String Logical Questions Solutions

Q-1.

Solution :

```
<script>
	let MAX_CHAR = 26;
	// function to print string in sorted order
	function sortString(str)
	{
		// Hash array to keep count of characters. Initially count of all charters is
		// initialized to zero.
		let charCount = new Array(MAX_CHAR);
		for(let i = 0; i < charCount.length; i++)
		{
			charCount[i] = 0;
		}
		// Traverse string and increment count of characters 'a'-'a' will be 0, 'b'-'a' will be 1,
		for (let i = 0; i < str.length; i++)
		{
			// so for location of character in count array we will do str[i]-'a'.
			charCount[str[i].charCodeAt(0) - 'a'.charCodeAt(0)]++;
		}
		// Traverse the hash array and print characters
		for (let i = MAX_CHAR - 1; i >= 0; i--)
		{
			for (let j = 0; j < charCount[i]; j++)
			{
				document.write(String.fromCharCode ('a'.charCodeAt(0) + i));
```

CAREER INFOTECH

```
                        }
                }
        }
        // Driver code
        let s = "alkasingh";
        sortString(s);
</script>
```

Q-2.

Solution :

```
#include <bits/stdc++.h>
using namespace std;
#define NO_OF_CHARS 256
/* function to check whether characters of a string can form a palindrome */
bool canFormPalindrome(string str)
{
        // Create a count array and initialize all values as 0
        int count[NO_OF_CHARS] = { 0 };
        // For each character in input strings, increment count in the corresponding count array
        for (int i = 0; str[i]; i++)
                count[str[i]]++;
        // Count odd occurring characters
        int odd = 0;
        for (int i = 0; i < NO_OF_CHARS; i++) {
                if (count[i] & 1)
                        odd++;
                if (odd > 1)
                        return false;
        }
```

CAREER INFOTECH

```
        // Return true if odd count is 0 or 1,
        return true;
}
/* Driver code*/
int main()
{
        canFormPalindrome("geeksforgeeks")
        ? cout << "Yes\n"
        : cout << "No\n";
        canFormPalindrome("geeksogeeks")
        ? cout << "Yes\n"
        : cout << "No\n";
        return 0;
}
```

CAREER INFOTECH

Q-3.

Solution :

```
class Solution
{
    static int isGoodorBad(String S)
    {
        HashSet<Character> set = new HashSet<Character>();
        set.add('a');
        set.add('e');
        set.add('i');
        set.add('o');
        set.add('u');

        int vow = 0, cons = 0;

        if(set.contains(S.charAt(0)))
          vow++;

        else if(S.charAt(0) != '?')
          cons++;

        else
        {
          vow++;
          cons++;
        }

        for(int i = 1; i < S.length(); i++)
        {
            if(set.contains(S.charAt(i)))
            {
                    cons = 0;
                    vow++;
            }

            else if (S.charAt(i) != '?')
            {
                vow = 0;
                cons++;
            }
            else
            {
                vow++;
                cons++;
            }

            if(vow > 5)
            {
                return 0;
            }
            if(cons > 3)
            {
                return 0;
            }
        }
        return 1;
    }
}
```

Q-4.

Solution :

```
<script>
	function reverse(str,start,end)
	{
		// Temporary variable to store character
		let temp;
		while (start <= end)
		{
			// Swapping the first and last character
			temp = str[start];
			str[start]=str[end];
			str[end]=temp;
			start++;
			end--;
		}
	}
	// Function to reverse words
	function reverseWords(s)
	{
		// Reversing individual words as explained in the first step
		s=s.split("");
		let start = 0;
		for (let end = 0; end < s.length; end++)
		{
	// If we see a space, we reverse the previous word (word between the indexes start and end-1
			// i.e., s[start..end-1]
			if (s[end] == ' ')
			{
```

CAREER INFOTECH

```
                    reverse(s, start, end);
                    start = end + 1;
                }
            }
            // Reverse the last word
            reverse(s, start, s.length - 1);
            // Reverse the entire String
            reverse(s, 0, s.length - 1);
            return s.join("");
        }
        // Driver Code
        var s = "i like this program very much ";
        document.write(reverseWords(s));
        </script>
```

CAREER INFOTECH

Q-5.

