Table: `Friends`

```
+---------------+---------+
| Column Name   | Type    |
+---------------+---------+
| id            | int     |
| name          | varchar |
| activity      | varchar |
+---------------+---------+
id is the id of the friend and primary key for this table.
name is the name of the friend.
activity is the name of the activity which the friend takes part in.
```

Table: `Activities`

```
+---------------+---------+
| Column Name   | Type    |
+---------------+---------+
| id            | int     |
| name          | varchar |
+---------------+---------+
id is the primary key for this table.
name is the name of the activity.
```

Write an SQL query to find the names of all the activities with neither the maximum nor the minimum number of participants.

Each activity in the `Activities` table is performed by any person in the table Friends.

Return the result table in **any order**.

The query result format is in the following example.

Example 1:**

```
Input:
Friends table:
+------+--------------+---------------+
| id   | name         | activity      |
+------+--------------+---------------+
| 1    | Jonathan D.  | Eating        |
| 2    | Jade W.      | Singing       |
```

```
| 3    | Victor J.   | Singing       |
| 4    | Elvis Q.    | Eating        |
| 5    | Daniel A.   | Eating        |
| 6    | Bob B.      | Horse Riding  |
+------+-------------+---------------+
```

Activities table:

```
+------+-------------+
| id   | name        |
+------+-------------+
| 1    | Eating      |
| 2    | Singing     |
| 3    | Horse Riding |
+------+-------------+
```

Output:

```
+--------------+
| activity     |
+--------------+
| Singing      |
+--------------+
```

Explanation:

Eating activity is performed by 3 friends, maximum number of participants, (Jonathan D. , El
Horse Riding activity is performed by 1 friend, minimum number of participants, (Bob B.)
Singing is performed by 2 friends (Victor J. and Jade W.)