CENG 202: DATA STRUCTURES

Programming Assignment 1

Deadline: WEDNESDAY, 29 MARCH 2023 by 20:00

Rules

1. You will use Java to implement the tasks
2. You will lose 20 points for each day your assignment is submitted late.
3. No deadline extensions will be granted.
4. Your code must be developed individually. You cannot collaborate with your peers. You cannot post your code on any public repository, nor can you share it with anybody. Any violations to this code of conduct will be treated as academic misconduct.
5. Submitted code will be automatically checked using tools that detect plagiarism.
6. In case cheating is detected, the grade of this assignment will be zero and a further penalty of 50 points will be applied to the next homework assignment.
7. You may be asked to explain your code and rewrite some parts of it in front of the instructor.
8. Submit your codes to AYBUZEM.
9. Write your name as a comment line in each file you submit.
10. In order to be accepted, your submission
    - must be of text format with a '.java' extension (no Word or other formats and no archives will be accepted)
    - You will receive a zero grade if you do not submit the assignment or you do not adhere to this rule!

Task:
In this assignment, you will implement an Earthquake Notification System. Your program will receive input from two input files (In real life these would have been real time data streams, but you will implement a discrete time simulation using static files.) The first file (`input.txt`) provides information on the location, time, and magnitude, and depth of the earthquakes in or around Türkiye. The second file (`request.txt`) contains user requests.

A sample content of `input.txt`, which was taken from KandilliRasathanesi, is as follows:

```
HATAY 19.03.2023 17:25:57 Magnitude: 3.2 Depth: 5.1 km
KAHRAMANMARAS 19.03.2023 17:18:28 Magnitude: 3.3 Depth: 5.0 km
KAHRAMANMARAS 18.03.2023 16:07:55 Magnitude: 3.2 Depth: 5.5 km
SURIYE 18.03.2023 15:43:39 Magnitude: 3.9 Depth: 6.4 km
MALATYA 16.03.2023 23:37:33 Magnitude: 3.0 Depth: 5.4 km
AKDENIZ 16.03.2023 22:59:12 Magnitude: 4.6 Depth: 7.6 km
KIBRIS_RUM_KESIMI 15.03.2023 09:27:13 Magnitude: 3.9 Depth: 7.8 km
HATAY 14.03.2023 23:42:28 Magnitude: 3.5 Depth: 5.0 km
KAHRAMANMARAS 14.03.2023 23:30:37 Magnitude: 3.0 Depth: 5.0 km
ADIYAMAN 14.03.2023 21:55:15 Magnitude: 4.0 Depth: 5.0 km
ELAZIG 14.03.2023 20:00:34 Magnitude: 3.2 Depth: 5.7 km
YUNANISTAN 14.03.2023 19:36:02 Magnitude: 5.5 Depth: 20.1 km
ADANA 14.03.2023 14:57:55 Magnitude: 3.6 Depth: 5.0 km
ADIYAMAN 13.03.2023 20:37:27 Magnitude: 3.3 Depth: 2.6 km
AKDENIZ 13.03.2023 09:17:43 Magnitude: 3.2 Depth: 25.2 km
MALATYA 12.03.2023 23:55:10 Magnitude: 3.4 Depth: 5.0 km
GAZIANTEP 12.03.2023 22:41:41 Magnitude: 3.4 Depth: 5.7 km
SURIYE 12.03.2023 16:54:20 Magnitude: 3.6 Depth: 2.9 km
AKDENIZ 12.03.2023 16:36:46 Magnitude: 3.6 Depth: 3.2 km
BOLU 12.03.2023 16:31:58 Magnitude: 3.7 Depth: 5.0 km
```

In each line, the earthquake is defined as follows:
`[LOCATION] [DATE given as DD.MM.YYYY] [TIME given as HH:MM:SS] [Magnitude: DoubleValue] [Depth: DoubleValue km]`

To store the information given in this file, you'll implement a doubly linked list where each node has the location, date, time, magnitude, and depth of an earthquake. The doubly linked list will have one node for each earthquake given in a separate line, i.e., if there are `N` lines in this file, the doubly linked list will have `N` nodes. To implement this, **skeleton code** is shared via AYBUZEM. Please do not change the class names.