Solution :

```
class Solution
{
    int isPlaindrome(String S)
    {
        int len = S.length();
        int end = len-1,start = 0;
        while(start < end)
        {
            char c1 = S.charAt(start);
            char c2 = S.charAt(end);
            if(c1 != c2)
            {
               return 0;
            }
            start++;
            end--;
        }
        return 1;
    }
}
```

Q-6.

Solution :

```
<script>
function firstNonRepeating(str)
{
	var fi=new Array(256);
	// array to store First Index
	fi.fill(-1);
	// initializing all elements to -1
	for(var i = 0; i<256; i++)
		fi[i] = -1;
// sets all repeating characters to -2 and non-repeating characters contain the index where they occur
	for(var i = 0; i<str.length; i++)
	{
		if(fi[str.charCodeAt(i)] ==-1)
		{
			fi[str.charCodeAt(i)] = i;
		}
		else
		{
			fi[str.charCodeAt(i)] = -2;
		}
	}
	var res = Infinity;
	for(var i = 0; i<256; i++) {
		// If this character is not -1 or -2 then it means that this character occurred only once
		// so find the min index of all characters that occur only once, that's our first index
		if(fi[i] >= 0)
			res = Math.min(res, fi[i]);
```

CAREER INFOTECH

```
        }
// if res remains INT_MAX, it means there are no characters that repeat only once or the string is empty
        if(res == Infinity) return -1;
        else return res;
}
var str;
        str = "geeksforgeeks";
        var firstIndex = firstNonRepeating(str);
        if (firstIndex === -1)
                document.write("Either all characters are repeating or string is empty");
        else
                document.write("First non-repeating character is "+ str.charAt(firstIndex));
</script>
```

CAREER INFOTECH

Q-7.

Solution :

```
        <script>
        function count(s, c)
        {
                let res = 0;
                for (let i = 0; i < s.length; i++)
                {
                        // checking character in string
                        if (s.charAt(i) == c)
                        res++;
                }
                return res;
        }
```

```
        // Driver method
                let str= "geeksforgeeks";
                let c = 'e';
                document.write(count(str, c));
</script>
```

Q-8.

Solution :

```
<script>
        function removeDuplicate( str , n)
        {
                // Create a set using String characters excluding '\0'
                var s = new Set();
                // HashSet doesn't allow repetition of elements
                for (var i = 0;i<n;i++)
                        s.add(str[i]);
                // Print content of the set
                for (const v of s) {
                        document.write(v);
        }
        }
        // Driver code
                var str = "geeksforgeeks";
                var n = str.length;
                removeDuplicate(str, n);
</script>
```

CAREER INFOTECH

Q-9.

Solution :

```
<script>
function longestUniqueSubsttr(str)
{
	var n = str.length;
	// Result
	var res = 0;
	for(var i = 0; i < n; i++)
	{
		// Note : Default values in visited are false
		var visited = new Array(256);
		for(var j = i; j < n; j++)
		{
			// If current character is visited Break the loop
			if (visited[str.charCodeAt(j)] == true)
				break;
		// Else update the result if this window is larger, and mark current character as visited.
				else
				{
					res = Math.max(res, j - i + 1);
					visited[str.charCodeAt(j)] = true;
				}
		}
	}
	return res;
}
```

CAREER INFOTECH

```
// Driver code

        var str = "geeksforgeeks";

        document.write("The input string is " + str);

        var len = longestUniqueSubsttr(str);

        document.write("The length of the longest " +"non-repeating character " +
        "substring is " + len);

</script>
```

CAREER INFOTECH