

MERN STACK DEVELOPMENT

Functions – Task 5

1. Do the below programs in anonymous function & IIFE

a. Print odd numbers in an array

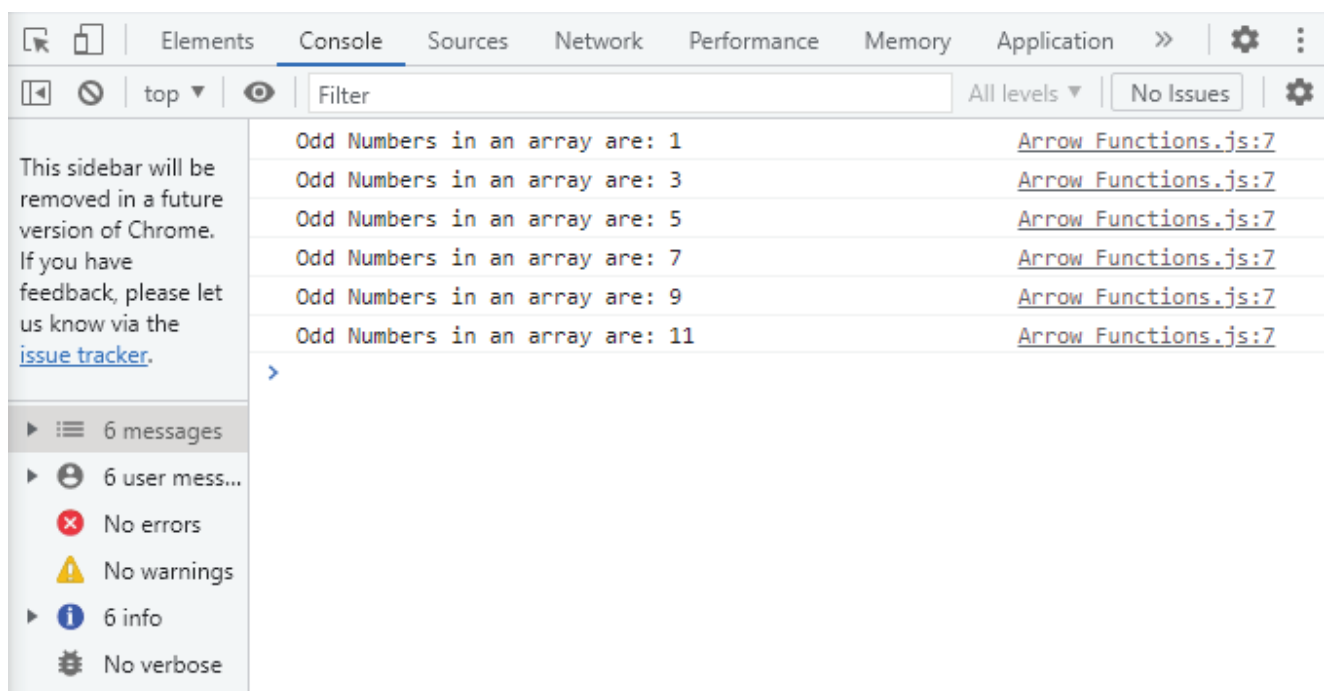
```
let numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11];
```

```
//Print Odd Numbers in an array Using Anonymous Function
```

```
const getOddNumbers = function(){  
  for(const odd of numbers){  
    if(odd % 2 !== 0){  
      console.log(`Odd Numbers in an array are: ${odd}`);  
    }  
  }  
};  
getOddNumbers();
```

```
//Print Odd Numbers in an array Using IIFE
```

```
(function(){  
  for(const odd of numbers){  
    if(odd % 2 !== 0){  
      console.log(`Odd Numbers in an array are: ${odd}`);  
    }  
  }  
})(numbers);
```



