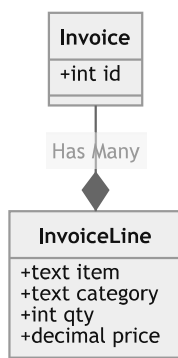# Promotion engine

Description

Our company wants to implement a *promotion engine* which automatically applies promotions to a specific invoice made by a customer.

For example:

```
Invoice #1

|-------------------------------------------------------------------|
| ITEM                        | QTY  | CATEGORY | PRICE | TOTAL |
| Banana                      | 3    |  FRUIT   | 2.50  |  7.50 |
| Promotion 2 buy 1 free (FRUIT) | 1 |  PROMO   | -2.50 | -2.50 |
|-------------------------------------------------------------------|
| TOTAL                       | 5.00                              |
```

Our data is structured as below:

The Promotion Engine

Once a customer is invoiced, his invoice passes through the promotion engine, and promotion lines are added accordingly to the list of available promotions.

Example (pseudocode):

```
/*
before:
invoice = {
   id: 1,
   lines: [
      {
         item: 'Banana',
         category: 'Fruit',
         qty: 3,
         price: 2.5
      }
   ]
}
*/

function onCheckout(invoice) {
   let promotionEngine = new PromotionEngine();
   promotionEngine.applyPromotions(invoice);
   return invoice;
}

/*
after:
invoice = {
   id: 1,
   lines: [
      {
```

```
      item: 'Banana',
      category: 'Fruit',
      qty: 3,
      price: 2.5
    },
    {
      item: 'Buy 2 get 1 free'
      category: 'Promotion'
      qty: 1,
      price: -2.5
    }
  ]
}
*/
```

Business requirement:

- Only one promotion can be active for any given instance of a product
- Promotion can target a specific product (e.g. Banana, milk etc...), or a specific product category (e.g. Fruit, Diairies etc...)
- Promotion can vary in conditions (e.g. for each X products buy) and ways of application (e.g. fixed versus percentage)
- Feeding the engine with a list of promotions, the engine will always select the promotion which is at the advantage of the customer.

Your mission

Write a prototype of the promotion engine, using Ruby, Javascript or Python. Your code must be shipped as a single file, with basic examples running.

- The maximum time allowed for this test is 2 hours.

- Getting 100% of the requirement is **very difficult**. Please do as best as you can. Do not hesitate to send back your code even if it doesn't work, or misses some edge cases.

You will ensure that:

- The engine performs accordingly to the requirements above.
- The engine can be extended; new promotion rules can be added easily.
- The code is clear and follow industry standards and design principles.