Table: `Sales`

```
+------------+-------+
| Column Name | Type  |
+------------+-------+
| sale_id     | int   |
| product_id  | int   |
| user_id     | int   |
| quantity    | int   |
+------------+-------+
```
sale_id is the primary key of this table.
product_id is a foreign key to Product table.
Each row of this table shows the ID of the product and the quantity purchased by a user.

Table: `Product`

```
+------------+------+
| Column Name | Type |
+------------+------+
| product_id  | int  |
| price       | int  |
+------------+------+
```
product_id is the primary key of this table.
Each row of this table indicates the price of each product.

Write an SQL query that reports the spending of each user.

Return the resulting table ordered by `spending` in **descending order**. In case of a tie, order them by `user_id` in ascending order.

The query result format is in the following example.

Example 1:**

```
Input:
Sales table:
+---------+------------+---------+----------+
| sale_id | product_id | user_id | quantity |
+---------+------------+---------+----------+
| 1       | 1          | 101     | 10       |
| 2       | 2          | 101     | 1        |
| 3       | 3          | 102     | 3        |
| 4       | 3          | 102     | 2        |
```

```
| 5        | 2          | 103     | 3         |
+---------+------------+---------+----------+
```
Product table:
```
+------------+-------+
| product_id | price |
+------------+-------+
| 1          | 10    |
| 2          | 25    |
| 3          | 15    |
+------------+-------+
```
Output:
```
+---------+----------+
| user_id | spending |
+---------+----------+
| 101     | 125      |
| 102     | 75       |
| 103     | 75       |
+---------+----------+
```
Explanation:
User 101 spent 10 * 10 + 1 * 25 = 125.
User 102 spent 3 * 15 + 2 * 15 = 75.
User 103 spent 3 * 25 = 75.
Users 102 and 103 spent the same amount and we break the tie by their ID while user 101 is