Javascript Programming Challenge

For applicants to Appointedd's software engineering roles

Preface

This challenge is designed to allow you to showcase your Javascript programming skills. There are no requirements apart from the fact that you must use Javascript. You are welcome to use any libraries you see fit and any online or offline materials to help you program a solution.

In this challenge you are given a text file containing a number of workers, each with an ID and a random number of ISO 8601 date/time intervals that represents an interval of time where that worker is free. Each worker is separated by a newline. Please ensure your solution reads the data from a file.

Don't worry if you can't create a perfect solution. If you write down any shortcomings of your code or if you describe how you would approach the problem we'll take that into consideration!

There is no requirement as to how you output your solution. Common methods are a script executable by node.js or an html page that displays the solution when opened works.

Example

```
1@[2019-12-31T23:45:00.000-03:00/2020-01-01T10:30:00.000+06:00,202
0-01-01T07:15:00.000+07:00/2019-12-31T16:00:00.000-10:00]
```

This is an example of an entry for one worker. The `@` (address) symbol is used as a divider between the worker's ID and the data representing the intervals of time for that worker. The characters on the left of the divider will always be numeric and represent the ID of the worker. For this worker their ID is `1`.

The characters on the right are the intervals of time that the worker is free. It is represented as an array that starts with the `[` character and ends with the `]` character. Each element in the array (if there is more than one) is separated by a comma character. In this example the worker has two intervals of time where they are free.

Each interval of time has a start and an end. Following the ISO 8601 the start date/time of the interval and the end date/time interval are separated by the `/` character. These intervals are given in a randomised order.

To summarise this worker:

- Has an ID of 1
- Has 2 intervals of time where they are free:
    - The first interval starts at `2019-12-31T23:45:00.000-03:00` and ends at `2020-01-01T10:30:00.000+06:00`.
    - The second interval starts at `2020-01-01T07:15:00.000+07:00` and ends at `2019-12-31T16:00:00.000-10:00`.

Test Case

```
1@[2019-12-31T23:45:00.000-03:00/2020-01-01T10:30:00.000+06:00,202
0-01-01T07:15:00.000+07:00/2019-12-31T16:00:00.000-10:00]
0@[2020-01-01T16:15:00.000+12:00/2019-12-31T18:30:00.000-10:00,202
0-01-01T02:15:00.000+02:00/2020-01-01T13:30:00.000+12:00,2020-01-0
1T09:00:00.000+07:00/2020-01-01T00:30:00.000-03:00]
2@[2020-01-01T11:00:00.000+09:00/2020-01-01T00:30:00.000-03:00,201
9-12-31T19:00:00.000-09:00/2019-12-31T22:45:00.000-06:00,2020-01-0
1T08:45:00.000+08:00/2019-12-31T14:15:00.000-11:00]
```

Above, is a simple test case for you to test your solvers against. For each question we'll provide the expected correct output for that question when run with the above test case.

Question 1

What is the starting date and time (in UTC) of the earliest interval where any of the workers are free?

Test Case Expected Output

```
2020-01-01T00:15:00.000Z
```

Question 2

What is the ending date and time (in UTC) of the latest interval where any of the workers are free?

Test Case Expected Output

```
2020-01-01T04:45:00.000Z
```

Question 3

What are the intervals of date and times (in UTC) where there are at least 2 workers free?

For example if you had 2 workers with one interval of time free each:

- Worker 1 is free between `2020-01-01T12:00:00.000Z/2020-01-01T15:30:00.000Z`
- Worker 2 is free between `2020-01-01T14:30:00.000Z/2020-01-01T16:30:00.000Z`

Then the interval of time that at least 2 workers are free is `2020-01-01T14:30:00.000Z/2020-01-01T15:30:00.000Z`.

Test Case Expected Output

```
2020-01-01T00:15:00.000Z/2020-01-01T01:30:00.000Z
2020-01-01T02:00:00.000Z/2020-01-01T03:30:00.000Z
2020-01-01T04:00:00.000Z/2020-01-01T04:30:00.000Z
```