You are given an array of variable pairs equations and an array of real numbers values, where equations[i] = [Ai, Bi] and values[i] represent the equation Ai / Bi = values[i]. Each Ai or Bi is a string that represents a single variable.

You are also given some queries, where queries[j] = [Cj, Dj] represents the jth query where you must find the answer for Cj / Dj = ?.

Return *the answers to all queries*. If a single answer cannot be determined, return -1.0.

**Note:** The input is always valid. You may assume that evaluating the queries will not result in division by zero and that there is no contradiction.

**Note:**The variables that do not occur in the list of equations are undefined, so the answer cannot be determined for them.

**Example 1:**

Input: equations = [["a","b"],["b","c"]], values = [2.0,3.0], queries = [["a","c"],["b","a"],["a","e"],["a","a"],["x","x"]]  
Output: [6.00000,0.50000,-1.00000,1.00000,-1.00000]  
Explanation:   
Given: a / b = 2.0, b / c = 3.0  
queries are: a / c = ?, b / a = ?, a / e = ?, a / a = ?, x / x = ?   
return: [6.0, 0.5, -1.0, 1.0, -1.0 ]  
note: x is undefined => -1.0

**Example 2:**

Input: equations = [["a","b"],["b","c"],["bc","cd"]], values = [1.5,2.5,5.0], queries = [["a","c"],["c","b"],["bc","cd"],["cd","bc"]]  
Output: [3.75000,0.40000,5.00000,0.20000]

**Example 3:**

Input: equations = [["a","b"]], values = [0.5], queries = [["a","b"],["b","a"],["a","c"],["x","y"]]  
Output: [0.50000,2.00000,-1.00000,-1.00000]

**Constraints:**

* 1 <= equations.length <= 20
* equations[i].length == 2
* 1 <= Ai.length, Bi.length <= 5
* values.length == equations.length
* 0.0 < values[i] <= 20.0
* 1 <= queries.length <= 20
* queries[i].length == 2
* 1 <= Cj.length, Dj.length <= 5
* Ai, Bi, Cj, Dj consist of lower case English letters and digits.