

Technical Test Data Quality Engineer Foxintelligence

You will find below three logic tests to be solved with python. For each of these tests, there is no single solution but rather we seek to evaluate your methodology for solving logic problems via code

Question 1.

Given a dataframe with three columns:

- client_id
- ranking
- value

Write a function to fill the NaN values in the **value** column with the previous non-NaN value from the same **client_id** ranked in ascending order.

Input:

```
clients = {"client_id" : [1001, 1001, 1001, 1002, 1002, 1002, 1003, 1003, 1003], "ranking" : [1, 2, 3, 1, 2, 3, 1, 2, 3], "value" : [1000, None, 1200, 1500, 1250, None, 1200, 1100, None]}
```

Question 2.

Given a list of timestamps in sequential order, return a list of lists grouped by week (7 days) using the first timestamp as the starting point.

Example:

```
input=["2019-01-01','2019-01-02','2019-01-08','2019-02-01','2019-02-02','2019-02-05']
```

```
output=>[
```

```
['2019-01-01','2019-01-02'],['2019-01-08'],['2019-02-01','2019-02-02'],['2019-02-05']
```

```
]
```

Question 3.

You're given two dataframes: transactions and products. The transactions dataframe contains transaction ids, product ids, and the total amount of each product sold.

The products dataframe contains product ids and prices. Write a function to return a dataframe containing every transaction with a total value of over \$100. Include the total value of the transaction as a new column in the dataframe.

Input:

```
import pandas as pd
transactions = {"transaction_id": [1, 2, 3, 4, 5], "product_id": [101, 102, 103, 104, 105],
"amount": [3, 5, 8, 3, 2]}
products = {"product_id": [101, 102, 103, 104, 105], "price": [20.00, 21.00, 15.00, 16.00,
52.00]}
df_transactions = pd.DataFrame(transactions)
df_products = pd.DataFrame(products)
```