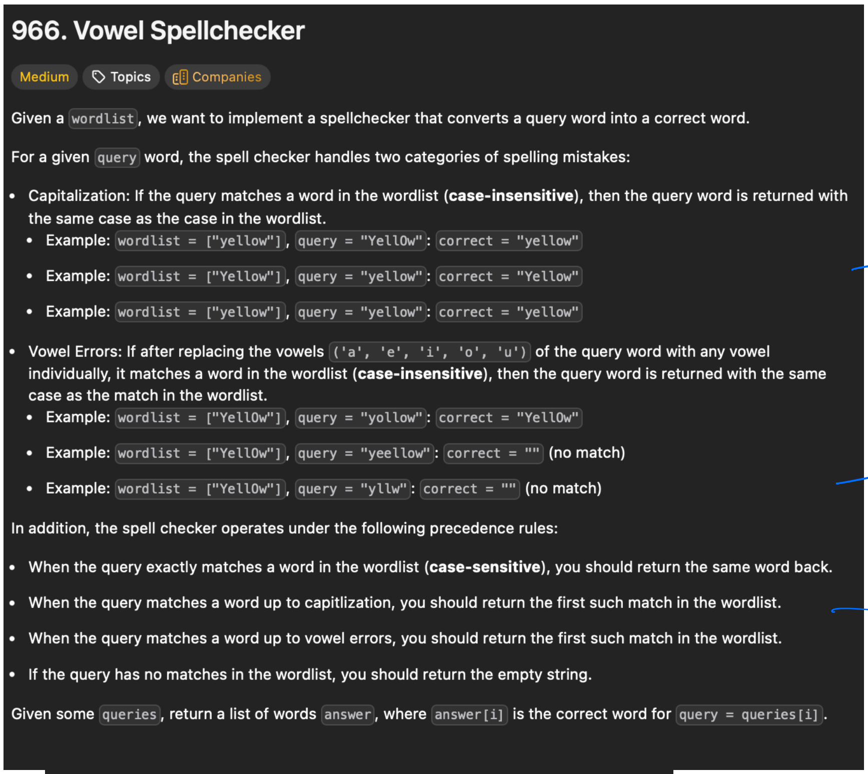


# 966. Vowel Spellchecker

Given a `wordlist`, we want to implement a spellchecker that converts a query word into a correct word.

For a given `query` word, the spell checker handles two categories of spelling mistakes:

- Capitalization: If the query matches a word in the wordlist (**case-insensitive**), then the query word is returned with the same case as the case in the wordlist.
  - Example: `wordlist = ["yellow"]`, `query = "YellOw"`: `correct = "yellow"`
  - Example: `wordlist = ["Yellow"]`, `query = "yellow"`: `correct = "Yellow"`
  - Example: `wordlist = ["yellow"]`, `query = "yellow"`: `correct = "yellow"`
- Vowel Errors: If after replacing the vowels `('a', 'e', 'i', 'o', 'u')` of the query word with any vowel individually, it matches a word in the wordlist (**case-insensitive**), then the query word is returned with the same case as the match in the wordlist.
  - Example: `wordlist = ["YellOw"]`, `query = "yollow"`: `correct = "YellOw"`
  - Example: `wordlist = ["YellOw"]`, `query = "yeellow"`: `correct = ""` (no match)
  - Example: `wordlist = ["YellOw"]`, `query = "yllw"`: `correct = ""` (no match)

In addition, the spell checker operates under the following precedence rules:

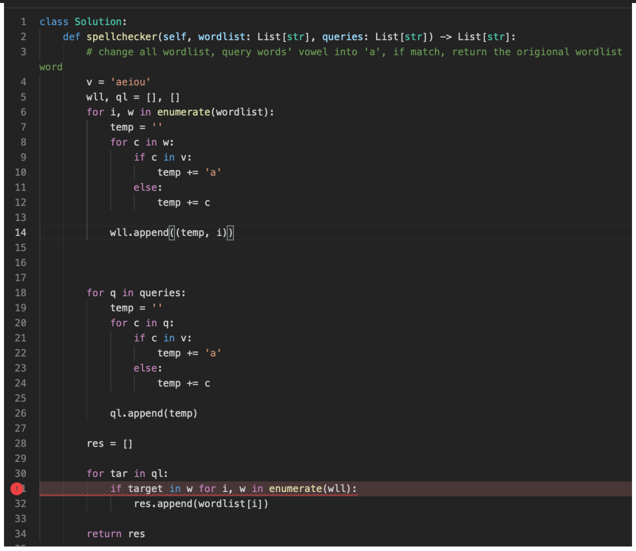
- When the query exactly matches a word in the wordlist (**case-sensitive**), you should return the same word back.
- When the query matches a word up to capitlization, you should return the first such match in the wordlist.
- When the query matches a word up to vowel errors, you should return the first such match in the wordlist.
- If the query has no matches in the wordlist, you should return the empty string.

