Problem of the Week : ***Step Words Finder***.
This problem was asked by Pivotal.

Problem Description:

Scenario:
In a word game, a **step word** is created by **adding exactly one letter** to a given word and then **anagramming** the result to form a valid dictionary word.

For example, from `"APPLE"`:

- Add `"A"` → `"APPLEA"`
- Anagram to form `"APPEAL"` (a valid dictionary word) → This is a **step word**.

You are given:

- A dictionary of valid English words (as a list of strings),
- An input word.

Your task is to write a function that returns **all valid step words** that can be created from the input word using the above logic.

Input Format:

- First line: A string `W` (the input word)
- Second line: An integer `N` (the number of words in the dictionary)
- Next `N` lines: Each line contains one valid word from the dictionary

Output Format:

- Print each valid **step word** (one per line) in **lexicographical order**

Constraints:

- `1 <= len(W) <= 15`
- `1 <= N <= 10^5`
- All dictionary words are lowercase, alphabetic, and contain no spaces

Example Input:

```
apple
5
appeal
apply
pepla
papple
apples
```

Example Output:

```
appeal
papple
```

Explanation:

- `"appeal"` is formed by adding `"a"` to `"apple"` → `"applea"` → anagram → `"appeal"`
- `"papple"` is formed by adding `"p"` → `"applep"` → anagram → `"papple"`
- `"apply"`, `"pepla"`, and `"apples"` are **not** valid because:
    - `"apply"` is missing `"e"`, not just one letter added
    - `"pepla"` is not formed by adding just one letter
    - `"apples"` has 6 letters, but the added letter doesn't result in a correct multiset match with `"apple"`

🚀 Approach Hint:

- Count the character frequencies of the input word
- For each word in the dictionary:
    - If its length is not exactly one more than the input word, skip it
    - Compare character frequency maps
    - The word is valid if **only one character is added**

⏱️ Expected Time Complexity:

- O(N * K), where K = max word length (up to 15)
- Optimize using `collections.Counter` or fixed-length frequency arrays

🛠️ Practice Links:

- 🔗 Leetcode – Group Anagrams (Similar logic)
- 🔗 GeeksforGeeks – Valid Anagram

📺 Video Resources:

- 🎥 YouTube – Check for Anagrams using Counter