Find duplicate characters in the String.

So we have got a problem to find the repeated characters from the String.

For example String is

Lenovo

o is repeated twice so output should be

o 2

banana

a is repeated 3 times, n is repeated 2 times.

a 3

n 2

We use Map data structure to solve the problem. We use key as the character and value as number of times that character is repeating.

We check the corner condition of string null, length 0 and length 1. If such string is input then we will just return.

Else we need to process the string. So now we declare the map and run the loop for length of string.

For every character in string if map contains the key if yes then increment the old value. Else set the value to 1.

Below program will give you proper idea of working of this problem. Go through the comments. I have written comments in such a way that you can understand it easily. Try to input it with different string. Please comment on this blog if there is anything incorrect or you can share more information about the topic.

**package** strings;

**import** java.util.LinkedHashMap;

**import** java.util.Map;

**import** java.util.Map.Entry;

**import** java.util.Set;

**public** **class** DuplicatesCharactersInString {

**public** **static** **void** main(String[] args) {

*findRepeatedCharacters*("a");

*findRepeatedCharacters*("characters");

*findRepeatedCharacters*("banana");

*findRepeatedCharacters*("mobile");

*findRepeatedCharacters*("laptop");

}

/\*\*

\* This method is used to find the repeated characters from the String.

\* \*/

**static** **void** findRepeatedCharacters(String str) {

/\*\*

\* If string is null then return, If string length is 0 or 1 then

\* return.

\* \*/

**if** (str == **null** || str.length() == 0 || str.length() == 1) {

**return**;

}

// Print the String

System.***out***.println(str);

/\*\*

\* We use LinkedHashMap<> class to store the data. K i.e key will be the

\* character. V i.e value will be the number of times character repeats.

\* \*/

Map<Character, Integer> map = **new** LinkedHashMap<Character, Integer>();

**for** (**int** i = 0; i < str.length(); i++) {

// if condition checks if character already exists

**if** (map.containsKey(str.charAt(i))) {

// if character exists then increment it count by 1

map.put(str.charAt(i), map.get(str.charAt(i)) + 1);

} **else** {

// if character does not exists then put its count as 1

map.put(str.charAt(i), 1);

}

}

/\*\*

\* Returns view of the mappings contained in this map.

\* \*/

Set<Entry<Character, Integer>> entrySet = map.entrySet();

**for** (Map.Entry<Character, Integer> entry : entrySet) {

// Condition to check if the character is repeated more than 1

// time.

**if** (entry.getValue() > 1) {

System.***out***.println(entry.getKey() + " : " + entry.getValue());

}

}

System.***out***.println();

}

}

Output

characters

c : 2

a : 2

r : 2

banana

a : 3

n : 2

mobile

laptop

p : 2