after the earthquake name and only one comma (",") between the elements of the earthquake. You may also assume that only one semicolon (";") comes after each and every earthquake (beside the last one).
 - There will not be any colon, semicolon or comma characters in the beginning or at the end of the input.
 - You don't need to perform any input checks for this input.
- targetEarthquake: The name of the earthquake that you want to check:

targetEarthquake

- targetRange: Radius of the searchfield (in km).

targetRange

- You may assume that the user will enter a positive real value.

After you get all the inputs, your program needs to perform a validity check to make sure that the target earthquake exists (which is **case sensitive**) in the database. If not, then, your program should output;

"targetEarthquake is not in the database."

If the target earthquake exists in the database, then, your program should make some checks to find foreshocks and aftershocks for the target earthquake. To check if an earthquake exists within the target range or not, we will use the Euclidean distance metric and multiply the outcome result by 100, as given below, where "*lon*" is longitude and "*lat*" is latitude information of the earthquake.

$$D_{dist\ from\ A\ to\ B} = 100 \times \sqrt{(lon_A - lon_B)^2 + (lat_A - lat_B)^2}$$

To check if an earthquake exists within 24 hours, your program will be using the timestamp information. Keep in mind that, when we increase the timestamp by 1, the result reflects a second later the timestamp, while when we increase the timestamp by 60, it is a minute after the timestamp.

If there are no foreshocks or aftershocks for the target earthquake, then your program should output;

"There were no foreshocks and aftershocks after the earthquake
targetEarthquake."

If there are foreshocks or aftershocks for the target earthquake, then, your program should output;

"For the earthquake *targetEarthquake*, following(s) are the foreshocks and aftershocks:

*** Earthquake *earthquake1* with latitude *latitude1* and longitude *longitude1*.**

*** Earthquake *earthquake2* with latitude *latitude2* and longitude *longitude2*.**

.

.

.

*** Earthquake *earthquakeN* with latitude *latitudeN* and longitude *longitudeN*."**

Keep in mind that you need to print out every foreshock and aftershock that exists in the database **one by one** with the same order as they exist in the database. You need to print 2 decimals after the point in latitude and longitude. **The dashes must be completely aligned with the title which means that they need to have the same number of characters. So if there are 25 characters in the title, then you need to print 25 characters. If there are 30 characters in the title, then you need to print 30 characters.**

Please see the "Sample Runs" section for some examples.

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

```
Please enter the dataset: KURUTILEK:-
947289600,39.8,39.51;BASISKELE:934848000,40.76,29.97;KORFEZ:934876800,40.75,29.75;YEMLI
CE:1319328000,38.7212,43.411;DEREBEY:1319371200,38.7953,43.505;GEDIKBULAK:1319400000,
38.81,43.3;HALKALI:1319529600,38.837,43.3102
Please enter the earthquake that you want to look for: kurutIlek
Please enter the radius (in km): 30
kurutIlek is not in the database.
```

Sample Run 2

```
Please enter the dataset: KURUTILEK:-
947289600,39.8,39.51;BASISKELE:934848000,40.76,29.97;KORFEZ:934876800,40.75,29.75;YEMLI
CE:1319328000,38.7212,43.411;DEREBEY:1319371200,38.7953,43.505;GEDIKBULAK:1319400000,
38.81,43.3;HALKALI:1319529600,38.837,43.3102
Please enter the earthquake that you want to look for: KURUTILEK
Please enter the radius (in km): 30
There were no foreshocks and aftershocks after the earthquake KURUTILEK.
```

Sample Run 3

```
Please enter the dataset: KURUTILEK:-
947289600,39.8,39.51;BASISKELE:934848000,40.76,29.97;KORFEZ:934876800,40.75,29.75;YEMLI
CE:1319328000,38.7212,43.411;DEREBEY:1319371200,38.7953,43.505;GEDIKBULAK:1319400000,
38.81,43.3;HALKALI:1319529600,38.837,43.3102
Please enter the earthquake that you want to look for: BASISKELE
Please enter the radius (in km): 30
For the earthquake BASISKELE; following(s) are the foreshocks and aftershocks:
-----
* Earthquake KORFEZ with latitude 40.75 and longitude 29.75.
```

Sample Run 4

```
Please enter the dataset: KURUTILEK:-
947289600,39.8,39.51;BASISKELE:934848000,40.76,29.97;KORFEZ:934876800,40.75,29.75;YEMLI
CE:1319328000,38.7212,43.411;DEREBEY:1319371200,38.7953,43.505;GEDIKBULAK:1319400000,
38.81,43.3;HALKALI:1319529600,38.837,43.3102
Please enter the earthquake that you want to look for: YEMLICE
```

Please enter the radius (in km): **30**

For the earthquake YEMLICE; following(s) are the foreshocks and aftershocks:

* Earthquake DEREBEY with latitude 38.80 and longitude 43.51.

