Mightier Engineering Coding Exercise

Before You Begin

Make your best effort to package and structure your program to allow for understandability and future maintenance—that is, write this code with the same quality as a feature you'd be shipping to mightier.com. We don't expect you to spend more than 2 hours on this. If you run out of time, you can document in the readme what you would have done with more time.

The Challenge

We want to build a tool that determines an Emotion Score for each game played in Mightier. Our Emotion Score indicates a player's ability to calm themselves down after gameplay causes their heart rate to rise.

Calculating Emotion Score

For each game, we collect heart rates to understand the stress levels experienced during play. When a game finishes the game sends our back end data on what happened, including a stream of heart rate measurements as an array of integers. Before each game we receive a value called *threshold* is a baseline heart rate (for the purposes of this exercise, assume it is calculated before game play)**.** When a player's heart rate goes over the threshold, they are in the "red zone", and when their heart rate is at or below the threshold they are in the "blue zone".

To calculate an emotion score after a game, we look at the changes in heart rate throughout the course of that game. We have two scoring algorithms we would like to support.

Scoring Algorithm #1

When a player goes from blue→red→blue, they earn 10 points. Going red→blue (e.g. player starts game in red zone) earns 5 points. The total points are multiplied by that game's score multiplier to reward a player for the extra challenge of a difficult game.

Scoring Algorithm #2

When a player goes from red→blue, they earn 20 points. Ending the game in red will deduct 10 points.

Example JSON submitted at end of game play session:


```
{"game_date":"2019-01-15","game_id":"HIBACHO_HERO","threshold_hr":66,"heartRates":[68,66,64,66,62,63,64,65,68,67,65,65,67]}
```


Example Game Properties CSV:

```
game_id,display_name,score_multiplier
BRICK_BREAKER,Brick Breaker,3
MINI_METRO,Mini Metro,1
HIBACHI_HERO,Hibachi Hero,4
TIKI_TAKA_SOCCER,Tiki Taka Soccer,3
```

In the example above, a player has played Hibachi Hero with a threshold heart rate value of 66. The game started with the player in the *red zone* (heartRates[0]), cools to *blue zone* (heartRates[1]), rises to the *red zone* (heartRates[8]), and ends in the *blue zone* (heartRates[10]). This constitutes as one instance of *red→blue (5 points)* and one instance of *blue→red→blue (10 points)*, a total of 15 points. With the Hibachi Hero score multiplier, we have an emotion score of 15*4=**60** using scoring algorithm #1.

Your Challenge

Write a command-line program that takes in two arguments:

1. A json file containing game HR data (sample provided)
2. A CSV with game id/display name/score multiplier rows (sample provided)

Your program should print out the date of the game played, the display name of the game, and the emotion score for both algorithms.

Your program should work on the unmodified sample json and csv files linked above, or on other json/csv files in the same format.

When run against the example files linked above, your program should produce the following output:

```
2019-01-15
Hibachi Hero
Score1: 60
Score2: 30
```

You can write your program in any language you like as long as we can run it on the command line (not in an IDE), but we encourage you to use a language that you're very comfortable with and that will let you make changes fluidly, as we often build on your code in later interview stages. Once you've completed your exercise, please send us a zip of your source code along with a README.txt explaining:

- Any interesting trade-offs or edge cases you ran into.
- Any steps necessary to run your program on a command line, including compilation steps (if required) and the runtime versions you used (e.g., python version 3.7.2) - please assume we are not familiar with the language development environment when writing your steps.