# 2307. Check for Contradictions in Equations

## Description

You are given a 2D array of strings `equations` and an array of real numbers `values`, where `equations[i] = [A_i, B_i]` and `values[i]` means that `A_i / B_i = values[i]`.

Determine if there exists a contradiction in the equations. Return `true` *if there is a contradiction, or* `false` *otherwise*.

**Note**:

- When checking if two numbers are equal, check that their **absolute difference** is less than `10^-5`.
- The testcases are generated such that there are no cases targeting precision, i.e. using `double` is enough to solve the problem.

**Example 1:**

```
Input: equations = [["a","b"],["b","c"],["a","c"]], values = [3,0.5,1.5]
Output: false
Explanation:
The given equations are: a / b = 3, b / c = 0.5, a / c = 1.5
There are no contradictions in the equations. One possible assignment to satisfy all equations is:
a = 3, b = 1 and c = 2.
```

**Example 2:**

```
Input: equations = [["le","et"],["le","code"],["code","et"]], values = [2,5,0.5]
Output: true
Explanation:
The given equations are: le / et = 2, le / code = 5, code / et = 0.5
Based on the first two equations, we get code / et = 0.4.
Since the third equation is code / et = 0.5, we get a contradiction.
```

**Constraints:**

- `1 <= equations.length <= 100`
- `equations[i].length == 2`
- `1 <= A_i.length, B_i.length <= 5`
- `A_i`, `B_i` consist of lowercase English letters.
- `equations.length == values.length`
- `0.0 < values[i] <= 10.0`
- `values[i]` has a maximum of 2 decimal places.