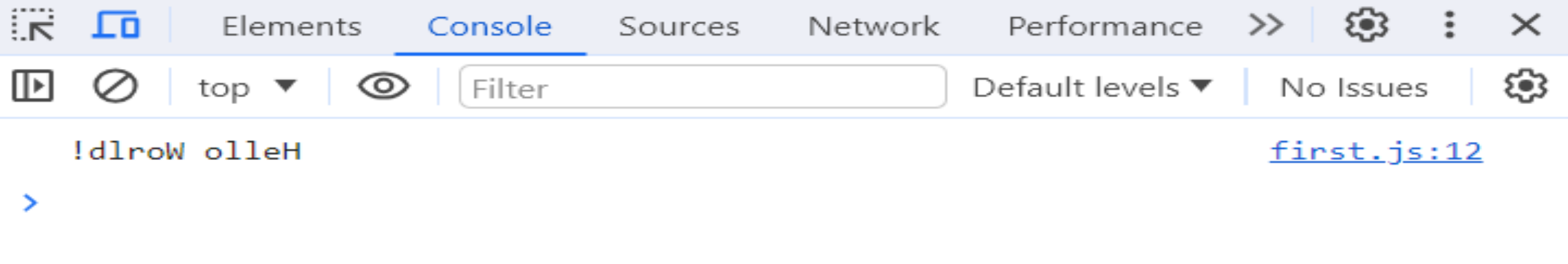







1. Question: Reverse a string without using the built-in reverse() method.

```
function reverseString(str) {  
  let reversedString = '';  
  for (let i = str.length - 1; i >= 0; i--) {  
    reversedString += str[i];  
  }  
  return reversedString;  
}  
let originalString = 'Jalal Altaf';  
let reversed = reverseString(originalString);  
console.log(reversed);
```



2. Question: Count the number of vowels in a given string.

```
function countVowels(str) {  
  const vowels = "aeiouAEIOU";  
  let count = 0;  
  for (let char of str) {  
    if (vowels.includes(char)) {  
      count++;  
    }  
  }  
  return count;  
}  
  
const inputString = "Jalal Altaf";  
const vowelCount = countVowels(inputString);  
console.log(`Number of vowels: ${vowelCount}`);
```

  Elements Console Sources Network Performance Memory >>   

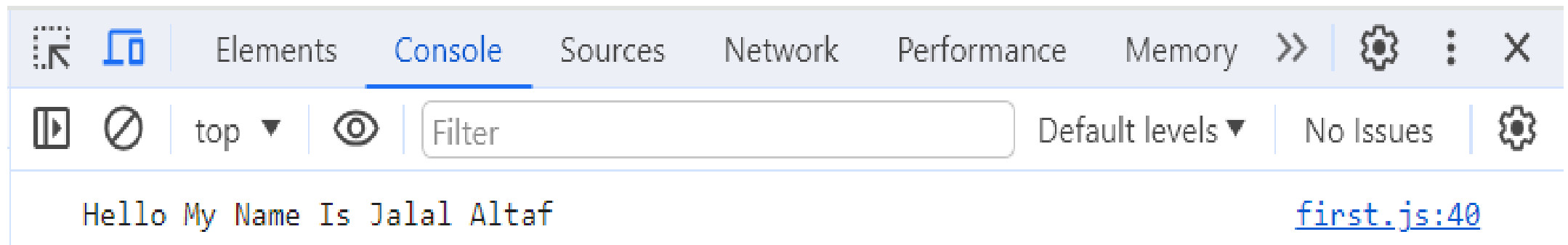
  top ▼  Filter Default levels ▼ No Issues 

Number of vowels: 4

[first.js:26](#)

3. Question: Convert the first letter of each word in a sentence to uppercase.

```
function capitalfirstWords(sentence) {  
  var words = sentence.split(' ');  
  var capitalizedWords = words.map(function(word) {  
    return word.charAt(0).toUpperCase() + word.slice(1).toLowerCase();  
  });  
  var capitalizedSentence = capitalizedWords.join(' ');  
  return capitalizedSentence;  
}  
var sentence = "hello my name is jalal altaf";  
var result = capitalfirstWords(sentence);  
console.log(result);
```



4. Question: Check if a string is a palindrome.

```
function isPalindrome(str) {  
    const cleanStr = str.replace(/[^\a-zA-Z0-9]/g, '').toLowerCase();  
    const length = cleanStr.length;  
    for (let i = 0; i < Math.floor(length / 2); i++) {  
        if (cleanStr[i] !== cleanStr[length - 1 - i]) {  
            return false;  
        }  
    }  
    return true;  
}  
  
const exampleString = "Level";  
console.log(isPalindrome(exampleString));
```

