Multiple Requests at a Time

In this challenge, read a text file and capture a timestamp from each line of text. Then **create a text file with a list of all timestamps that occur multiple times**, each on its own line. Naming convention and a data description are as follows:

Naming convention: You will be provided with an input file name called *filename*. **Your output filename should be req_*filename*** (replace *filename*).

Data description: Each line of the .txt file contains a single log record for *July 1995* with the following columns in order:

1. The *hostname* of the *host* making the request.
2. This column's values are missing and described by a hyphen (i.e., -).
3. This column's values are missing and described by a hyphen (i.e., -).
4. A timestamp enclosed in square brackets following the format *[DD/mmm/YYYY:HH:MM:SS -0400]*, where *DD* is the day of the month, *mmm* is the name of the month, *YYYY* is the year, *HH:MM:SS* is the time in *24*-hour format, and *-0400* is the time zone.
5. The *request*, enclosed in quotes (e.g., "*GET /images/NASA-logosmall.gif HTTP/1.0*").
6. The *HTTP response code*.
7. The total number of *bytes* sent in the response.

For example, given the following log record:

```
burger.letters.com - - [01/Jul/1995:00:00:12 -0400] "GET /shuttle/countdown/video/livevideo.gif HTTP/1.0" 200 0
```

We can label each column in the record like so:

Hostname	-	-	Timestamp	Request	HTTP Response Code	Bytes
burger.letters.com	-	-	[01/Jul/1995:00:00:12 -0400]	"GET /shuttle/countdown/video/livevideo.gif HTTP/1.0"	200	0

Given a string, *filename*, that denotes the name of a real text file, **create an output file named *req_filename*** to store timestamp records. Each line of the output file must contain a timestamp in the format *DD/mmm/YYYY:HH:MM:SS* for any timestamp that appears in more than one request in *filename*. The line order in the output file does not matter.

Constraints

- The log file contains no more than 2×10^5 records.

▶ **Input Format for Custom Testing**

▼ **Sample Case 0**

Sample Input

```
hosts_access_log_00.txt
```

Sample Output

Given *filename* = "*hosts_access_log_00.txt*", process the records in *hosts_access_log_00.txt* and create an output file named *req_hosts_access_log_00.txt* containing the following rows:

```
01/Jul/1995:00:00:12
01/Jul/1995:00:00:14
01/Jul/1995:00:00:15
```

Explanation 0

The log file *hosts_access_log_00.txt* contains the following log records:

```
unicomp6.unicomp.net - - [01/Jul/1995:00:00:06 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985
burger.letters.com - - [01/Jul/1995:00:00:11 -0400] "GET /shuttle/countdown/liftoff.html HTTP/1.0" 304 0
burger.letters.com - - [01/Jul/1995:00:00:12 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 304 0
burger.letters.com - - [01/Jul/1995:00:00:12 -0400] "GET /shuttle/countdown/video/livevideo.gif HTTP/1.0" 200 0
d104.aa.net - - [01/Jul/1995:00:00:13 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985
unicomp6.unicomp.net - - [01/Jul/1995:00:00:14 -0400] "GET /shuttle/countdown/count.gif HTTP/1.0" 200 40310
unicomp6.unicomp.net - - [01/Jul/1995:00:00:14 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 200 786
unicomp6.unicomp.net - - [01/Jul/1995:00:00:14 -0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 200 1204
d104.aa.net - - [01/Jul/1995:00:00:15 -0400] "GET /shuttle/countdown/count.gif HTTP/1.0" 200 40310
d104.aa.net - - [01/Jul/1995:00:00:15 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 200 786
```

The data confirms the following:

1. The timestamp *01/Jul/1995:00:00:12* occurs *two* times.
2. The timestamp *01/Jul/1995:00:00:14* occurs *three* times.
3. The timestamp *01/Jul/1995:00:00:15* occurs *two* times.

Strip the brackets and time zones from the three timestamps occurring more than once and append them to the output file.

We recommend you take a quick tour of our editor before you proceed. The timer will pause up to 90 seconds for the tour. Start tour ×

View Code Diff | Java 8

```
import java.io.*;⋯

public class Solution {
    private static final Scanner scan = new Scanner(System.in);

    public static void main(String args[]) throws Exception {
        // read the string filename
        String filename;
        filename = scan.nextLine();

    }
}


```

Line: 5 Col: 1

Test against custom input

Run Code

Submit code & Continue

(You can submit any number of times)

Download sample test cases

The input/output files have Unix line endings. Do not use Notepad to edit them on windows.