FASHION CLOUD

Test Assignment

WE ARE THRILLED YOU WANT TO JOIN OUR TEAM

```
>_ three-steps-2-met-fcteam
>_ 1-code --python
>_ 2-think --flowchart
>_ 3-architect --aws
```

Task description

For this coding challenge, you will process a catalog with shoes as we receive it from a well-known shoe supplier. You have received two different files that you need to accomplish this: `pricat.csv` and `mappings.csv`

pricat.csv
This file is what we call a price catalog. It represents the catalog of a shoe supplier containing articles. It is a flat format file that uses the semicolon `';'` as column separator. That means that every possible configuration of a type of shoe is represented by a single line.

For example, we have a shoe with article number `15189-02-001` that is available in sizes `36`, `37` and `38` and the colors black and white. However, size `37` is only available in `white`. This would be represented by 5 records in the file:

```
- 15189-02-001;36;white
- 15189-02-001;37;white
- 15189-02-001;38;white
- 15189-02-001;36;black
- 15189-02-001;38;black
```

The `pricat.csv` has more of these variations. The first line represents the header of each column. In principle, this could be different for every pricat received.

The goal of the challenge is to transform this price catalog into another format. The configuration of the mapping is defined in the other file.

mappings.csv

This file is also a column separated file using the `';'` as separator with the first line as header. Each line contains a mapping from a source field to a destination field. The simple example of the first two non-header lines is as following:

```
- Values of winter in the season column need to be
mapped to the destination type  season with value
Winter.
- Values of summer in the season column need to be
mapped to the destination type  season with value
Summer.
```

It is also possible that multiple source values need to be combined into a new destination type. For example, the value `size_group_code` need to be combined with the value of the `size_code`. Only with those two columns combined you know to which destination value it needs to be mapped. The values `EU` and `36` together become `European size 36` with the type `size`.

Not all columns are mapped in the `mappings.csv`. Columns that are not empty should be copied to the same type in the result. An example of this is the `brand` column.

>_ next

Grouping

As mentioned before, the `pricat.csv` is a flat file. We would like to have the result in a more structured way. That structure looks as following:

```
Catalog —> Article —> Variation
```

This means we create one Catalog that contains multiple Articles. Each Article is defined by a unique article number, and can contain multiple Variations. After grouping the Variations into Articles by article number, move attributes that are common for all children up to their parent, both from Variation to Article and from Article to Catalog level. Here are some hints:

- the **brand** is the same in each row and therefore belongs to the Catalog
- the **ean** is different for every Variation

This would result in something as following:

```
Catalog
     brand: Via Vai
     ??
          Article
               article_number: 15189-02
               ??
                    Variation
                         ??
                         ??
                    Variation
                         ??
                         ??
          Article
               article_number: 4701013-00
               ??
                    Variation
                         ??
                         ??
```

Output

The result should be the entire structured `Catalog`, including all `Articles` and `Variations`, outputted in `JSON` format. You're free to determine the exact format, but it should be something that could be returned by an API. The focus of this challenge is on the algorithm itself. You don't need to build a web framework or a proper UI. Instead, it's better to spend that time fulfilling all requirements, writing clean code and writing good test cases.

Bonus points

- Create a configurable option to combine multiple fields into a new field. For example:
  combine `` `price_buy_net` `` and `` `currency` `` into `` `price_buy_net_currency` `` which would create `` `58.5 EUR` `` out of `` `58.5` `` and `` `EUR` `` for the first article.
- Write unit and/or integration tests for your code

Constraints

When using Python, do not use the library pandas.

PLEASE SHARE YOUR GITHUB REPO WITH CODING CHALLENGE WITH/OR EMAIL IT IN A ZIP ARCHIVE TO

`reviewer@fashion.cloud`

Task description

Let's assume we have two collections:

```
Product {
   ean: string;
   brandId: string;
   articleNumber: string;
   description: string;
   size: string;
   color: string;
}
PurchaseOrder {
   id: string;
   retailerId: string;
   ean: string;
   quantity: number;
   purchaseDate: Date
}
```


An `ean` (European Article Number) is an identifier for the product.

Please explain how to loop through these collections and get a list of 3 most popular colors per year outlining the logic while being programming language agnostic.

```
[{
    year: number;
    popularColors: [{
        color: string,
        numberOfPurchaseQuantities : number
    }]
}]
```

Turn your thoughts into a flowchart diagram. Imagine that we would give your flowchart to a junior developer who only knows the syntax of some programming language. He or she should be able to follow your flowchart and successfully implement the solution.
You can use https://app.diagrams.net.

AS AN OUTPUT YOU NEED TO PROVIDE A FLOWCHART DIAGRAM EXPLAINING THE LOGICAL FLOW. EMAIL IT TO `reviewer@fashion.cloud`

Task description

Let's assume we have an API in which we will get a large json file of product information and import it in our system.
Since there's going to be so much data to process in the API synchronously, first we get the information and save it somewhere temporarily, and then we will process the data.

This is the list of processes that we have to do on product data to get it ready to use in our system.

- *We need to validate if the data we're given is valid or not (for example if all necessary attributes are provided)*
- *We need to scan the file with antivirus software.*
- *We need to standardize the metadata we get (for example we get one color with different names like Galaxy Blue, Navy Blue, Royal Blue, etc but we want to convert them all to one format like Dark blue)*
- *We need to create different resolutions of the media files we get to use in different platforms*
- *We need to save the processed data in the correct collection.*

If any error occurs in this process we need to notify the person who has uploaded the data of the situation (let’s assume we have their contact details).

Also we need to be able to do high-performing search queries against processed data. For example we need to be able to search for items in a specific color, size, material, etc.

You need to create a diagram representing the architecture (ideally using AWS landscape) to cater for the requirements above. You can use https://app.diagrams.net.

Please submit your diagram as an image. Feel free to support your diagram with explanation to cover your decision making process.

AS AN OUTPUT YOU NEED TO PROVIDE A DIAGRAM OUTLINING YOUR ARCHITECTURAL SOLUTION. EMAIL IT TO `reviewer@fashion.cloud`

WE'LL USE OUR ~~ARTIFICIAL~~ MANUAL INTELLIGENCE TO EVALUATE YOUR RESULTS AND WILL GET BACK TO YOU SHORTLY!

THANK YOU