Introduction

A web application writes logs into the two separate log files access.log and forensics.json. The log data cannot be handled correctly by the in-house SIEM system. Therefore, a Python 3 program has to be developed to merge and convert the data of the two log files into a new file output.json with serialized JSON objects.

In the resources section you will find a zip file containing the following files:

- template.py: A template as starting point for the program to be developed. The template file handles optional arguments to override the location of the log files, looking in the current directory by default. The template reads the files from their relative locations and print the output to the STDOUT. access.log: The web server access log contains data in a standard httpd access log format with the following fields:
  - Unique Request ID
  - Remote IP
  - Remote Log Name (always -)
  - HTTP username
  - Timestamp
  - Request
  - Status code
  - Response size
- forensics.json: The forensics log file is a structured log file from the web application providing additional header data of the requests. The file contains one JSON object per line in the following format:

```
{"requestId":"XgwCwX8AAAEAAHE2NZoAAAAF","request":"GET /cron/vmcontrol.html?job=getList HTTP/
```

Goal & Tasks

Develop a Python program which meets the following requirements:

- Read and parse both log files
- Match log file entries using the unique request ID
- Write a JSON object to STDOUT or into the file output.json for each identified request in the file access.log
- If you encounter an unspecified situation, write an error message to STDERR
- Convert timestamps into ISO 8601 format using the Python 3 datetime library with the format string %d/%m/%Y:%H:%M:%S %z to parse the date

- Convert headers into a dictionary of lists (e.g. {‘Host’: [‘www.hacking-lab.com’], ‘Connection’: [‘keep-alive’], …})
- header is specified multiple times, all values should be retained in order
- Each JSON object in the file output.json must consist the following structure and partially converted data.

```
{
  "requestId": "XgwCwX8AAAEAAHE2NZoAAAAF",
  "remoteAddress": "212.254.246.102",
  "timestamp": "2020-01-01T03:24:01+01:00",
  "method": "GET",
  "url": "/cron/vmcontrol.html?job=getList",
  "version": "HTTP/1.1",
  "responseCode": 200,
  "responseSize": 64,
  "requestHeaders": {
    "Accept-Encoding": ["identity"],
    "Host": ["www.hacking-lab.com"],
    "Connection": ["close"],
    "User-Agent": ["Python-urllib/2.7"]
  }
}
```

The JSON objects in the file output.json must be separated with a newline character

Hint: Consider reducing the size of the log files during development. The following command will run it on only the first 30 lines of the access.log file: ./template.py -a <(head -n30 access.log)

Submission

ZIP file with the following contents:

- Your Python program (regardless of whether it is finished or not)
- The generated file output.json (if available)
- If external modules are used, list them in a file requirements.txt or Pipfile or include a README file with instructions