MERN STACK DEVELOPMENT

Functions – Task 5

b. Convert all the strings to title caps in a string array

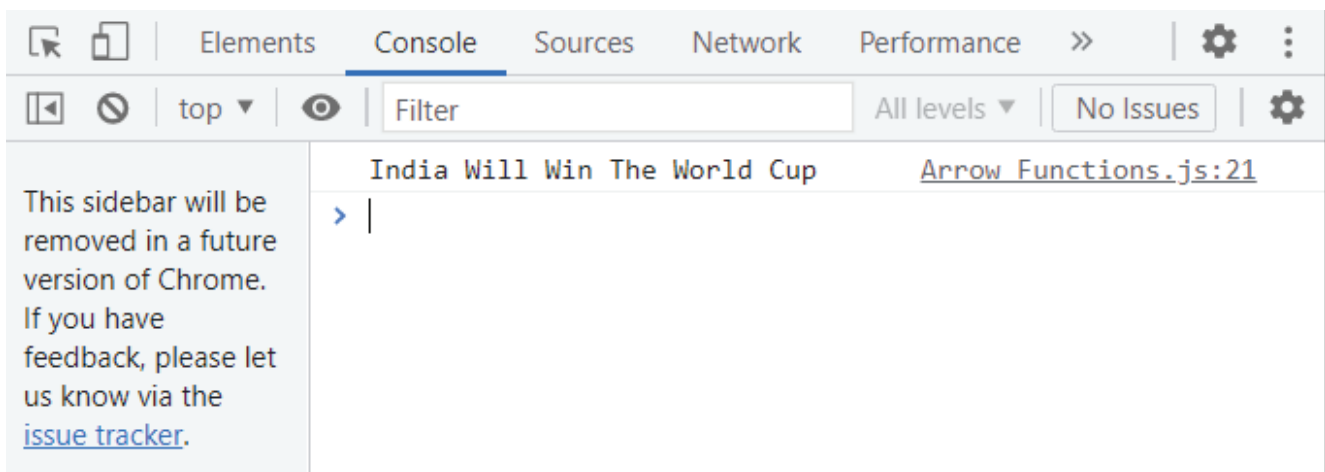
//Convert all the strings to title caps in a string array using Anonymous Function

```
let strings = ["India", "will", "win", "the", "world", "cup"];
```

```
const changetoTitleCase = function(){  
  for(let i = 0; i < strings.length; i++){  
    strings[i] = strings[i].charAt(0).toUpperCase() + strings[i].slice(1);  
  }  
  return strings.join(' ');  
}  
console.log(changetoTitleCase());
```

//Convert all the strings to title caps in a string array using IIFE

```
(function(){  
  for(let i in strings){  
    strings[i] = strings[i].charAt(0).toUpperCase() + strings[i].slice(1);  
  }  
  console.log(strings.join(" "));  
})(strings);
```



MERN STACK DEVELOPMENT

Functions – Task 5

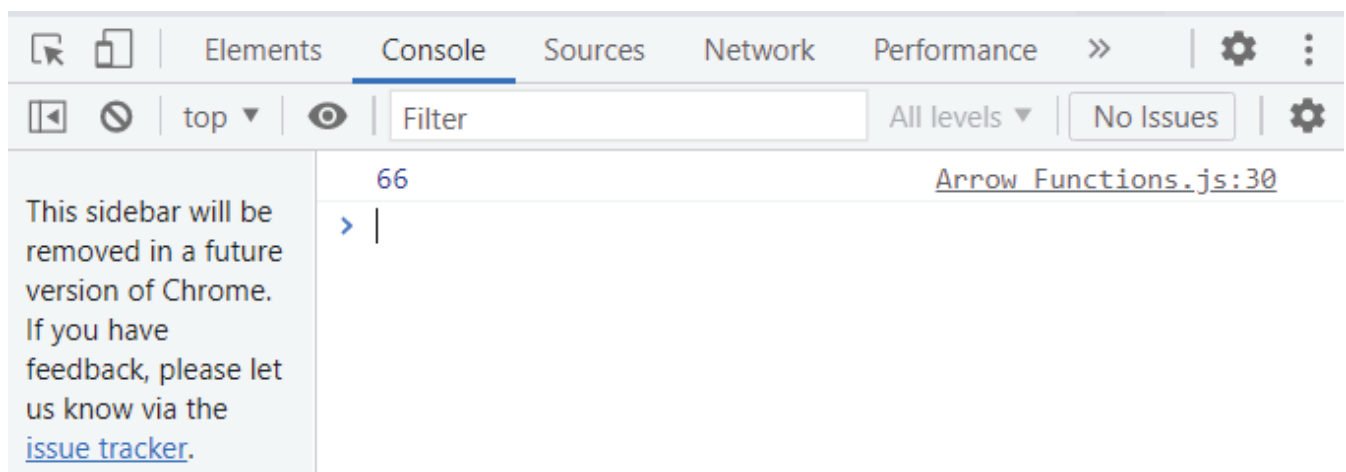
c. Sum of all numbers in an array

//Sum of all numbers in an array using Anonymous Function

```
const getSumofArray = function(){  
  let sum = 0;  
  for (let i in numbers){  
    sum += numbers[i];  
  }return sum;  
}  
console.log(getSumofArray());
```

//Sum of all numbers in an array using IIFE

```
(function(){  
  let sum = 0;  
  for (let i in numbers){  
    sum += numbers[i];  
  }  
  console.log(sum);  
})();
```



