2559. Count Vowel Strings in Ranges

You are given a **0-indexed** array of strings `words` and a 2D array of integers `queries`.

Each query `queries[i] = [li, ri]` asks us to find the number of strings present in the range `li` to `ri` (both **inclusive**) of `words` that start and end with a vowel.

Return *an array* `ans` *of size* `queries.length`, *where* `ans[i]` *is the answer to the* `i`th *query*.

Note that the vowel letters are `'a'`, `'e'`, `'i'`, `'o'`, and `'u'`.

Example 1:

```
Input: words = ["aba","bcb","ece","aa","e"], queries = [[0,2],[1,4],[1,1]]
Output: [2,3,0]
Explanation: The strings starting and ending with a vowel are "aba", "ece", "aa"
and "e".
The answer to the query [0,2] is 2 (strings "aba" and "ece").
to query [1,4] is 3 (strings "ece", "aa", "e").
to query [1,1] is 0.
We return [2,3,0].
```

Example 2:

```
Input: words = ["a","e","i"], queries = [[0,2],[0,1],[2,2]]
Output: [3,2,1]
Explanation: Every string satisfies the conditions, so we return [3,2,1].
```

Constraints:

- `1 <= words.length <=` 10^5
- `1 <= words[i].length <= 40`
- `words[i]` consists only of lowercase English letters.
- `sum(words[i].length) <= 3 *` 10^5
- `1 <= queries.length <=` 10^5
- `0 <= li <= ri < words.length`

2559. Count Vowel Strings in Ranges

```
/*
   Prefix sum
   Time complexity: O(n+q)
   Space complexity: O(n)
   n: # words
   q: #queries
*/
class Solution{
   public:
      std::vector<int> vowelStrings(std::vector<std::string>& words, std::vector<std::vector<int>>&
queries){
         auto is_vowel=[&](char c)->bool{
            return c=='a' || c=='e' || c=='i' || c=='o' || c=='u';
         };

         auto is_vowel_stack_over_flow=[&](char c)->bool{
            return (0x208222>>(c&0x1f))&1;
         };

         int n=words.size();

         std::vector<int> prefix_vowels(n+1,0);
         for(int i=1;i<=n;++i){
            int len=words[i-1].size();
            char c1=words[i-1][0];
            char c2=words[i-1][len-1];
            prefix_vowels[i]=prefix_vowels[i-1]+int(is_vowel(c1) && is_vowel(c2));
         }

         std::vector<int> ans;
         for(auto& q: queries){
            int l=q[0];
            int r=q[1];
            ans.push_back(prefix_vowels[r+1]-prefix_vowels[l]);
         }
         return ans;
      }
};
```