TOBB ETU COMPUTER ENGINEERING DEPARTMENT

BIL 334 Formal Languages and Automata

HOMEWORK 3 [100 Points]

Due Date: November 30, 23:59

In this homework, you are expected to parse a web server log file using python and regular expressions. For all of the parsing and string manipulation (i.e, split, replace, find, contains) jobs, you must use the **re** (import re, regular expressions) package. You are **NOT** allowed to use any other string manipulation methods.

We are providing you with a small subset of the grading logs called **web_log.txt** under the "Homework Solutions" in your piazza class so you can test your code. In all of the tasks you should expect your input to be in the same form as this file.

1 EXTRACT IP ADDRESSES AND DATES [40 POINTS]

In this task, you are asked to extract IP Addresses and Dates from the server logs. However, since the existing dates have unnecessary information we want you to convert the format as well. You should remove hour, minute and second part of the date and replace **'/'** characters of the date with **'-'**.

Example Input:

```
54.36.148.10 - - [22/Jan/2019:03:56:58 +0330] ...
5.211.97.39 - - [22/Jan/2019:03:56:58 +0330] ...
```

Console Command:

```
python task1.py
```

Example Output:

```
54.36.148.10 22-Jan-2019
5.211.97.39 22-Jan-2019
```

Your python script should have the name **task1.py** and should read its inputs from a file called **log_task1.txt** under the same directory and write it's results to a file called **output_task1.txt**.

2 Find Most Accesses in Time Range [60 Points]

We would like to run a giveaway for an arbitrary product in the future. We will do this by specifying a resource in our website and looking at which IP address accessed it the most for a given time period. You can assume that we can associate IP addresses with individuals for this task.

You are asked to parse the logs file given a **date** and a **duration in seconds** for a **resource** and expected to find the (IP Address, Access Count, First Access Time) tuples for the given time period in **descending access count order**. You must ignore requests that don't make a GET request to the specified resource. Your program should accept these parameters from the command line, so invocation of your program will be in the following form.

python task2.py <date> <duration> <resource>

To calculate the Access Time (in seconds) of the dates in your logs use the following conversion. Notice that your dates have the **DD/MM/YY:hh:mm:ss** format.

$$\text{AccessTime} = \text{DD} * 86400 + \text{MM} * 2628288 + (\text{YY} - 1970) * 31536000 + \text{hh} * 3600 + \text{mm} * 60 + \text{ss}$$

Example Input:

```
5.78.198.52 - - [22/Jan/2019:03:56:32 +0330] "GET /games/hollow_knight HTTP/1.1" ...
5.78.198.52 - - [22/Jan/2019:03:56:32 +0330] "GET /games/hollow_knight HTTP/1.1" ...
2.177.12.140 - - [22/Jan/2019:03:57:32 +0330] "GET /image/shiba.jpg HTTP/1.1" ...
2.177.12.140 - - [23/Jan/2019:03:56:31 +0330] "GET /games/hollow_knight HTTP/1.1" ...
2.177.12.140 - - [23/Jan/2019:03:56:32 +0330] "GET /games/hollow_knight HTTP/1.1" ...
```

Console Command:

```
python task2.py 22/Jan/2019:03:56:32 86400 /games/hollow_knight
```

Example Output:

```
5.78.198.52 2 1549807280 → AccessTime(22/Jan/2019:03:56:32)
2.177.12.140 1 1549893679 → AccessTime(23/Jan/2019:03:56:31)
```

Notice that last request of **2.177.12.140** is **excluded** from our list, as:

AccessTime(23/Jan/2019:03:56:32) == AccessTime(22/Jan/2019:03:56:32) + 86400. You should only consider access times that is less than **date + duration**.

You are **ALLOWED** to use existing sorting functions and any data structure to keep track of these accesses.

Your python script should have the name **task2.py** and should read its inputs from a file called **log_task2.txt** under the same directory and write it's results to a file called **output_task2.txt**.

SUBMISSIONS

Create a folder in the form <name>_<surname>_<student_id>_HW3. Using Turkish characters and any capitalization of letters in the folder name is fine. **i.e.,** *john_nash_181101014_HW3*

Place all of your solutions (**task1.py** and **task2.py**) in this folder. You are not required to place any log or output files in this folder, as they will not be graded.

Compress this folder as **.zip** (preferred) or **.rar**, make sure that your archive also has the same name with the folder. **i.e.,** *john_nash_181101014_HW3.zip*

Attach this archive to an e-mail with title "**BIL334 HW3**" and send it to **canpolatog@gmail.com**.