* Earthquake GEDIKBULAK with latitude 38.81 and longitude 43.30.

Sample Run 5

Please enter the dataset: **KURUTILEK:-**

947289600,39.8,39.51;BASISKELE:934848000,40.76,29.97;KORFEZ:934876800,40.75,29.75;YEMLICE:1319328000,38.7212,43.411;DEREBEY:1319371200,38.7953,43.505;GEDIKBULAK:1319400000,38.81,43.3;HALKALI:1319529600,38.837,43.3102

Please enter the earthquake that you want to look for: **YEMLICE**

Please enter the radius (in km): **12**

For the earthquake YEMLICE; following(s) are the foreshocks and aftershocks:

* Earthquake DEREBEY with latitude 38.80 and longitude 43.51.

Sample Run 6

Please enter the dataset: **KURUTILEK:-**

947289600,39.8,39.51;BASISKELE:934848000,40.76,29.97;KORFEZ:934876800,40.75,29.75;YEMLICE:1319328000,38.7212,43.411;DEREBEY:1319371200,38.7953,43.505;GEDIKBULAK:1319400000,38.81,43.3;HALKALI:1319529600,38.837,43.3102

Please enter the earthquake that you want to look for: **HALKALI**

Please enter the radius (in km): **30**

There were no foreshocks and aftershocks after the earthquake HALKALI.

Sample Run 7

Please enter the dataset: **KURUTILEK:-**

947289600,39.8,39.51;BASISKELE:934848000,40.76,29.97;KORFEZ:934876800,40.75,29.75;YEMLICE:1319328000,38.7212,43.411;DEREBEY:1319371200,38.7953,43.505;GEDIKBULAK:1319400000,38.81,43.3;HALKALI:1319529600,38.837,43.3102;UGUR:947289601,40.74,31.21

Please enter the earthquake that you want to look for: **UGUR**

Please enter the radius (in km): **30**

There were no foreshocks and aftershocks after the earthquake UGUR.

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 take-home exam. Let us repeat,

- You must use Google Colab to develop your code from scratch (from beginning till the end), and then submit it **through SUCourse+ only**! Once you are done with developing your code on Google Colab, then you will copy your code to the CodeRunner to see if your program can produce the correct outputs. At the end, you will submit your code through CodeRunner (and SUCourse+). You should keep your Google Colab file until the end of the semester, we might want to look at this. If you fail to provide this Google Colab file anytime in the semester, you may not earn any credits from this Take Home Exam.
- In the CodeRunner, there are some visible and invisible (hidden) test cases. You will see your final grade (including hidden test cases) before submitting your code. Thus, it will be possible to know your THE grade before submitting your solution.
- **There is no re-submission.** You don't have to complete your task in one time, you can continue from where you left last time but you should not press submit before finalizing it. Therefore, you should make sure that it's your final solution version before you submit it.

General Take-Home Exam Rules

- Successful submission is one of the requirements of the take-home exam. If, for some reason, you cannot successfully submit your take-home exam and we cannot grade it, your grade will be 0.
- There is NO late submission. You need to submit your take-home exam before the deadline. Please be careful that SUCourse+ time and your computer time may have 1-2 minutes differences. You need to take this time difference into consideration.
- Do NOT submit your take-home exam via email or in hardcopy! SUCourse+ is the only way that you can submit your take-home exam.

- If your code does not work because of a syntax error, then we cannot grade it; and thus, your grade will be 0.
- Please submit your **own** work only. It is really easy to find "similar" programs!
- Plagiarism will not be tolerated. Please check our plagiarism policy given in the syllabus of the course.

Good luck!
Özgün Yargı
& IF100 Instructors