Elements Console Sources Network Performance Memory >> ⚙️ ⋮ ✕

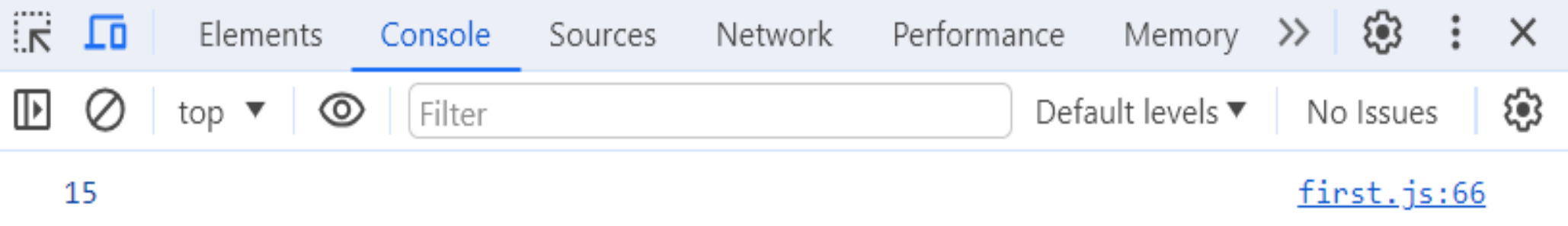
🔍 🚫 top ▼ 🔍 Filter Default levels ▼ No Issues ⚙️

true

[first.js:52](#)

5. Question: Find the sum of all positive numbers in an array.

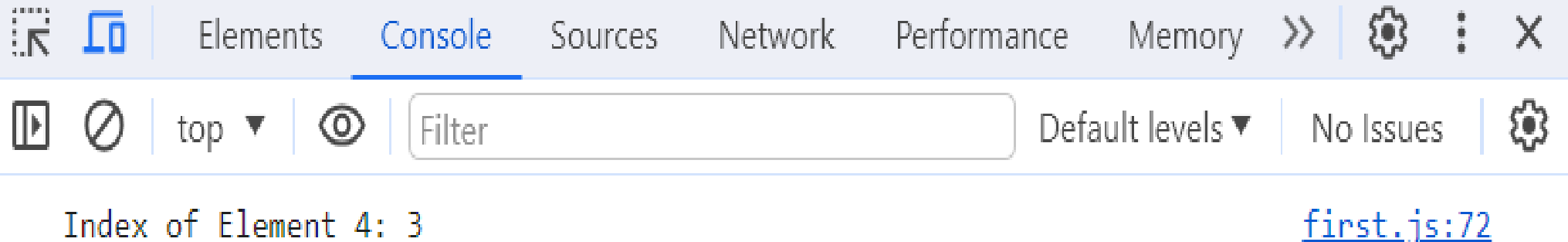
```
function sumOfPositiveNumbers(arr) {  
    let sum = 0;  
    for (let i = 0; i < arr.length; i++) {  
        if (arr[i] > 0) {  
            sum += arr[i];  
        }  
    }  
    return sum;  
}  
  
const Arr1 = [1, 2, 3, 4, 5];  
const result = sumOfPositiveNumbers(Arr1);  
console.log(result);
```



15 [first.js:66](#)

6. Question: Find the index of the first occurrence of a specific element in an array.

```
const Arr1 = [1,2,3,4,5];
const indexOfElement = Arr1.indexOf(4);
if(indexOfElement !== -1 ){
  console.log(`Index of Element 4: ${indexOfElement}`);
}else{
  console.log('Element not found in the array. ');
}
```



7. Question: Remove all duplicates from an array without built-in methods.

```
function removeDuplicates(arr) {  
  var uniqueArray = [];  
  for (var i = 0; i < arr.length; i++) {  
    if (uniqueArray.indexOf(arr[i]) === -1) {  
      uniqueArray.push(arr[i]);  
    }  
  }  
  return uniqueArray;  
}  
  
var originalArray = [1, 2, 3, 4, 2, 3, 5];  
var newArrayWithoutDuplicates = removeDuplicates(originalArray);  
console.log(newArrayWithoutDuplicates);
```

