Ques1)

```javascript
js-classroom > JS string-reversal.js > ✪ stringReverse
 1  function stringReverse(str) {
 2    let arr = str.split("");
 3    let i = 0;
 4    let j = arr.length - 1;
 5    while (i < j) {
 6      let temp = arr[i];
 7      arr[i] = arr[j];
 8      arr[j] = temp;
 9      i++;
10      j--;
11    }
12    return arr.join("");
13  }
14
15  console.log(stringReverse("jashan madan"));
16  console.log(stringReverse("web development"));
17
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS E:\visual studio code\js-geek\js-classroom> node .\string-reversal.js
nadam nahsaj
tnempoleved bew
PS E:\visual studio code\js-geek\js-classroom>
```

Ln 6, Col 5 (18 selected)   Spaces: 2   UTF-8   CRLF   {} JavaScript   Go Live   Quokka   Prettier

Ques2)

```javascript
js-classroom > JS valid-anagram.js > ✪ areAnagrams
 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
function arrayIntersection(arr1, arr2) {
  let s = new Set();
  let result = [];
  for (const el of arr1) {
    s.add(el);
  }
  for (const el of arr2) {
    if (s.has(el)) {
      result.push(el);
      s.delete(el);
    }
  }
  return result;
}

console.log(arrayIntersection([1, 2, 4], [2, 4, 5]));
console.log(arrayIntersection([6, 9, 8], [0, 1, 3]));
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS E:\visual studio code\js-geek\js-classroom> node .\array-intersection.js
[ 2, 4 ]
[]
PS E:\visual studio code\js-geek\js-classroom>
```

Ques4)

```js
function isAlphaNumeric(ch) {
  return (
    typeof ch === "number" ||
    (typeof ch === "string" &&
      ((ch >= "a" && ch <= "z") || (ch >= "A" && ch <= "Z")))
  );
}
function isPalindrome(str) {
  let arr = [];
  for (let i = 0; i < str.length; i++) {
    if (isAlphaNumeric(str[i])) {
      arr.push(str[i].toLowerCase());
    }
  }
  let i = 0;
  let j = arr.length - 1;
  while (i < j) {
    if (arr[i] !== arr[j]) return false;
    i++;
    j--;
  }
  return true;
}

console.log(isPalindrome("A man, a plan, a canal: Panama"));
console.log(isPalindrome("race a car"));
```

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

Ques5)

```javascript
function reverse(start, end, arr) {
  let i = start,
    j = end;
  while (i < j) {
    let temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
    i++;
    j--;
  }
  return arr;
}
function rotateArray(arr, k) {
  let n = arr.length;
  arr = reverse(0, n - k - 1, arr);
  arr = reverse(n - k, n - 1, arr);
  arr = reverse(0, n - 1, arr);
  return arr;
}

console.log(rotateArray([1, 2, 3, 4, 5, 6, 7], 3));
console.log(rotateArray([-1, -100, 3, 99], 2));
console.log(rotateArray([1, 2, 3, 4, 5, 6, 7, 8, 9], 3));
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\visual studio code\js-geek\js-classroom> node .\rotate-array.js
[
  5, 6, 7, 1,
  2, 3, 4
]
[ 3, 99, -1, -100 ]
[
  7, 8, 9, 1, 2,
  3, 4, 5, 6
]
PS E:\visual studio code\js-geek\js-classroom>
```

Ques6)

```javascript
function stringCompression(s) {
  let n = s.length,
    j = 0,
    res = "";
  while (j < n) {
    let ch = s[j],
      ct = 1;
    while (j + 1 < n && ch === s[j + 1]) {
      ct++;
      j++;
    }
    res += s[j] + ct;
    j++;
  }
  return res;
}

console.log(stringCompression("aabcccccaaa"));
console.log(stringCompression("aabbccc"));
console.log(stringCompression("abbb"));
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\visual studio code\js-geek\js-classroom> node .\string-compression.js
a2b1c5a3
a2b2c3
a1b3
PS E:\visual studio code\js-geek\js-classroom>
```

Ques7)

```
js-classroom > JS array-sum.js > ...
  1   function findPairWithSum(arr, targetSum) {
  2     let pairs = [];
  3     let map = {};
  4
  5     for (let i = 0; i < arr.length; i++) {
  6       let remaining = targetSum - arr[i];
  7       if (map[remaining] !== undefined) {
  8         pairs.push([remaining, arr[i]]);
  9       }
 10       map[arr[i]] = i;
 11     }
 12     return pairs;
 13   }
 14
 15   console.log(findPairWithSum([1, 2, 3, 4, 5, 6, 7], 6));
 16   console.log(findPairWithSum([3, 8, 12, 4, 6, 1], 9));
 17   console.log(findPairWithSum([5, 2, 8, 10, 3], 12));
 18
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS E:\visual studio code\js-geek\js-classroom> node .\array-sum.js
[ [ 2, 4 ], [ 1, 5 ] ]
[ [ 3, 6 ], [ 8, 1 ] ]
[ [ 2, 10 ] ]
PS E:\visual studio code\js-geek\js-classroom>
```

Ques8)

```
js-classroom > JS longest-substring.js > ✪ longestLength
  1   function longestLength(str) {
  2     let map = {};
  3     let j = 0,
  4       res = 1;
  5     for (let i = 0; i < str.length; i++) {
  6       if (map.hasOwnProperty(str[i])) {
  7         if (j < map[str[i]]) {
  8           j = map[str[i]];
  9         }
 10       }
 11       if (i - j + 1 > res) {
 12         res = i - j + 1;
 13       }
 14       map[str[i]] = i + 1;
 15     }
 16     return res;
 17   }
 18
 19   console.log(longestLength("abcabcbb"));
 20   console.log(longestLength("bbbbb"));
 21   console.log(longestLength("pwwkew"));
 22
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS E:\visual studio code\js-geek\js-classroom> node .\longest-substring.js
3
1
3
PS E:\visual studio code\js-geek\js-classroom>
```