

Untitled   

JAVASCRIPT  

```
1 const arr = [1 , 2, 4, 9, 12, 13, 20];
2 const oddNumbers = []
3 for(let i = 0; i < arr.length; i++){
4     arr[i] % 2 === 1 && oddNumbers.push(arr[i]);
5 }
6 console.log(oddNumbers);
7 |
```

Input Output

 Debug

 Explain

 Generate Code



Output:

```
[ 1, 9, 13 ]
```

Execution Time:

```
0.036s
```

Memory Used:

```
9056kb
```

```
1 function sentenceCase(str) {
2     if ((str === null) || (str === '')) {
3         return false;
4     } else {
5         str = str.toString();
6     }
7     return str.replace(/\w\S*/g,
8         function (txt) {
9             return txt.charAt(0).toUpperCase() +
10                txt.substr(1).toLowerCase();
11         });
12 }
13
14 console.log(sentenceCase('geeks for geeks'));
15
16
17
```

Input Output

Debug

Explain

Generate Code

**Output:**

Geeks For Geeks

Execution Time:

0.037s

Memory Used:

8880kb

```
1 // Creating array
2 let arr = [10, 8, 17, 13, 22]
3
4 // Creating variable to store the sum
5 let sum = 0;
6
7 // Running the for loop
8 for (let i = 0; i < arr.length; i++) {
9     sum += arr[i];
10 }
11
12 console.log("Sum is " + sum) // Prints: 44
13
14
15
16
```

Input Output

Debug

Explain

Generate Code



Output:

Sum is 70

Execution Time:

0.036s

Memory Used:

8932kb

```
1 function isprime(num) {
2     if (num < 2){
3         return false;
4     }
5     for(let i= 2; i <= Math.sqrt(num); i++) {
6         if (num % i === 0) {
7             return false;
8         }
9     }
10    return true;
11 }
12 }
13 function findprimes(n) {
14     const primes = [2,3,5,7,11,13,15,17,19];
15     for (let i = 2; i <= n; i++){
16         if (isprime(i)){
17             findprimes.push (i);
18         }
19     }
20 }
21 return primes;
22 }
23 console. log(findprimes(20))
24 |
25
```

Input Output

Debug

Explain

Generate Code



```
1 function isPalindrome(str) {  
2     let j = str.length - 1  
3     for (let i = 0; i < str.length / 2; i++) {  
4         if (str[i] != str[j]) {  
5             return false;  
6         }  
7         j--;  
8     }  
9     return true;  
10 }  
11  
12 let str1 = "car";  
13 let str2 = "nitin";  
14 let str3 = "maha";  
15  
16 console.log(isPalindrome(str1));  
17 console.log(isPalindrome(str2));  
18 console.log(isPalindrome(str3));  
19  
20  
21
```

Input Output

Debug

Explain

Generate Code



Output:

```
false  
true  
false
```

Execution Time:

```
0.037s
```

Memory Used:

```
9096kb
```

```
66         }
67         // For case when j<m,
68         else
69         {
70             m1 = ar2[j++];
71         }
72     }
73 }
74 return (m1 + m2) / 2;
75 }
76 }
77
78 // Driver code
79 let ar1 = [1000];
80 let ar2 = [3, 8, 10, 20];
81
82 let n1 = ar1.length;
83 let n2 = ar2.length;
84 console.log(getMedian(ar1, ar2, n1, n2));
85
86 // This code is contributed by divyeshrabiya07
87
88
89
90 |
```

Input Output

Debug

Explain

Generate Code



Output:

10

Execution Time:

0.036s

Memory Used:

9060kb

```
1 let arr = ["apple", "mango", "apple",
2             "orange", "mango", "mango"];
3
4 function removeDuplicates(arr) {
5     return [...new Set(arr)];
6 }
7
8 console.log(removeDuplicates(arr));
9
10
11
12
13
```

Run

Input Output

Debug Explain Generate Code ►

Output:

['apple', 'mango', 'orange']

Execution Time:

0.037s

Memory Used:

9020kb

```
1 let Arr = ['ABC_1', 'ABC_2', 'ABC_3', 'ABC_4'];
2
3 function arrayRotate(arr) {
4     arr.push(arr.shift());
5     return arr;
6 }
7
8 function myABC() {
9     let rotateArr = arrayRotate(Arr);
10    console.log("Elements of array = ["
11        + rotateArr + "]");
12 }
13
14 myABC();
15
16
17
18
19
20
```

Run

Input Output

Debug Explain Generate Code

**Output:**

Elements of array = [ABC_2,ABC_3,ABC_4,ABC_1]

Execution Time:

0.036s

Memory Used:

8936kb

```
1 const arr = [3, 2, 5, 9, 12, 13, 20];
2 const oddNumbers = arr.filter((num) => num%2 === 1);
3 console.log(oddNumbers);
4
5
6
7
8
9
10
```

Input Output

Debug

Explain

Generate Code

**Output:**

```
[ 3, 5, 9, 13 ]
```

Execution Time:

```
0.039s
```

Memory Used:

```
9108kb
```

```
1 function sentenceCase(str) {
2     if ((str === null) || (str === '')) {
3         return false;
4     } else {
5         str = str.toString();
6     }
7     return str.replace(/\w\S*/g,
8         function (txt) {
9             return txt.charAt(0).toUpperCase() +
10                txt.substr(1).toLowerCase();
11         });
12 }
13
14 console.log(sentenceCase('get and get'));
15
16
17
18
19
20
21
22
```

Input Output

Debug

Explain

Generate Code



Output:

Get And Get

Execution Time:

0.037s

Memory Used:

8892kb

```
1  sum = (a, b) => {
2    return(a+b)
3  }
4  x= sum(100,78)
5  console.log(x)
6
7
8
9
10
11
12
```

Input Output

Debug

Explain

Generate Code

**Output:**

```
178
```

Execution Time:

```
0.037s
```

Memory Used:

```
8928kb
```

```
1 const newArray = [1, 3, 7, 5, 10, 13];
2 const myPrimeArray = newArray.filter(num => {
3   for (let i = 2; i < num; i++) {
4     if (num % i === 0) return false;
5   }
6   return num !== 1;
7 });
8 });
9 console.log(myPrimeArray);
10
11
12
13
14
15
16
```

Input Output

Debug

Explain

Generate Code

**Output:**

```
[ 3, 7, 5, 13 ]
```

Execution Time:

```
0.038s
```

Memory Used:

```
9020kb
```