Problem # 824. Goat Latin

<https://leetcode.com/problems/goat-latin/>

1. Define the set of vowels

2. Split the string S into words and store it in a list called S\_list

3. Find the length of S\_list which is n

4. Iterate through the for loop for i in range(n). If the first letter of the word is a consonant, then slice the word so that it starts with the second letter and the first letter is added to the end of the word.

5. Every word whether vowel or consonant has "ma" added at the end and also "a" repeated index times starting with 1 (not 0).

6. Append the modified word to the result list.

7. Return the result list as a string using join().

Note: It is more efficient in terms of memory usage to add the words to a list and do the join, than to keep appending to a string as strings are immutable.

```

class Solution:

def toGoatLatin(self, S: str) -> str:

vowels = set('aeiouAEIOU') # define the set of vowels

S\_list = S.split()

n = len(S\_list)

result = []

for i in range(n):

word = S\_list[i]

if word[0] not in vowels: # word begins with a consonant

word = word[1:] + word[0]

word = word + "ma" + "a" \* (i + 1)

result.append(word)

return(" ".join(result))

```