TECHNION
Azrieli Continuing Education and
External Studies Division

Python Project

Tasks

For the following tasks you are provided with files that you will have to work with.

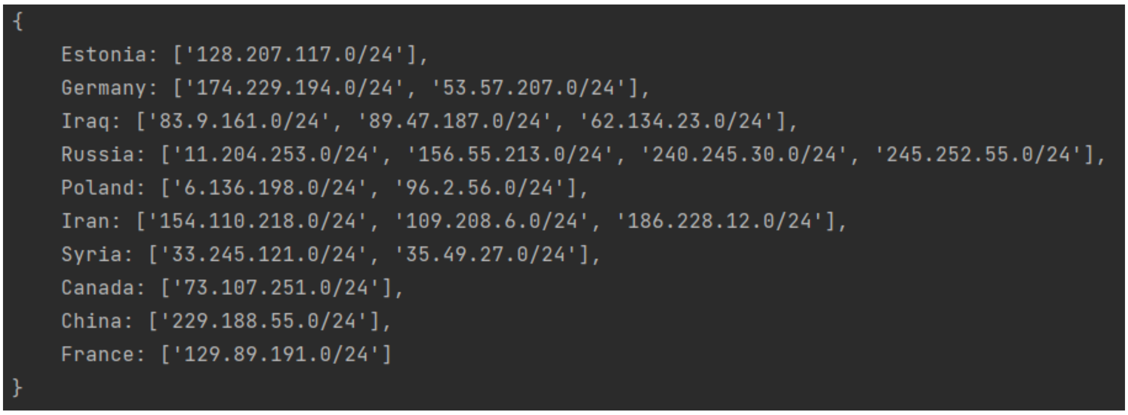
In the tasks you will be asked to write functions and create variables with given names. Make sure you name the functions and variables according to what you have been asked to. Functions with wrong names will lead to lower grade.

1. You are provided with a file named `**ranges.txt**`. The file contains a list of countries and a range of IP addresses in the following format:
   <country>:<address/CIDR>

   Write a function named `**read_ranges**` that does the following.

   a. The function takes a name of a file as an argument.

   b. The function reads the file and creates a dictionary in which the keys are the names of the countries, and the values are lists that contain the IP networks that belong to that country. The dictionary must look as follows:



```
{
    Estonia: ['128.207.117.0/24'],
    Germany: ['174.229.194.0/24', '53.57.207.0/24'],
    Iraq: ['83.9.161.0/24', '89.47.187.0/24', '62.134.23.0/24'],
    Russia: ['11.204.253.0/24', '156.55.213.0/24', '240.245.30.0/24', '245.252.55.0/24'],
    Poland: ['6.136.198.0/24', '96.2.56.0/24'],
    Iran: ['154.110.218.0/24', '109.208.6.0/24', '186.228.12.0/24'],
    Syria: ['33.245.121.0/24', '35.49.27.0/24'],
    Canada: ['73.107.251.0/24'],
    China: ['229.188.55.0/24'],
    France: ['129.89.191.0/24']
}
```

   c. The function should return the dictionary.

2. You are provided with a log file that contains evidence of a brute force attack against an SSH server. Before proceeding to the task, look at the file and see how it is constructed. For simplicity reasons, all the unnecessary information has been removed from the file.

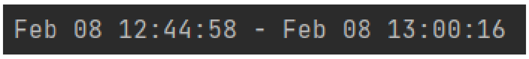
   The log file contains timestamps, IP addresses, usernames, and some additional information on failed SSH connections. In this task you will have to analyze the log file using a Python script and write functions that will answer all the questions/tasks.

   a. Write a function named `**read_log**` that takes one argument, the name of the log file. The function reads the file and returns a list that contains all the lines from that file without the trailing new lines.

b. Write a function named `**get_usernames**` that takes one argument. The argument that it takes is a list from the previous task. The function should return another list that contains all the usernames that appear in the log file. There should be no duplicates in the list that the function returns.

c. Write a function named `**get_addresses**` that gets all the IP addresses form the file. The function takes one argument, which is the list from task A. The function should return a list that contains all the IP addresses from the log file. Make sure that each IP address appears only ones in the returned list.

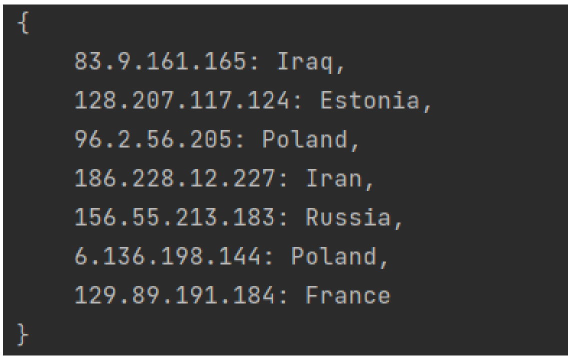
d. Write a function named `**start_end**`. The function takes one argument, the list from task A. The function should return a string that contains the first timestamp in the file and the last time stamp in the file. The following screenshot shows an example of the returned string. Notice how the timestamps are separated.



```
Feb 08 12:44:58 - Feb 08 13:00:16
```

e. (Bonus Task) Write a function named `**duration**` that takes the string from task D and calculates the duration of the attack. The function should return a string in the following format `**HH:MM:SS**` that represents the duration of the attack. For that task you are allowed to use libraries that can help you with the calculation of the duration.

f. Write a function named `**country_check**` that takes two arguments. The first argument is the dictionary from task 1, and the second argument is the list of IP addresses from task 2-C. The function should return a dictionary that contains the IP addresses from the list as keys and the country to which it belongs as arguments. The dictionary should look as the follows.



```
{
    83.9.161.165: Iraq,
    128.207.117.124: Estonia,
    96.2.56.205: Poland,
    186.228.12.227: Iran,
    156.55.213.183: Russia,
    6.136.198.144: Poland,
    129.89.191.184: France
}
```

Use the `**ipaddress**` library for this task.