▼ (5) [1, 2, 3, 4, 5] i

- 0: 1
- 1: 2
- 2: 3
- 3: 4
- 4: 5
- length: 5

8. Question: Sort the array in ascending and descending without built-in methods.

```
// Bubble Sort for ascending order
function bubbleSortAscending(arr) {
  let n = arr.length;
  for (let i = 0; i < n - 1; i++) {
    for (let j = 0; j < n - i - 1; j++) {
      if (arr[j] > arr[j + 1]) {
        // swap arr[j] and arr[j + 1]
        let temp = arr[j];
        arr[j] = arr[j + 1];
        arr[j + 1] = temp;
      }
    }
  }
  return arr;
}

// Bubble Sort for descending order
function bubbleSortDescending(arr) {
  let n = arr.length;
  for (let i = 0; i < n - 1; i++) {
    for (let j = 0; j < n - i - 1; j++) {
      if (arr[j] < arr[j + 1]) {
        // swap arr[j] and arr[j + 1]
        let temp = arr[j];
        arr[j] = arr[j + 1];
        arr[j + 1] = temp;
      }
    }
  }
  return arr;
}

let myArray = [64, 34, 25, 12, 22, 11, 90];

console.log("Original Array:", myArray);

let ascendingArray = bubbleSortAscending([...myArray]);
console.log("Ascending Order:", ascendingArray);

let descendingArray = bubbleSortDescending([...myArray]);
console.log("Descending Order:", descendingArray);
```

Original Array: Array(7) [first.js:125](#)

- 0: 64
- 1: 34
- 2: 25
- 3: 12
- 4: 22
- 5: 11
- 6: 90
- length: 7

Ascending Order: Array(7) [first.js:128](#)

- 0: 11
- 1: 12
- 2: 22
- 3: 25
- 4: 34
- 5: 64
- 6: 90
- length: 7

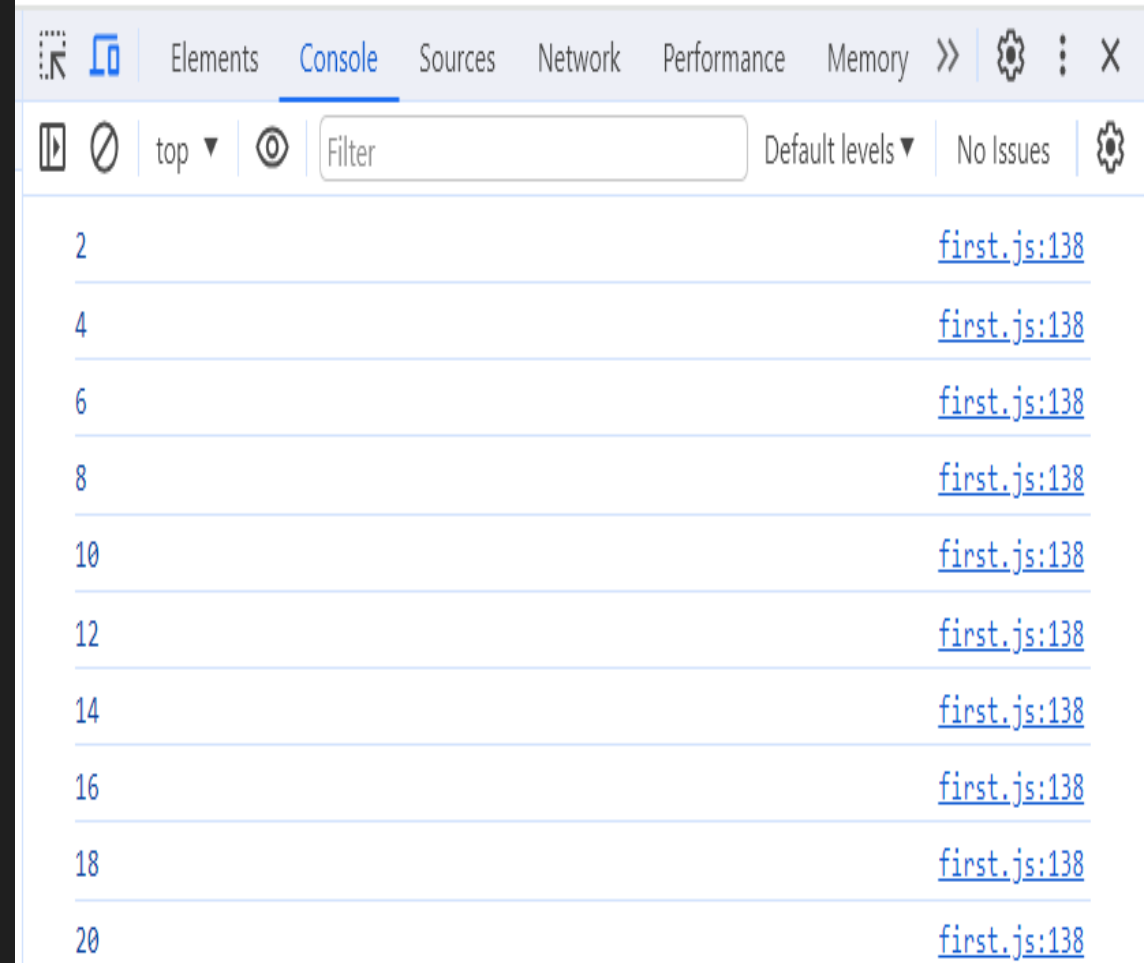
Descending Order: Array(7) [first.js:131](#)

