Fashion Cloud Coding Challenge

_____

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.

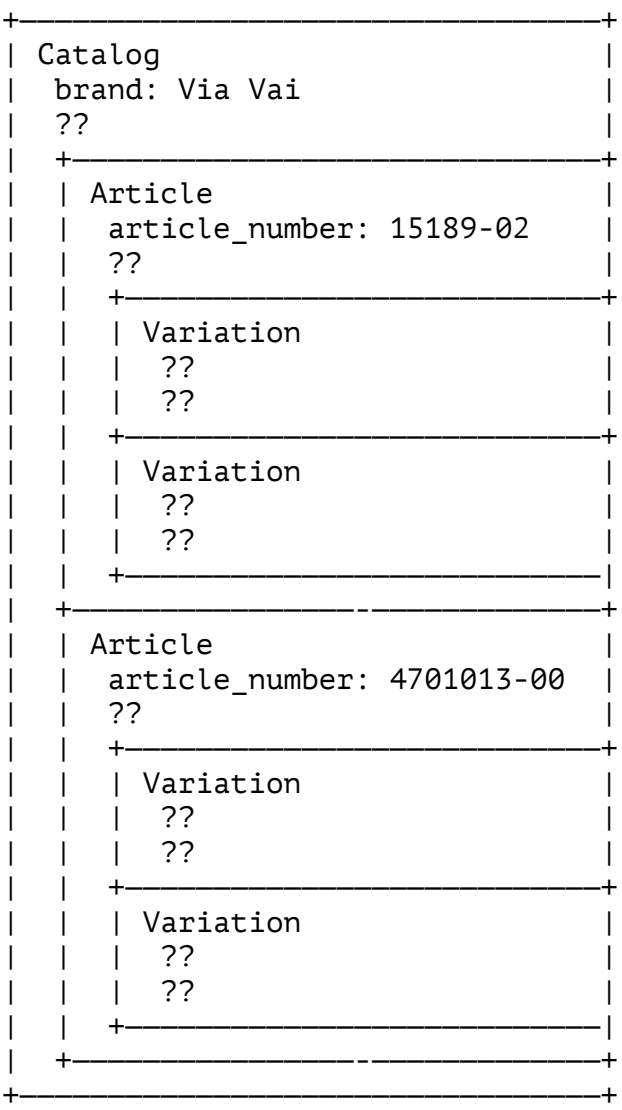
Grouping

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:



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.

Bonus points

i) 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.
ii) Write unit and/or integration tests for your code

Constraints

When using Python, do not use the library pandas.

Submission

Please send your working code (plus any other resources you might have such as documentation, explanations, notes or tests) to viktoria.hrechyshnikova@fashion.cloud and erwin.rossen@fashion.cloud in a zip archive (no tar.gz or rar please). Include instructions how to run the code.