The second input file (`request.txt`) will be a series of requests by the users of the notification system. The users may request adding/removing an earthquake, or query for earthquakes. Each query can only have a single attribute and a condition. For the add and delete requests there should be all the information about an earthquake. A sample content of `request.txt` is as follows:

```
Query magnitude > 5.1
Query depth > 3.1
Query magnitude < 6.0
Query depth = 25.2
Query location = BOLU
Query date = 19.03.2023
Query date = 11.03.2023
Add MALATYA 11.03.2023 00:44:51 Magnitude: 3.3 Depth: 5.0 km
Query date = 11.03.2023
Delete HATAY 19.03.2023 17:25:57 Magnitude: 3.2 Depth: 5.1 km
Query date = 19.03.2023
Query all
```

For **magnitude** and **depth** attributes, there can only be (> , <, =) operators
For **location** and **date** attributes, there can only be (=) operator
You don't need to write cases for **time** attributes.
You need to print the "-" sign after each query.
Example output is given in the Java file. The output must be exactly matching.
You should read the Earthquakes and display them correctly. Otherwise you won't be able to get grades from the queries.

The explanation of each line is given below.

- `Query magnitude > 5.1:` All earthquakes whose magnitudes are greater than 5.1 are printed to screen. Since there is only one such earthquake in this example, your code should print the following line:
  `YUNANISTAN 14.03.2023 19:36:02 Magnitude: 5.5 Depth: 20.1 km`
- `Query depth > 3.1:` All earthquakes whose depths are greater than 3.1 are printed to screen.
- `Query magnitude < 6.0:` All earthquakes whose magnitudes are less than 6.0 are printed to screen. (For this example, you'll print all earthquakes in the doubly linked list since all their magnitudes are less than 6.0)

- `Query depth = 25.2:` All earthquakes whose depths are equal to 25.2 are printed to screen. (There is a single earthquake with depth being 25.2 in this example)
- `Query location = BOLU:` All earthquakes whose locations are BOLU are printed to screen. (There is a single earthquake with location being BOLU in this example)
- `Query date = 19.03.2023:` All earthquakes which occurred on 19.03.2023 are printed to screen. (There are two such earthquakes)
- `Query date = 11.03.2023:` All earthquakes which occurred on 11.03.2023 are printed to screen. (You'll print NONE since there is no earthquake occurred on 11.03.2023)
- `Add MALATYA 11.03.2023 00:44:51 Magnitude: 3.3 Depth: 5.0 km:` A new earthquake with the given information is added to the beginning of the doubly linked list. You need to print all elements of the list after the add operation.
- `Query date = 11.03.2023:` All earthquakes which occurred on 11.03.2023 are printed to screen. (Now, there is a single earthquake occurred on 11.03.2023)
- `Delete HATAY 19.03.2023 17:25:57 Magnitude: 3.2 Depth: 5.1 km:` An existing earthquake with the given information is removed from the doubly linked list. If there is no such earthquake, then the linked list will not be changed. You need to print all elements of the list after the delete operation.
- `Query date = 19.03.2023:` All earthquakes which occurred on 19.03.2023 are printed to screen. (After the previous Delete operation, there is a single earthquake occurred on 19.03.2023)
- `Query all:` All earthquakes in the doubly linked list are printed to screen.

There should be an **output** file (`output.txt`) that will be a series of responses to the queries. A sample content of `output.txt` is given in the skeleton code file. Output should be **exactly** the same.

Submission:
In this assignment, you are going to submit a **single java file** which contains Earthquake, EarthquakeLinkedList and Test classes. You won't submit your output.txt file.