Given some `queries`, return a list of words `answer`, where `answer[i]` is the correct word for `query = queries[i]`.

→ 大小写可以忽略

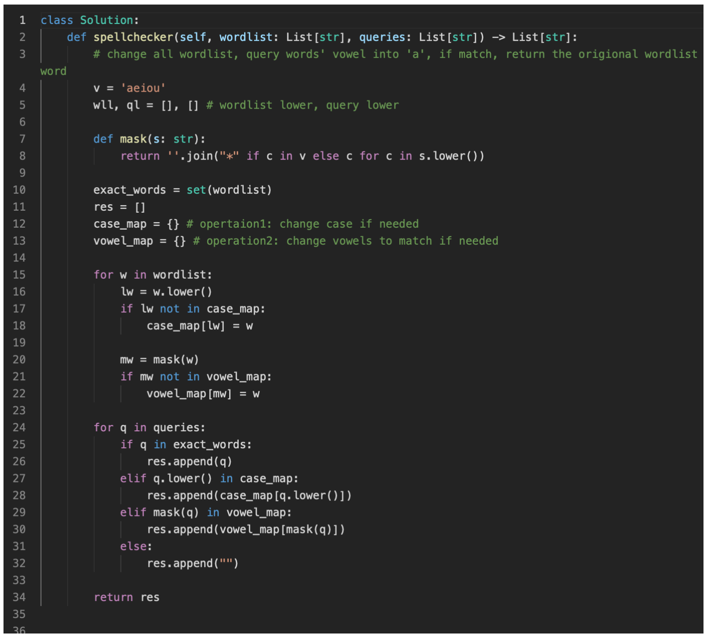
→ vowel 不同可以忽略.

→ 以上两个忽略时，如果没匹配上，return the exact word in wordlist.



```python
class Solution:
    def spellchecker(self, wordlist: List[str], queries: List[str]) -> List[str]:
        # change all wordlist, query words' vowel into 'a', if match, return the origional wordlist 
word
        v = 'aeiou'
        wll, ql = [], []
        for i, w in enumerate(wordlist):
            temp = ''
            for c in w:
                if c in v:
                    temp += 'a'
                else:
                    temp += c

            wll.append((temp, i))



        for q in queries:
            temp = ''
            for c in q:
                if c in v:
                    temp += 'a'
                else:
                    temp += c

            ql.append(temp)

        res = []

        for tar in ql:
            if target in w for i, w in enumerate(wll):
                res.append(wordlist[i])

        return res
```

version 1,
brute force.



```python
class Solution:
    def spellchecker(self, wordlist: List[str], queries: List[str]) -> List[str]:
        # change all wordlist, query words' vowel into 'a', if match, return the origional wordlist 
word
        v = 'aeiou'
        wll, ql = [], [] # wordlist lower, query lower

        def mask(s: str):
            return ''.join("*" if c in v else c for c in s.lower())

        exact_words = set(wordlist)
        res = []
        case_map = {} # opertaion1: change case if needed
        vowel_map = {} # operation2: change vowels to match if needed

        for w in wordlist:
            lw = w.lower()
            if lw not in case_map:
                case_map[lw] = w

            mw = mask(w)
            if mw not in vowel_map:
                vowel_map[mw] = w

        for q in queries:
            if q in exact_words:
                res.append(q)
            elif q.lower() in case_map:
                res.append(case_map[q.lower()])
            elif mask(q) in vowel_map:
                res.append(vowel_map[mask(q)])
            else:
                res.append("")

        return res
```

correct version.

1° mask word only needed

2° follow the matching order:
[unchange, case, vowel]