- 0: 90
- 1: 64
- 2: 34
- 3: 25
- 4: 22
- 5: 12
- 6: 11
- length: 7

9. Question: Print all even numbers between 1 and 20 using a while loop.

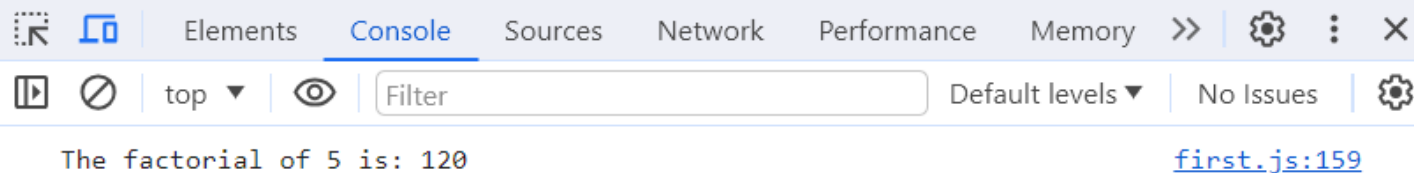
```
let number = 1;

while (number <= 20) {
  if (number % 2 === 0) {
    console.log(number);
  }
  number++;
}
```



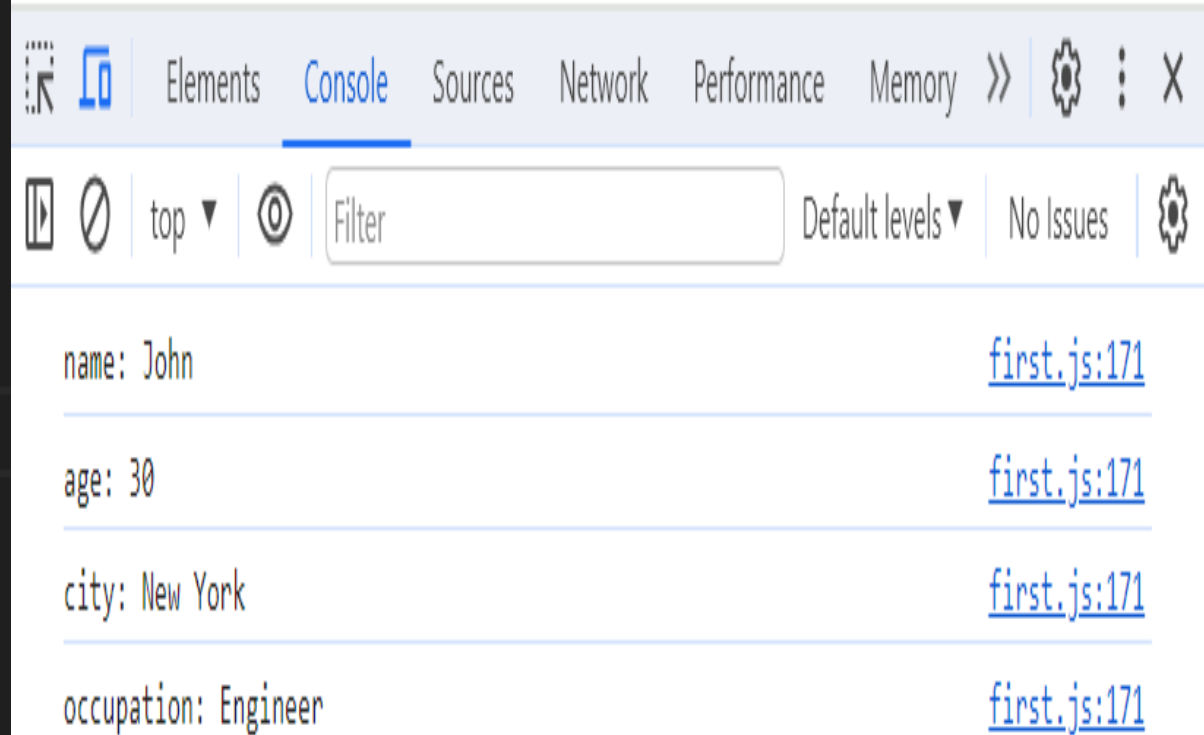
10. Question: Calculate the factorial of a number using a do-while loop.

```
function calculateFactorial(number) {  
  if (number < 0) {  
    return "Please Enter Only Positive Integer:";  
  }  
  let factorial = 1;  
  let i = 1;  
  do {  
    factorial *= i;  
    i++;  
  } while (i <= number);  
  return factorial;  
}  
  
const num = 5;  
const result = calculateFactorial(num);  
  
console.log(`The factorial of ${num} is: ${result}`);
```



