

## Ques1)

```
js-classroom > remove-duplicates.js > removeDuplicates
1 function removeDuplicates(str) {
2   let stack = [];
3
4   for (let i = 0; i < str.length; i++) {
5     if (stack.length != 0 && stack[stack.length - 1] === str[i]) {
6       stack.pop();
7     } else stack.push(str[i]);
8   }
9   return stack.join("");
10 }
11
12 console.log(removeDuplicates("abbaca"));
13 console.log(removeDuplicates("azxxzy"));
14
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\visual studio code\js-geek\js-classroom> node .\remove-duplicates.js
ca
ay
PS E:\visual studio code\js-geek\js-classroom>
```

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF {} JavaScript Go Live Quokka ✓ Prettier

## Ques2)

```
1 function areAnagrams(s, t) {
2   if (s.length !== t.length) return false;
3
4   let arr = new Array(26).fill(0);
5
6   for (let i = 0; i < s.length; i++) {
7     arr[s.charCodeAt(i) - "a".charCodeAt(0)]++;
8     arr[t.charCodeAt(i) - "a".charCodeAt(0)]--;
9   }
10
11   for (let i = 0; i < 26; i++) {
12     if (arr[i] !== 0) return false;
13   }
14   return true;
15 }
16
17 console.log(areAnagrams("anagram", "nagaram"));
18 console.log(areAnagrams("rat", "car"));
19
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\visual studio code\js-geek\js-classroom> node .\valid-anagram.js
true
false
PS E:\visual studio code\js-geek\js-classroom>
```

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF {} JavaScript Go Live Quokka ✓ Prettier

### Ques3)

```
js-classroom > decode-string.js > isDigit
1 function isDigit(num) {
2   let temp = Number(num);
3   return temp >= 0 && temp <= 9;
4 }
5 function isAlpha(ch) {
6   return typeof ch === "string" && ch >= "a" && ch <= "z";
7 }
8 function decodeString(str) {
9   let stack = [];
10  let num = 0;
11  let result = "";
12  for (let i = 0; i < str.length; i++) {
13    if (isDigit(str[i])) {
14      num = num * 10 + Number(str[i]);
15    } else if (isAlpha(str[i])) {
16      result += str[i];
17    } else if (str[i] === "[") {
18      stack.push(result);
19      stack.push(num);
20      result = "";
21      num = 0;
22    } else if (str[i] === "]") {
23      let repeat = Number(stack.pop());
24      let tempString = stack.pop();
25      for (let k = 0; k < repeat; k++) {
26        tempString += result;
27      }
28      result = tempString;
29    }
30  }
31  return result;
32 }
33
34 console.log(decodeString("3[a]2[bc]"));
35 console.log(decodeString("3[a2[c]]"));
36 console.log(decodeString("2[abc]3[cd]ef"));
37
```

PS E:\visual studio code\js-geek\js-classroom> node .\decode-string.js  
aaabcbc  
accaccacc  
abcabccdcdef  
PS E:\visual studio code\js-geek\js-classroom>

### Ques4)

```
js-classroom > reorganize-string.js > reorganizeString
1 function reorganizeString(str) {
2   let freq = new Array(26).fill(0);
3   let res = new Array(str.length).fill("");
4
5   for (let i = 0; i < str.length; i++) {
6     freq[str.charCodeAt(i) - "a".charCodeAt(0)]++;
7   }
8   let maxOcc = 0;
9   let maxOccEl = "";
10  for (let i = 0; i < 26; i++) {
11    if (maxOcc < freq[i]) {
12      maxOcc = freq[i];
13      maxOccEl = String.fromCharCode(i + "a".charCodeAt(0));
14    }
15  }
16  let ind = 0;
17  while (maxOcc > 0 && ind < str.length) {
18    res[ind] = maxOccEl;
19    ind += 2;
20    maxOcc--;
21  }
22  if (maxOcc > 0) return "Not Possible";
23  freq[maxOccEl - "a".charCodeAt(0)] = 0;
24
25  for (let i = 0; i < 26; i++) {
26    while (freq[i] > 0) {
27      ind = ind < str.length ? ind : 1;
28      res[ind] = String.fromCharCode(i + "a".charCodeAt(0));
29      ind += 2;
30      freq[i]--;
31    }
32  }
33  return res.join("");
34 }
35 console.log(reorganizeString("aab"));
36 console.log(reorganizeString("aaab"));
37
```

PS E:\visual studio code\js-geek\js-classroom> node .\reorganize-string.js  
aba  
Not Possible  
PS E:\visual studio code\js-geek\js-classroom>