**Q. Same Letter Distances**

Same letter distance is the number of characters between two consecutive occurrences of the same character. Given a string, write a code to find the letter that has most frequently occurred in it. And also find same letter distance between the occurrences of the most frequently occurred character. If distance is zero then print ‘No characters’, if distance is 1 then print ‘1 character’ and if distance is ‘d’, greater than 1 then print “d characters”. For example, if the string is ‘accommodation’, the most frequently occurred character is ‘o’ and the same letter distances are

1 character

4 character

If the string is ‘bookkeeper’, the most frequently occurred character is ‘e’ and the same letter distances are

No characters

1 character

If more than one character occurs maximum number of times then consider the first character in the string.

Input Format

First line contains the string, w

Output Format

Print the letter that has occurred most frequently, ch

Print the same letter distance for consecutive occurrences of ch in next few lines

**Python Program**

word = input()  
occurence = 0  
for temp\_char in word:  
 if (word.count(temp\_char)) > occurence:  
 occurence = word.count(temp\_char)  
 char = temp\_char  
position = []  
temp\_position = 0  
for counter in range(0,occurence):  
 position.append((word.find(char,temp\_position)))  
 temp\_position = word.find(char,temp\_position) + 1  
print(char)  
for counter in range(0,(occurence-1)):  
 distance = position[counter+1] - position[counter]  
 if distance == 1:  
 print(**"No characters"**)  
 elif distance == 2:  
 print(**"1 character"**)  
 else:  
 print(**"%s characters"** %(distance-1))