# Clojure programming exercise

This exercise will test your skills on data processing challenges we face at DCI .

You need to examine a JSON file and extract the data from this file to
answer the following questions.

The JSON file is a deeply nested map containing ecommerce data about 100 lipstick products, their pricing, discounts, and likewise. This is a JSON representation of a page on an ecommerce site.

Each product (a lipstick) can have a number of SKUs (color variations).

Italicized text in the question are keys in the map represented by the JSON file

The exercise

## Question 1:

### Input data:

The value against the '*listItems*' (at line 508) key is an array, where each item in the array is a map. For each item in *listItems*

- get *productUrl*, *price*, *originalPrice*
- count the number of *skus*.

e.g. the *skus* field for the first product on line 549, has 6 skus listed in the array.

### Output:

- spit a CSV file (named first.csv) with the columns: productUrl, price, originalPrice, number of skus for each item in *listItems*

Example output for the first product.

--

productUrl, price, originalPrice, numberOfSKUs.
‘//www.lazada.co.th/products/qianxiu-q127-the-new-moisturizing-and-waterproof-moisturizer-is-not-easy-to-wear-cokkicosmetic-i224295407-s946252295.html?search=1’,”39.00”, ‘\u0e3f109.00’, 6
--

## Question 2

Calculate the average price of all products that satisfies these conditions:
number of *skus* (for that product) is > 2  and *brandName* = "OEM".

### Output data:

Spit the result into a file “second.csv”.

## Question 3

Group the products by *brandName*, return a map where the key is *brandName* and value is the count of products (not SKUs) for that brand.

### Output data:

Convert the map into a JSON file and write the result to third.json

## Question 4

Find all the *image* keys at any depth in the json file,and get the value against the *image* key.

example: "image": "https://th-test-11.slatic.net/p/195cf3282d11fd25e181b55c298ebc14.jpg"

Then extract the filename and strip the path (e.g. 195cf3282d11fd25e181b55c298ebc14.jpg)

### Output data:

Spit out all the **unique (**remove duplicates**)** image file names into a csv file named fourth.csv.

# Question 5 (bonus question if you have the time):

Assume that you have won a gift voucher worth 250.00. You would like to purchase

- The maximum number of products possible.
- The products selected must be of different *brandNames*
- There should be little or no money left over.

Assume that the *price* field for computing the price.

## Output data:

Spit out csv file named fifth.csv that has a list of *itemId, brandName, price* for the selected products.

To submit your exercise,

- create a new repo that has the source code for the above exercise.
- running ‘lein run’ should read the json file (in the same directory) and output all the result files into the same directory.
- Send us
    - the repo containing the source code
    - the output files.