Table: Visits

+---------------+---------+  
| Column Name | Type |  
+---------------+---------+  
| user\_id | int |  
| visit\_date | date |  
+---------------+---------+  
(user\_id, visit\_date) is the primary key (combination of columns with unique values) for this table.  
Each row of this table indicates that user\_id has visited the bank in visit\_date.

Table: Transactions

+------------------+---------+  
| Column Name | Type |  
+------------------+---------+  
| user\_id | int |  
| transaction\_date | date |  
| amount | int |  
+------------------+---------+  
This table may contain duplicates rows.  
Each row of this table indicates that user\_id has done a transaction of amount in transaction\_date.  
It is guaranteed that the user has visited the bank in the transaction\_date.(i.e The Visits table contains (user\_id, transaction\_date) in one row)

A bank wants to draw a chart of the number of transactions bank visitors did in one visit to the bank and the corresponding number of visitors who have done this number of transaction in one visit.

Write a solution to find how many users visited the bank and didn't do any transactions, how many visited the bank and did one transaction, and so on.

The result table will contain two columns:

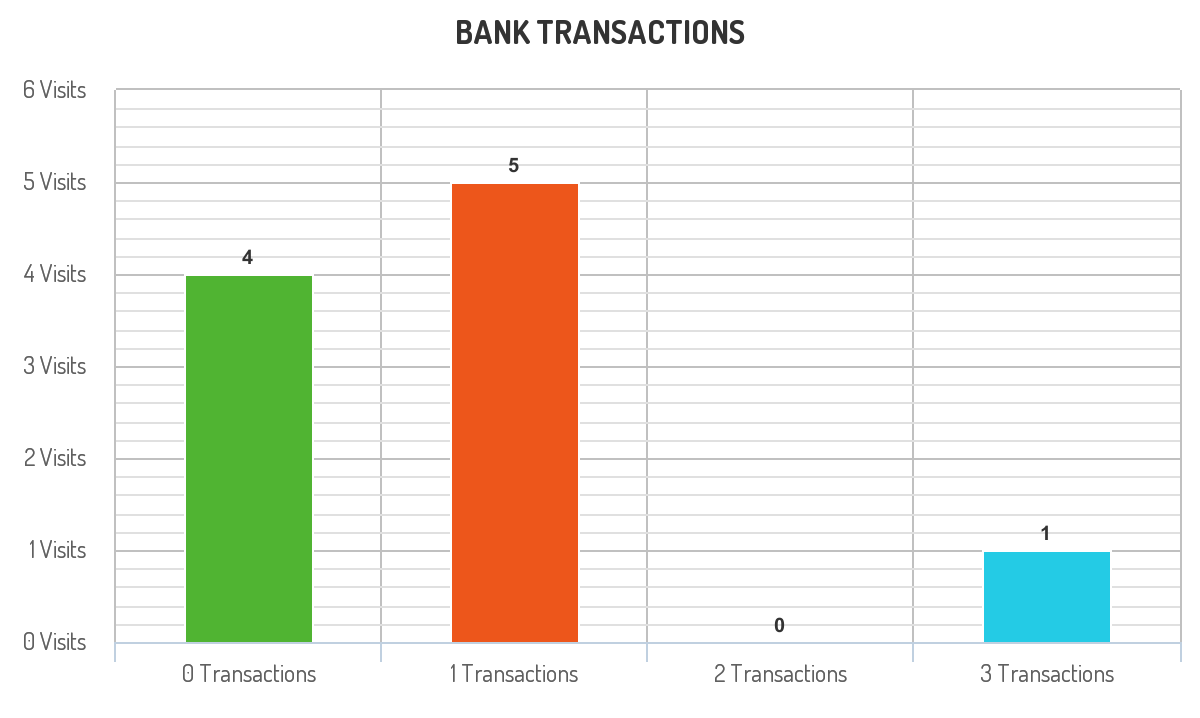
* transactions\_count which is the number of transactions done in one visit.
* visits\_count which is the corresponding number of users who did transactions\_count in one visit to the bank.

transactions\_count should take all values from 0 to max(transactions\_count) done by one or more users.

Return the result table ordered by transactions\_count.

The result format is in the following example.

**Example 1:**



Input:   
Visits table:  
+---------+------------+  
| user\_id | visit\_date |  
+---------+------------+  
| 1 | 2020-01-01 |  
| 2 | 2020-01-02 |  
| 12 | 2020-01-01 |  
| 19 | 2020-01-03 |  
| 1 | 2020-01-02 |  
| 2 | 2020-01-03 |  
| 1 | 2020-01-04 |  
| 7 | 2020-01-11 |  
| 9 | 2020-01-25 |  
| 8 | 2020-01-28 |  
+---------+------------+  
Transactions table:  
+---------+------------------+--------+  
| user\_id | transaction\_date | amount |  
+---------+------------------+--------+  
| 1 | 2020-01-02 | 120 |  
| 2 | 2020-01-03 | 22 |  
| 7 | 2020-01-11 | 232 |  
| 1 | 2020-01-04 | 7 |  
| 9 | 2020-01-25 | 33 |  
| 9 | 2020-01-25 | 66 |  
| 8 | 2020-01-28 | 1 |  
| 9 | 2020-01-25 | 99 |  
+---------+------------------+--------+  
Output:   
+--------------------+--------------+  
| transactions\_count | visits\_count |  
+--------------------+--------------+  
| 0 | 4 |  
| 1 | 5 |  
| 2 | 0 |  
| 3 | 1 |  
+--------------------+--------------+  
Explanation: The chart drawn for this example is shown above.  
\* For transactions\_count = 0, The visits (1, "2020-01-01"), (2, "2020-01-02"), (12, "2020-01-01") and (19, "2020-01-03") did no transactions so visits\_count = 4.  
\* For transactions\_count = 1, The visits (2, "2020-01-03"), (7, "2020-01-11"), (8, "2020-01-28"), (1, "2020-01-02") and (1, "2020-01-04") did one transaction so visits\_count = 5.  
\* For transactions\_count = 2, No customers visited the bank and did two transactions so visits\_count = 0.  
\* For transactions\_count = 3, The visit (9, "2020-01-25") did three transactions so visits\_count = 1.  
\* For transactions\_count >= 4, No customers visited the bank and did more than three transactions so we will stop at transactions\_count = 3