# Problem 1023: Camelcase Matching

## Problem Information

**Difficulty:** Medium
**Acceptance Rate:** 64.63%
**Paid Only:** No
**Tags:** Array, Two Pointers, String, Trie, String Matching

## Problem Description

Given an array of strings `queries` and a string `pattern`, return a boolean array `answer` where `answer[i]` is `true` if `queries[i]` matches `pattern`, and `false` otherwise.

A query word `queries[i]` matches `pattern` if you can insert lowercase English letters into the pattern so that it equals the query. You may insert a character at any position in pattern or you may choose not to insert any characters **at all**.

**Example 1:**

**Input:** queries = ["FooBar","FooBarTest","FootBall","FrameBuffer","ForceFeedBack"], pattern = "FB" **Output:** [true,false,true,true,false] **Explanation:** "FooBar" can be generated like this "F" + "oo" + "B" + "ar". "FootBall" can be generated like this "F" + "oot" + "B" + "all". "FrameBuffer" can be generated like this "F" + "rame" + "B" + "uffer".

**Example 2:**

**Input:** queries = ["FooBar","FooBarTest","FootBall","FrameBuffer","ForceFeedBack"], pattern = "FoBa" **Output:** [true,false,true,false,false] **Explanation:** "FooBar" can be generated like this "Fo" + "o" + "Ba" + "r". "FootBall" can be generated like this "Fo" + "ot" + "Ba" + "ll".

**Example 3:**

**Input:** queries = ["FooBar","FooBarTest","FootBall","FrameBuffer","ForceFeedBack"], pattern = "FoBaT" **Output:** [false,true,false,false,false] **Explanation:** "FooBarTest" can be generated like this "Fo" + "o" + "Ba" + "r" + "T" + "est".

**Constraints:**

* `1 <= pattern.length, queries.length <= 100` * `1 <= queries[i].length <= 100` * `queries[i]` and `pattern` consist of English letters.

## Code Snippets

**C++:**

```cpp
class Solution {
public:
vector<bool> camelMatch(vector<string>& queries, string pattern) {


}
};
```

**Java:**

```java
class Solution {
public List<Boolean> camelMatch(String[] queries, String pattern) {


}
}
```

**Python3:**

```python
class Solution:
def camelMatch(self, queries: List[str], pattern: str) -> List[bool]:
```