2337. Move Pieces to Obtain a String

Description

You are given two strings `start` and `target`, both of length `n`. Each string consists **only** of the characters `'L'`, `'R'`, and `'_'` where:

- The characters `'L'` and `'R'` represent pieces, where a piece `'L'` can move to the **left** only if there is a **blank** space directly to its left, and a piece `'R'` can move to the **right** only if there is a **blank** space directly to its right.
- The character `'_'` represents a blank space that can be occupied by **any** of the `'L'` or `'R'` pieces.

Return `true` *if it is possible to obtain the string* `target` *by moving the pieces of the string* `start` ***any*** *number of times*. Otherwise, return `false`.

Example 1:

```
Input: start = "_L__R__R_", target = "L______RR"
Output: true
Explanation: We can obtain the string target from start by doing the following moves:
- Move the first piece one step to the left, start becomes equal to "L___R__R_".
- Move the last piece one step to the right, start becomes equal to "L___R___R".
- Move the second piece three steps to the right, start becomes equal to "L______RR".
Since it is possible to get the string target from start, we return true.
```

Example 2:

```
Input: start = "R_L_", target = "__LR"
Output: false
Explanation: The 'R' piece in the string start can move one step to the right to obtain "_RL_".
After that, no pieces can move anymore, so it is impossible to obtain the string target from start.
```

Example 3:

```
Input: start = "_R", target = "R_"
Output: false
Explanation: The piece in the string start can move only to the right, so it is impossible to obtain the string target from start.
```

Constraints:

- `n == start.length == target.length`
- `1 <= n <= 10^5`
- `start` and `target` consist of the characters `'L'`, `'R'`, and `'_'`.