Wandering fingers

def outside(word):  
 return word[0] + word[-1]  
  
  
def inside(word):  
 return word[1:-1]  
  
  
def issubword(word, words):  
 woord = list(word)  
 for i in range(len(woord)-1):  
 if word[i] == word[i+1]:  
 woord[i] = ''  
 woord = ''.join(woord)  
 woord = list(woord)  
 for i in words:  
 if i == woord[0]:  
 del woord[0]  
 if len(woord) == 0:  
 return True  
 return False  
  
  
def iswandering(word, words):  
 if issubword(word, words) == True and word[0] == words[0] and word[-1] == words[-1]:  
 return True  
 else:  
 return False

일단 제가 푼 건 여기까지구요 def 2개 남았어요

Def read\_dictionary(txt):

**>>>** dictionary = read\_dictionary('[dictionary.txt](https://medusa.ugent.be/en/activities/574186898/description/3KrsMJMFxddsngbw/media/data/dictionary.txt)')

**>>>** dictionary['qk']

{'uir', 'uar', 'uarterdec', 'uac', 'uarterbac', 'uic'}

이건 첫번째 def outside를 이용해서 해당 txt에서 앞자리가 q, 마지막자리가 k인 단어를 뽑아내기

**>>>** dictionary['qn']

{'uotidia', 'uee', 'uestio', 'uicke'}

이건 첫번째 단어가 q, 마지막 단어가 n인 단어를 뽑아내기

**>>>** dictionary['xx']

Traceback (most recent call last):

KeyError: 'xx'

이건 첫번째 단어나 마지막단어나 둘 다 없는 단어일 때 keyerror

Def wanderings(words, dictionary):

**>>>** wanderings('qwertyuihgfcvbnhjk', dictionary)

{'quick'}

**>>>** wanderings('qwertyuytresdftyuiokn', dictionary)

{'queen', 'question'}

**>>>** wanderings('ghijakjthoijerjidsdfnokg', dictionary)

{'gating', 'geeing', 'goring', 'going', 'gathering'}

**>>>** wanderings('xkzjunspebfgslddfksdrx', dictionary)

set()

일단 4번째 def iswandering같은 경우는 (1)첫번째 단어가 words의 첫번쨰 단어랑 똑같고, (2)마지막 단어가 words의 마지막 단어랑 똑같고, (3)issubword를 만족하는 단어일 때(3가지 조건 다 만족해야함)

#issubword는 해당 words에서 word가 순서대로 포함되어있으면 true

그래서 def wanderings(word, dictionary): 에서는 해당 dictionary.txt에서 단어들 중에 words에 맞는 iswandering을 만족시키는 단어를 뽑아내는 것