A)

1) Count the occurrences of each letter in the text.

freq = {}  
# A  
for i in teststr:  
 if i in freq:  
 freq[i] += 1  
 else:  
 freq[i] = 1

1. Create an empty dictionary.
2. Update values when it finds the same character

Example:

Count of all characters:

{'L': 1, 'e': 32, 't': 27, ' ': 59, 'm': 10, 'n': 20, 'o': 19, 'h': 16, 'a': 15, 'r': 17, 'i': 19, 'g': 4, 'f': 3, 'u': 3, 'd': 8, 's': 17, 'A': 2, 'p': 2, 'l': 6, 'v': 7, 'W': 2, 'c': 1, 'w': 5, '(': 1, ')': 1, 'O': 1, 'b': 3, 'x': 1, 'è': 1, 'k': 6, 'T': 1, 'I': 1, 'y': 1, "'": 2, '\n': 1}

2) Print the number of one-letter, two-letter, three-letter words and so on.

count1 = 0;  
count2 = 0;  
count3 = 0;  
count4 = 0;  
count5 = 0;  
count6 = 0;  
count7 = 0;  
count8 = 0;  
count9 = 0;  
count10 = 0;  
for j in data:  
 if len(j) == 1:  
 count1 = count1 + 1;  
 elif len(j) == 2:  
 count2 = count2 + 1;  
 elif len(j) == 3:  
 count3 = count3 + 1;  
 elif len(j) == 4:  
 count4 = count4 + 1;  
 elif len(j) == 5:  
 count5 = count5 + 1;  
 elif len(j) == 6:  
 count6 = count6 + 1;  
 elif len(j) == 7:  
 count7 = count7 + 1;  
 elif len(j) == 8:  
 count8 = count8 + 1;  
 elif len(j) == 9:  
 count9 = count9 + 1;  
 elif len(j) == 10:  
 count10 = count10 + 1;  
  
print("One letter word " + str(count1))  
print("Two letter word " + str(count2))  
print("Three letter word " + str(count3))  
print("Four letter word " + str(count4))  
print("Five letter word " + str(count5))  
print("Six letter word " + str(count6))  
print("Seven letter word " + str(count7))  
print("Eight letter word " + str(count8))  
print("Nine letter word " + str(count9))  
print("Ten letter word " + str(count10))

1. Let counts of each letter word.
2. Initialize them with zero
3. Check lengths of words and update according to it

Example:

One letter word 0

Two letter word 17

Three letter word 9

Four letter word 10

Five letter word 10

Six letter word 5

Seven letter word 3

Eight letter word 3

Nine letter word 1

Ten letter word 1

3) Print the number of occurrences of each different word in the text.

counts = dict()  
for word in data:  
 if word in counts:  
 counts[word] += 1  
 else:  
 counts[word] = 1  
  
print(counts)

1. Create an empty tuple
2. Count words and if word is found increase the count

Example:

{'Let': 1, 'me': 1, 'not': 2, 'to': 3, 'the': 3, 'marriage': 1, 'of': 1, 'true': 1, 'minds': 1, 'Admit': 1, 'impediments': 1, 'love': 2, 'is': 4, 'Which': 1, 'alters': 1, 'when': 1, 'it': 2, 'alteration': 1, 'finds': 1, '(A)': 1, 'Or': 1, 'bends': 1, 'with': 1, 'remover': 1, 'remove': 1, 'no': 1, 'an': 1, 'ever': 1, 'fixèd': 1, 'mark': 1, 'That': 1, 'looks': 1, 'on': 1, 'tempests': 1, 'and': 1, 'never': 1, 'shaken': 1, 'It': 1, 'star': 1, 'every': 1, "wand'ring": 1, 'bark': 1, 'Whose': 1, "worth's": 1, 'unknown': 1, 'although': 1, 'his': 1, 'height': 1, 'be': 1, 'taken\n': 1}

B) find the fewest words that will link them.

def minladder(beg, end, list):  
 *"""* ***:type*** *beginWord: object  
 """* list = set(list)  
 if end not in list:  
 return 0  
  
 curr = {beg}  
 res = 1  
 while curr:  
 list -= curr  
 next = set()  
 for word in curr:  
 for i in range(len(word)):  
 for c in 'abcdefghijklmnopqrstuvwxyz':  
 new = word[:i] + c + word[i + 1:]  
 if new == end:  
 return 1 + res  
 if new in list:  
 next.add(new)  
 curr = next  
 res += 1  
 return 0  
  
  
Lists = open('Words.txt').read().splitlines()  
  
start1 = "flour"  
target1 = "bread"  
start2 = "chaos"  
target2 = "peace"  
start3 = "river"  
target3 = "shore"  
start4 = "sleep"  
target4 = "dream"  
start5 = "black"  
target5 = "white"  
start6 = "witch"  
target6 = "fairy"  
start7 = "tears"  
target7 = "smile"  
start8 = "which"  
target8 = "think"  
start9 = "paper"  
target9= "story"  
start10 = "early"  
target10 = "trees"  
  
  
print(minladder(start1, target1, Lists))  
print(minladder(start2, target2, Lists))  
print(minladder(start3, target3, Lists))  
print(minladder(start4, target4, Lists))  
print(minladder(start5, target5, Lists))  
print(minladder(start6, target6, Lists))  
print(minladder(start7, target7, Lists))  
print(minladder(start8, target8, Lists))  
print(minladder(start9, target9, Lists))  
print(minladder(start10, target10, Lists))

1. Convert list into set for duplicate values
2. If end word is not in set return 0;
3. Create a dictionary ‘curr’. Initialize with start word.
4. Initialize res (answer)
5. Create an empty tuple next
6. For each word check if there exists a word which have just one letter difference
7. Update the res
8. Add word to next

Example:

7

8

9

8

8

13

7

9

10

9