**Practical 6**

**Aim:** You are given n words. Some words may repeat. For each word, output its number of occurrences. The output order should correspond with the input order of appearance of the word. See the sample input/output for clarification.

**Sample Input**

4

bcdef

abcdefg

bcde

bcdef

**Sample Output**

3

2 1 1

**Explanation:** There are 3 distinct words. Here, "bcdef" appears twice in the input at the first and last positions. The other words appear once each. The order of the first appearances are "bcdef", "abcdefg" and "bcde" which corresponds to the output.

s/w: PyCharm Community edition 2021.3

**Theory:**

Basically, we want to display all the unique Strings the number of times the String was duplicated.

First, we take input ‘n’ which is total numbers of words for next input.

We’ll be using dictionary for this program. We use key-value pair to store a String and number of times it has been taken as input.

While taking input, we will check if the String already exists in the dictionary; if it does, we will simply increment the corresponding value. Else we will add that String to the dictionary. This way we get total number of Strings in the dictionary without any duplicates while the corresponding value represents the number of times the String was repeated.

At the end we display the length of dictionary which is equivalent to total number of unique strings. We will also display the values of those Strings as an output.

dictionary.get(keyname, value) => The get() method returns the value of the item with the specified key.

len() will return the length of dictionary.

dictionary.values() => This will return list of values from the dictionary.

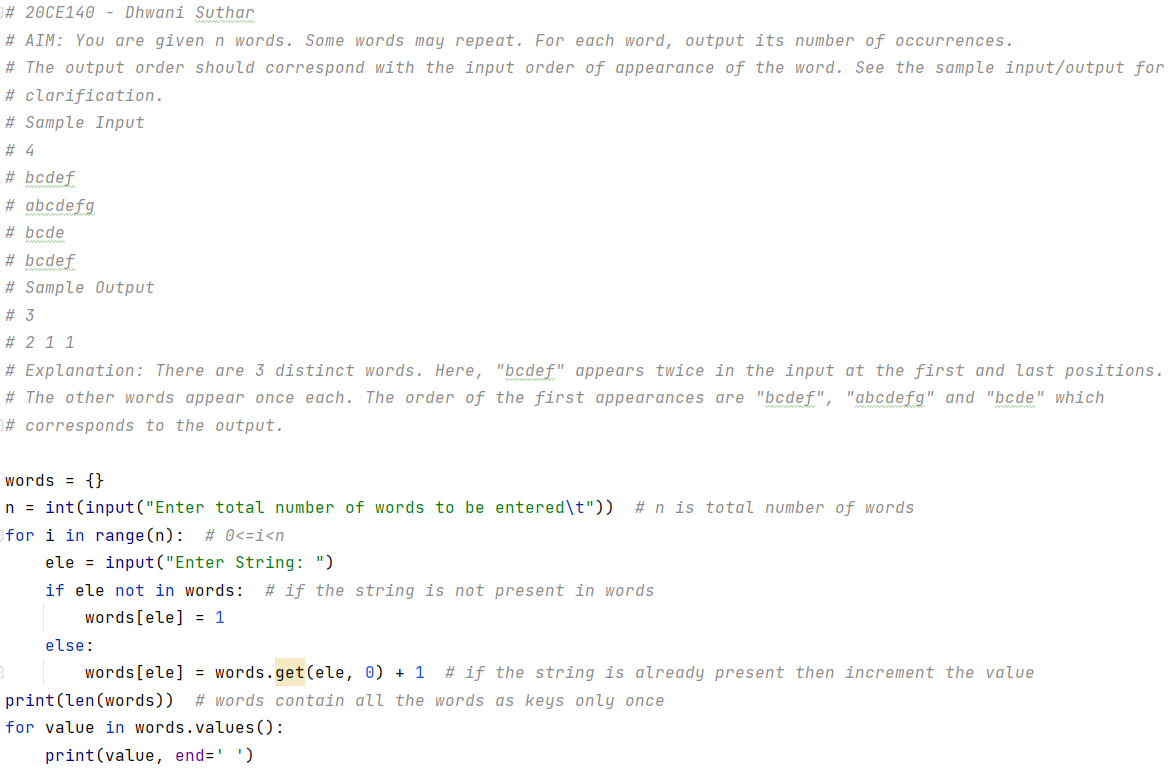
**Program:**

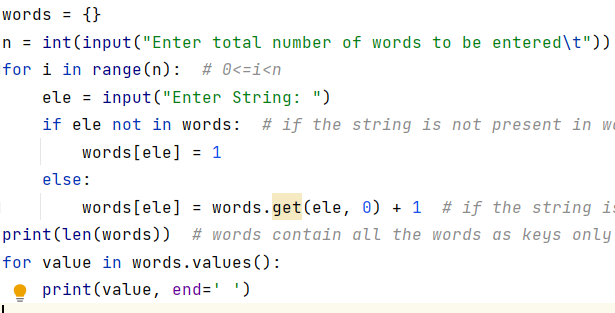
words = {}  
n = int(input("Enter total number of words to be entered\t"))

for i in range(n): *# 0<=i<n* ele = input("Enter String: ")  
 if ele not in words: *# if the string is not present in words* words[ele] = 1  
 else:  
 words[ele] = words.get(ele, 0) + 1 *# increment if present*

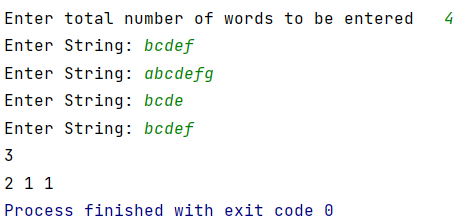
print(len(words)) *# words contain all the words as keys only once*

for value in words.values():  
 print(value, end=' ')





**Output:**



**Conclusion:**

Dictionary is used to store key:value pair information just like in this program we used it to store a unique String (key) and a number to denote the total number of times it occurs (value).

Dictionary items are ordered, changeable, and does not allow duplicates.

GitHub: