


PROBLEM

Arrange Consonants and Vowels



Medium Accuracy: 49.98% Submissions: 28K+ Points: 4

Done with winning Geekbits? Now win GfG Bag, GfG T-shirt & much more just by writing your experiences. Start Writing, Start Winning. 

Given a singly linked list having n nodes containing english alphabets ('a'-'z'). Rearrange the linked list in such a way that all the vowels come before the consonants while maintaining the **order of their arrival**.

Example 1:

Input:

$n = 9$

linked list: a -> b -> c -> d -> e -> f -> g -> h -> i

Output:

a -> e -> i -> b -> c -> d -> f -> g -> h

Explanation:

After rearranging the input linked list according to the condition the resultant linked list will be as shown in output.

Example 2:

Input:

$n = 8$

linked list: a -> b -> a -> b -> d -> e -> e -> d

Output:

a -> a -> e -> e -> b -> b -> d -> d

Explanation:

After rearranging the input linked list according to the condition the resultant linked list will be as shown in output.

Your Task:

Your task is to complete the function **arrangeCV()**, which takes **head** of linked list and arranges the list in such a way that all the vowels come before the consonants while maintaining the **order of their arrival** and returns the head of the updated linked list.

Expected Time Complexity : $O(n)$

Expected Auxiliary Space : $O(1)$

Constraints:

$1 \leq n \leq 10^4$

'a' <= elements of linked list <= 'z'

CODE

#User function Template for python3

"""

Node Class

class Node:

def __init__(self, val):

self.data = val

self.next = None

"""

class Solution:

#Function to reverse a linked list.

def arrangeCV(self, head):

Code here

vowels = ["a","e","i","o","u"]

dummy = Node("p")

dummy.next = head

temp = dummy

curr = dummy

while temp and temp.next:

if temp.next.data not in vowels:

break

temp = temp.next

curr = curr.next

while curr and curr.next:

if curr.next.data in vowels:

front = temp.next

temp.next = curr.next

temp = temp.next

curr.next = temp.next

temp.next = front

else:

curr = curr.next

return dummy.next