Table Variables:

+---------------+---------+  
| Column Name | Type |  
+---------------+---------+  
| name | varchar |  
| value | int |  
+---------------+---------+  
In SQL, name is the primary key for this table.  
This table contains the stored variables and their values.

Table Expressions:

+---------------+---------+  
| Column Name | Type |  
+---------------+---------+  
| left\_operand | varchar |  
| operator | enum |  
| right\_operand | varchar |  
+---------------+---------+  
In SQL, (left\_operand, operator, right\_operand) is the primary key for this table.  
This table contains a boolean expression that should be evaluated.  
operator is an enum that takes one of the values ('<', '>', '=')  
The values of left\_operand and right\_operand are guaranteed to be in the Variables table.

Evaluate the boolean expressions in Expressions table.

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

The result format is in the following example.

**Example 1:**

Input:   
Variables table:  
+------+-------+  
| name | value |  
+------+-------+  
| x | 66 |  
| y | 77 |  
+------+-------+  
Expressions table:  
+--------------+----------+---------------+  
| left\_operand | operator | right\_operand |  
+--------------+----------+---------------+  
| x | > | y |  
| x | < | y |  
| x | = | y |  
| y | > | x |  
| y | < | x |  
| x | = | x |  
+--------------+----------+---------------+  
Output:   
+--------------+----------+---------------+-------+  
| left\_operand | operator | right\_operand | value |  
+--------------+----------+---------------+-------+  
| x | > | y | false |  
| x | < | y | true |  
| x | = | y | false |  
| y | > | x | true |  
| y | < | x | false |  
| x | = | x | true |  
+--------------+----------+---------------+-------+  
Explanation:   
As shown, you need to find the value of each boolean expression in the table using the variables table.