1058. Minimize Rounding Error to Meet Target

Description

Given an array of `prices` `[p 1 ,p 2 ...,p n ]` and a `target`, round each price `p i` to `Round i (p i )` so that the rounded array `[Round 1 (p 1 ),Round 2 (p 2 )...,Round n (p n )]` sums to the given `target`. Each operation `Round i (p i )` could be either `Floor(p i )` or `Ceil(p i )`.

Return the string `"-1"` if the rounded array is impossible to sum to `target`. Otherwise, return the smallest rounding error, which is defined as `Σ |Round i (p i ) - (p i )|` for `i` from `1` to `n`, as a string with three places after the decimal.

Example 1:

```
Input: prices = ["0.700","2.800","4.900"], target = 8
Output: "1.000"
Explanation:
Use Floor, Ceil and Ceil operations to get (0.7 - 0) + (3 - 2.8) + (5 - 4.9) = 0.7 + 0.2 + 0.1 = 1.0 .
```

Example 2:

```
Input: prices = ["1.500","2.500","3.500"], target = 10
Output: "-1"
Explanation: It is impossible to meet the target.
```

Example 3:

```
Input: prices = ["1.500","2.500","3.500"], target = 9
Output: "1.500"
```

Constraints:

- `1 <= prices.length <= 500`
- Each string `prices[i]` represents a real number in the range `[0.0, 1000.0]` and has exactly 3 decimal places.
- `0 <= target <= 10 6`