One iota Technical Task

Import Product Data

We sometimes receive product data in CSV format, and have to reconstruct it into an object structure. For this task, we'd like you to convert some product data from CSV to JSON.

We need one JSON object for each product, which might have several sizes. The CSV data has one row for each size, identified by its SKU (Stock Keeping Unit) code. To make products with sizes, you'll need to group together sizes based on their 'PLU' (product listing unit) code. For example, given CSV data like this:

```
SKU, PLU, name,            size, sizeSort
100, ABC, Vision Windrush, 11,   SHOE_UK
101, ABC, Vision Windrush, 9,    SHOE_UK
109, ABC, Vision Windrush, 12,   SHOE_UK
200, DEF, Vision Jacket,   S,    CLOTHING_SORT
201, DEF, Vision Jacket,   L,    CLOTHING_SORT
209, DEF, Vision Jacket,   M,    CLOTHING_SORT
```

We need JSON like this:

```
[
    {
      "PLU": "ABC",
      "name": "Vision Windrush",
      "sizes": [
        {
          "SKU": "101",
          "size": "9"
        },
        {
          "SKU": "100",
          "size": "11"
        },
        {
          "SKU": "109",
          "size": "12"
        }
      ]
    },
    {
      "PLU": "DEF",
      "name": "Vision Jacket",
```

```
        "sizes": [
          {
            "SKU": "200",
            "size": "S"
          },
          {
            "SKU": "209",
            "size": "M"
          },
          {
            "SKU": "201",
            "size": "L"
          }
        ]
      }
  ]
```

Note that the options are sorted by their size. The only trouble is that different products use different values for size, and they don't always sort alphabetically. For example `"large" < "small"`. Your solution will need to look at the `sizeSort` value from the CSV and sort the options accordingly. At present, we know of three orderings for sizes, but your solution should be open for more to be added. The possible orderings are:

`SHOE_UK`

- 1 (Child)
- 1.5 (Child)
- 2 (Child)
- 2.5 (Child)
- ... and so on, up to... 12 (Child)
- 1
- 1.5
- 2
- 2.5
- ... etc.

`SHOE_EU`

Numeric values from 20 to 50.

`CLOTHING_SHORT`

- XS
- S

- M
- L
- XL
- XXL
- XXXL
- XXXXL

The data to be converted should be supplied with this document. Please submit your code, any tests, and instructions for running, along with the JSON data produced by your code.

If you have any questions feel free to get in touch with the Platform Lead via email at recruitment@oneiota.co.uk

Good luck!