Strings Logical Questions in JavaScript

Q-1. Given a string **str** containing only lower case alphabets, the task is to sort it in lexicographically-descending order.

Sample Input :

Input: str = "geeks"
Output: "skgee"
Explanation: It's the lexicographically-descending order.

Q-2. Given a string **S**, Check if characters of the given string can be rearranged to form a palindrome.

Note: You have to return 1 if it is possible to convert the given string into palindrome else return 0.

Sample Input :

Input:
S = "geeksogeeks"
Output: Yes
Explanation: The string can be converted into a palindrome: geeksoskeeg

Q-3. In this problem, a *String* **S** is composed of lowercase alphabets and wildcard characters i.e. '**?'.** Here, '?' can be replaced by any of the lowercase alphabets. Now you have to classify the given *String* on the basis of following rules:

If there are more than **3 consonants together or more than 5 vowels together**, the *String* is considered to be "**BAD**". A *String* is considered "**GOOD" only if it is not "BAD"**.

NOTE: *String* is considered as "BAD" if the above condition is satisfied even once. Else it is "GOOD" and the task is to make the string "BAD".

Sample Input :

Input:
S = aeioup??
Output:
1
Explanation: The String doesn't contain more than 3 consonants or more than 5 vowels together. So, it's a GOOD string.

Q-4. Given a String S, reverse the string without reversing its individual words. Words are separated by dots.

Sample Input :

Input:
S = i.like.this.program.very.much
Output: much.very.program.this.like.i
Explanation: After reversing the whole string(not individual words), the input string becomes much.very.program.this.like.i

Q-5. Given a string **S**, check if it is palindrome or not.

Sample Input :

Input: S = "abba"
Output: 1
Explanation: S is a palindrome

Q-6. Given a string **S** consisting of **lowercase** Latin Letters. Return the first non-repeating character in S. If there is no non-repeating character, return **'$'.**

Sample Input :

Input:
S = hello
Output: h
Explanation: In the given string, the first character which is non-repeating is h, as it appears first and there is no other '**h**' in the string.

Q-7. Given a string, our task is finding the occurrence of a character in the string with the help of user-defined function.

Sample Input :

```
Example:
Input : "hello"
Output : h occur 1 times
         e occur 1 times
         l occur 2 times
         o occur 1 times
Explanation : here "hello" have 1 h, so it have 1 value.
               as same e have 1, l have 2 , o have 1.
```

Q-8. Given a string S, the task is to remove all the duplicates in the given string.

Below are the different methods to remove duplicates in a string.

Sample Input :

```
Input string:  geeksforgeeks
1) Sort the characters
   eeeefggkkorss
2) Remove duplicates
    efgkorskkorss
3) Remove extra characters
     efgkors
```


Q-9. Given a string str, find the length of the longest substring without repeating characters.

Sample Input :

For "ABDEFGABEF", the longest substring are "BDEFGA" and "DEFGAB", with length 6.

For "BBBB" the longest substring is "B", with length 1.

Q-10. Two strings are said to be isomorphic if it is possible to map every character of the first string to every character of the second string. Basically, in isomorphic strings, there is a one-to-one mapping between every character of the first string to every character of the second string.

Sample Input :

```
str1 = 'ABCA'
str2 = 'XYZX'
'A' maps to 'X'
'B' maps to 'Y'
'C' maps to 'Z'
```