Dear candidate, we are so excited that you are interviewing with Fooda! I hope you will enjoy this challenge. We request that you return the coding challenge back to us in one week's time or sooner. Please focus on writing clear, well-tested code. We will assume that this is the quality of code you write in production. Please write your code using Ruby.

# Coding Challenge

## Product requirements

Fooda is going to start rewarding our users for purchases. Every time you buy with Fooda you will get "points". Points can be redeemed for discounts on future purchases. Reward points are calculated on the following schedule:

|  |  |
| --- | --- |
| Local Time of Transaction | Reward Points |
| 12pm - 1pm | 1 point per $3 spent |
| 11am - 12pm and 1pm - 2pm | 1 point per $2 spent |
| 10am - 11am and 2pm - 3pm | 1 point per $1 spent |
| Any other time | 0.25 points per $1 |

Any orders that result in rewards points less than 3 or greater than 20 do not receive points. All rewards points are round up to the next integer.

## Technical requirements

You’re going to help us generate a simple report on rewards points for our customers.

The code will process a JSON input. You can imagine that the JSON is returned from a request to a service, but you can either read it in from a file or declare it in the code.

The JSON is composed of a collection of events.

* One type of event is “new customer”, which will register a new Customer in the app.
* The second type of event is “new order”, which will notify you of an order attributed to a customer.

Generate a report containing each customer with total rewards and average rewards per order. Report output should order users by total rewards most to least.

Example input:

{

"events":[

{

"action":"new\_customer",

"name":"Jessica",

"timestamp":"2020-07-01T00:00:00-05:00"

},

{

"action":"new\_customer",

"name":"Will",

"timestamp":"2020-07-01T01:00:00-05:00"

},

{

"action":"new\_customer",

"name":"Elizabeth",

"timestamp":"2020-07-01T12:00:00-05:00"

},

{

"action":"new\_order",

"customer":"Jessica",

"amount":12.50,

"timestamp":"2020-07-01T12:15:57-05:00"

},

{

"action":"new\_order",

"customer":"Jessica",

"amount":16.50,

"timestamp":"2020-07-01T10:01:00-05:00"

},

{

"action":"new\_order",

"customer":"Will",

"amount":8.90,

"timestamp":"2020-07-01T12:20:00-05:00"

},

{

"action":"new\_order",

"customer":"Will",

"amount":1.50,

"timestamp":"2020-07-01T12:21:00-05:00"

}

]

}

Expected output:

Jessica: 22 points with 11 points per order.

Will: 3 points with 3 points per order.

Elizabeth: No orders.

## How to submit?

Please use git and begin with an “Initial commit”. When you’re finished, please email us a git bundle (GIT\_DIR=your-code-dir/.git git bundle create your-code.gitbundle --all).