Customer Reward System

Your task is to design a customer reward system that matches customer purchases with a set of rules that are condition for a purchase to be eligible for a reward.

Rules are dynamically added to the system - in the beginning there are three types of rules

- Purchase amount higher than x
- Second purchase in last thirty days
- Purchase on 4th of may

These rules can be added to the system in any order. When added to the system and more than one rule matches, the last matching rule will count and be applied to determine the reward.

List of customer purchases

```
CUSTOMER_PURCHASES = {

{ customer_id: 65, purchase_amount_cents: 1800, created_at:
Time.utc(2009, 1, 2, 6, 1) }

{ customer_id: 31337, purchase_amount_cents: 6522, created_at:
Time.utc(2009, 5, 4, 6, 12) }

{ customer_id: 4465, purchase_amount_cents: 987, created_at:
Time.utc(2010, 8, 17, 11, 9) }

{ customer_id: 234234, purchase_amount_cents: 200, created_at:
Time.utc(2010, 11, 1, 16, 12) }

{ customer_id: 12445, purchase_amount_cents: 1664, created_at:
Time.utc(2010, 11, 18, 13, 19) }

{ customer_id: 234234, purchase_amount_cents: 1200, created_at:
Time.utc(2010, 12, 2, 16, 12) }

{ customer_id: 12445, purchase_amount_cents: 1800, created_at:
Time.utc(2010, 12, 3, 11, 17) }

{ customer_id: 65, purchase_amount_cents: 900, created_at: Time.utc(2011,
4, 28, 13, 16) }

{ customer_id: 65, purchase_amount_cents: 1600, created_at:
Time.utc(2011, 5, 4, 11, 1) }

}
```

Requirements

Dynamically addable rules on how customers get rewarded

- Add rule: Purchase amount higher than 1500 cents, "next purchase free"-reward
- Add rule: Second purchase in the past thirty days, "twenty percent off next order"-reward
- Add rule: Purchase on 4th May, "Star Wars themed item added to delivery"-reward

List of results

- List of results: customer actions and the determined rewards (if any)
- Rules are evaluated in the order that they were added to the system, with the newest rule that matches winning
- It is possible that no rule matches, give a sensible output for that case
- In your design, consider that new rules will get added on a regular basis
- Make sure your code is tested based on the given requirements (i.e. using RSpec)
- Consider a linter of your choice to keep a consistent style in your code
- Use git and appropriate commit-messages to document your changes