To some string S, we will perform some replacement operations that replace groups of letters with new ones (not necessarily the same size).

Each replacement operation has 3 parameters: a starting index i, a source word x and a target word y.  The rule is that if x starts at position i in the **original** **string** **S**, then we will replace that occurrence of x with y.  If not, we do nothing.

For example, if we have S = "abcd" and we have some replacement operation i = 2, x = "cd", y = "ffff", then because "cd" starts at position 2 in the original string S, we will replace it with "ffff".

Using another example on S = "abcd", if we have both the replacement operation i = 0, x = "ab", y = "eee", as well as another replacement operation i = 2, x = "ec", y = "ffff", this second operation does nothing because in the original string S[2] = 'c', which doesn't match x[0] = 'e'.

All these operations occur simultaneously.  It's guaranteed that there won't be any overlap in replacement: for example, S = "abc", indexes = [0, 1], sources = ["ab","bc"] is not a valid test case.

**Example 1:**

**Input:** S = "abcd", indexes = [0,2], sources = ["a","cd"], targets = ["eee","ffff"]

**Output:** "eeebffff"

**Explanation:** "a" starts at index 0 in S, so it's replaced by "eee".

"cd" starts at index 2 in S, so it's replaced by "ffff".

**Example 2:**

**Input:** S = "abcd", indexes = [0,2], sources = ["ab","ec"], targets = ["eee","ffff"]

**Output:** "eeecd"

**Explanation:** "ab" starts at index 0 in S, so it's replaced by "eee".

"ec" doesn't starts at index 2 in the **original** S, so we do nothing.

Notes:

1. 0 <= indexes.length = sources.length = targets.length <= 100
2. 0 < indexes[i] < S.length <= 1000
3. All characters in given inputs are lowercase letters.