

Project Fruitpal

Take-Home Assignment

is expanding into the algorithmic trading of tropical fruits!

We need your help building a tool called **fruitpal**, which allows a trader to understand the full cost of buying fruit from various countries of origin.

Inputs

A trader using this tool will specify the contemplated purchase, including:

1. The commodity (e.g. mangos)
2. The price per ton (in dollars)
3. The trade volume (in tons)

Example

A trader who wants to know the full cost of buying 405 tons of mangos at \$53 a ton would run:

```
> fruitpal mango 53 405
```

Outputs

The tool must return a list of all available countries of origin, and, for each:

- The country code
- The total cost of the purchase
- The breakdown of the costs between variable and fixed components, as shown in the example

The list must be sorted by total cost from highest to lowest.

Example

In response to the Example Input, the trader might see:

```
< BR 22060.10 | (54.42*405)+20  
< MX 21999.20 | (54.24*405)+32
```

This means the 405 tons of mangos would cost \$22,060.10 if bought in Brazil and \$21,999.20 if

bought in Mexico. The Brazilian mangoes cost \$54.42 per-ton and \$20 per-trade.

Calculations

The cost is computed from a combination of the trader's inputs with market overhead data (fees and taxes) that we are getting from a 3rd-party API.

- The cost per pound is calculated as the price from the trader + the "variable overhead" from our data vendor.
- The cost per trade is strictly the "fixed overhead" from our data vendor.

3rd-Party Data

We have a deal with a 3rd-party that provides us the latest fixed and variable overhead data for various fruits across a variety of countries.

The data is provided to us on a nightly basis as a flat file transmitted via FTP (however you can assume that your program will have access to the file locally).

The file contains json with the following format:

```
[
  {
    "COUNTRY": "MX",
    "COMMODITY": "mango",
    "FIXED_OVERHEAD": "32.00",
    "VARIABLE_OVERHEAD": "1.24"
  },
  {
    "COUNTRY": "BR",
    "COMMODITY": "mango",
    "FIXED_OVERHEAD": "20.00",
    "VARIABLE_OVERHEAD": "1.42"
  }
]
```

Considerations

- Assume that this is a proof of concept that will eventually be plugged into a larger software system. You should treat this like a piece of core business logic rather than a throwaway script.
- Implement your solution as production quality code that will be maintained by a team of engineers in the long run. You or your teammates should be able to come back to this

code in six months and be able to easily extend it.

- The stability of your system is important, so consider writing some automated tests or using some other best practices to ensure it is stable and correct, such as creating additional test input.
- Your onsite interview will dive more into this problem and extend it. If you have a laptop, please bring it to the interview and have it setup to work on your solution. If you don't have a laptop, let us know and we will provide one.