MERN STACK DEVELOPMENT

Functions – Task 5

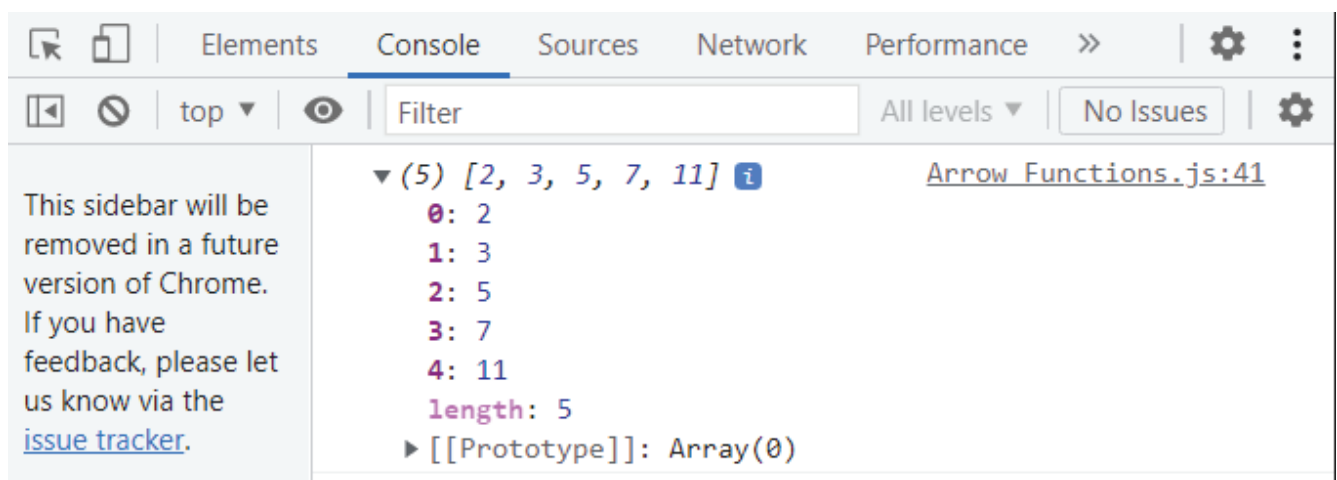
d. Return all the prime numbers in an array

//Return all the prime numbers in an array using Anonymous Function

```
const getPrimeNumbers = function(num){  
  for(let i = 2; num > i; i++){  
    if(num % i == 0){  
      return false;  
    }  
  }  
  return num > 1;  
}  
console.log(numbers.filter(getPrimeNumbers));
```

//Return all the prime numbers in an array using IFE

```
(function(num){  
  for(let i = 2; num > i; i++){  
    if(num % i == 0){  
      return false;  
    }  
  }  
  return num > 1;  
})(numbers);  
console.log(numbers.filter(getPrimeNumbers));
```



MERN STACK DEVELOPMENT

Functions – Task 5

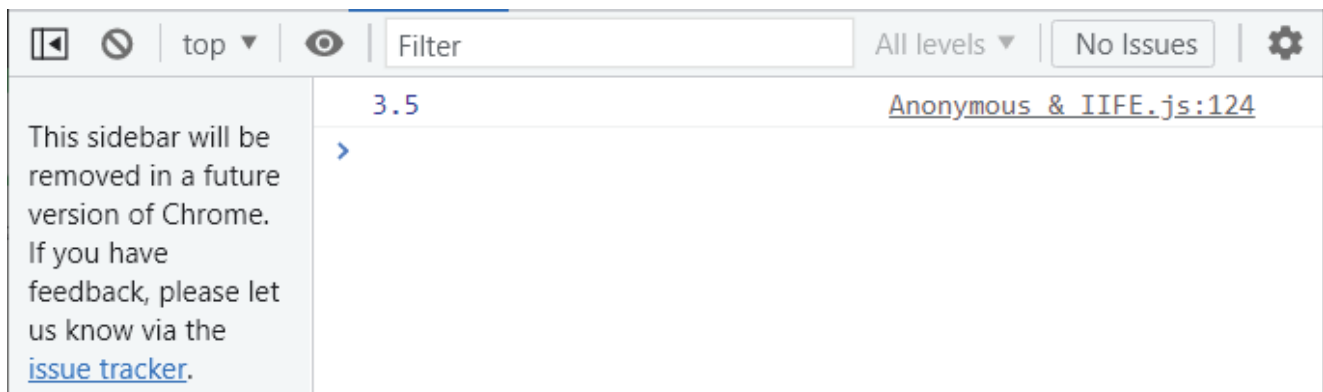
e. Return median of two sorted arrays of same size

