# 2922. Market Analysis III

## Description

Table: `Users`

```
+----------------+---------+
| Column Name    | Type    |
+----------------+---------+
| seller_id      | int     |
| join_date      | date    |
| favorite_brand | varchar |
+----------------+---------+
seller_id is column of unique values for this table.
This table contains seller id, join date, and favorite brand of sellers.
```

Table: `Items`

```
+---------------+---------+
| Column Name   | Type    |
+---------------+---------+
| item_id       | int     |
| item_brand    | varchar |
+---------------+---------+
item_id is the column of unique values for this table.
This table contains item id and item brand.
```

Table: `Orders`

```
+---------------+---------+
| Column Name   | Type    |
+---------------+---------+
| order_id      | int     |
| order_date    | date    |
| item_id       | int     |
| seller_id     | int     |
+---------------+---------+
order_id is the column of unique values for this table.
item_id is a foreign key to the Items table.
seller_id is a foreign key to the Users table.
This table contains order id, order date, item id and seller id.
```

Write a solution to find the **top seller** who has sold the highest number of **unique** items with a **different** brand than their favorite brand. If there are multiple sellers with the same highest count, return all of them.

Return *the result table ordered by* `seller_id` *in* ***ascending*** *order.*

The result format is in the following example.

### Example 1:

```
 Input:
Users table:
+-----------+------------+----------------+
| seller_id | join_date  | favorite_brand |
+-----------+------------+----------------+
| 1         | 2019-01-01 | Lenovo         |
| 2         | 2019-02-09 | Samsung        |
| 3         | 2019-01-19 | LG             |
+-----------+------------+----------------+
Orders table:
+----------+------------+---------+-----------+
| order_id | order_date | item_id | seller_id |
+----------+------------+---------+-----------+
| 1        | 2019-08-01 | 4       | 2         |
| 2        | 2019-08-02 | 2       | 3         |
| 3        | 2019-08-03 | 3       | 3         |
| 4        | 2019-08-04 | 1       | 2         |
| 5        | 2019-08-04 | 4       | 2         |
+----------+------------+---------+-----------+
Items table:
+---------+------------+
| item_id | item_brand |
+---------+------------+
| 1       | Samsung    |
| 2       | Lenovo     |
| 3       | LG         |
| 4       | HP         |
+---------+------------+
 Output:
+-----------+-----------+
| seller_id | num_items |
+-----------+-----------+
| 2         | 1         |
| 3         | 1         |
+-----------+-----------+
 Explanation:
- The user with seller_id 2 has sold three items, but only two of them are not marked as a favorite. We will include a unique count of 1 because both of these items are identical.
- The user with seller_id 3 has sold two items, but only one of them is not marked as a favorite. We will include just that non-favorite item in our count.
Since seller_ids 2 and 3 have the same count of one item each, they both will be displayed in the output.
```