//Median of two sorted arrays of same size using Anonymous Function

```
let num1 = [1, 3, 2];
let num2 = [5, 6, 4];
let arr = [...num1, ...num2];
const median = function (arr) {
  const midNum = Math.floor(arr.length / 2),
    sortedArray = [...arr].sort((a, b) => a - b);
  if (arr.length % 2 !== 0) {
    return sortedArray[midNum];
  } else {
    return (sortedArray[midNum - 1] + sortedArray[midNum]) / 2;
  }
};
console.log(median(arr));
```

//Median of two sorted arrays of same size using IIFE

```
(function (arr) {
  const midNum = Math.floor(arr.length / 2),
    sortedArray = [...arr].sort((a, b) => a - b);
  if (arr.length % 2 !== 0) {
    let result = sortedArray[midNum];
    console.log(result);
  } else {
    let result = (sortedArray[midNum - 1] + sortedArray[midNum]) / 2;
    console.log(result);
  }
})(arr);
```



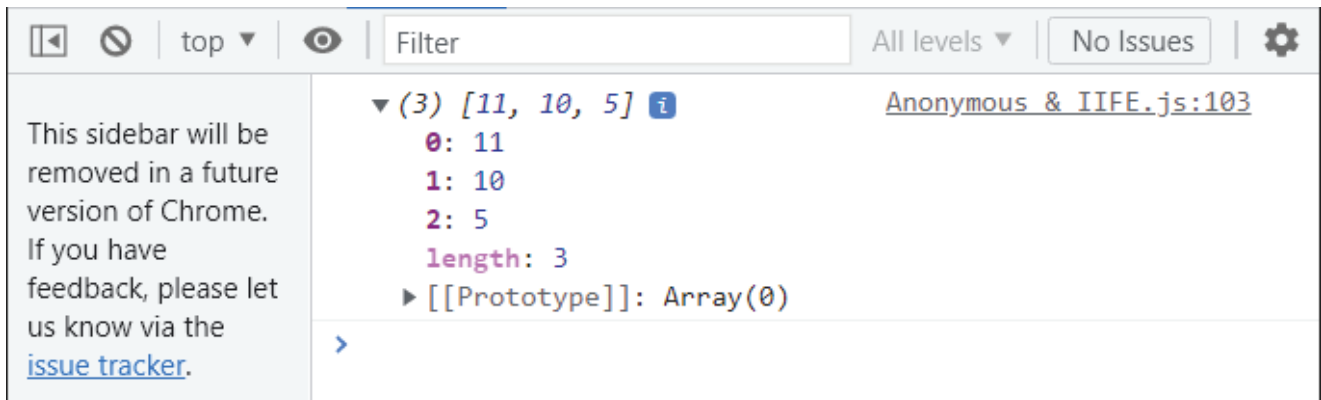
MERN STACK DEVELOPMENT

Functions – Task 5

f. Remove duplicates from an array

```
//Remove duplicates from an array using Anonymous Function
let numArray = [11, 10, 11, 5, 5];
let removeDuplicates = function(num){
  return num.filter((value, index) => num.indexOf(value) === index);
}
console.log(removeDuplicates(numArray));
```

```
//Remove duplicates from an array using IIFE
(function(num){
  let result = num.filter((value, index) => num.indexOf(value) === index);
  console.log(result);
})(numArray);
```



MERN STACK DEVELOPMENT

Functions – Task 5

g. Rotate an array by k times

// Rotate an array by k times using Anonymous Function

```
const rotateArray = function(num, k){  
  for(let i = 0; i < k; i++){  
    num.unshift(num.pop());  
  }  
  return num;  
}  
console.log(rotateArray(numbers, 3));
```

//Rotate an array by k times using IIFE

```
(function(num, k){  
  for(let i = 0; i < k; i++){  
    num.unshift(num.pop());  
  }  
  console.log(num);  
})(numbers, 4);
```

The screenshot shows the Chrome DevTools console. On the left, a sidebar contains a message about its future removal and a list of messages: 1 message, 1 user message, No errors, No warnings, 1 info, and No verbose. The main console area displays the output of the IIFE function: `(11) [9, 10, 11, 1, 2, 3, 4, 5, 6, 7, 8]`. Below this, the array elements are listed with their indices: `0: 9, 1: 10, 2: 11, 3: 1, 4: 2, 5: 3, 6: 4, 7: 5, 8: 6, 9: 7, 10: 8`. The `length` property is `11`, and the `[[Prototype]]` is `Array(0)`. The file path `Anonymous & IIFE.js:88` is visible at the top right of the console area.