11. Question: Iterate through the properties of an object using a for-in loop.

```
let myObject = {  
  name: "John",  
  age: 30,  
  city: "New York",  
  occupation: "Engineer"  
};  
  
// Iterate through the properties using a for-in loop  
for (let key in myObject) {  
  if (myObject.hasOwnProperty(key)) {  
    console.log(`${key}: ${myObject[key]}`);  
  }  
}
```

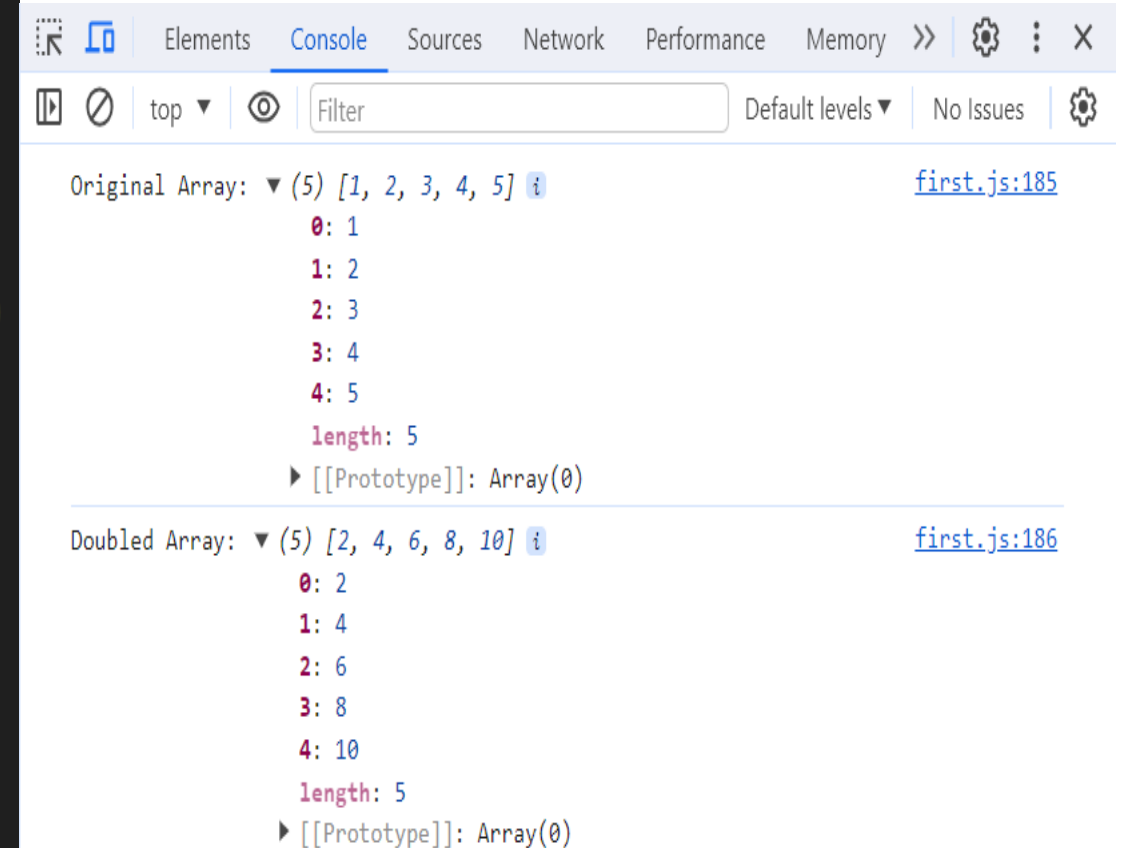


12. Question: Loop through an array using a for-of loop and double each element.

```
let myArray = [1, 2, 3, 4, 5];
let doubledArray = [];

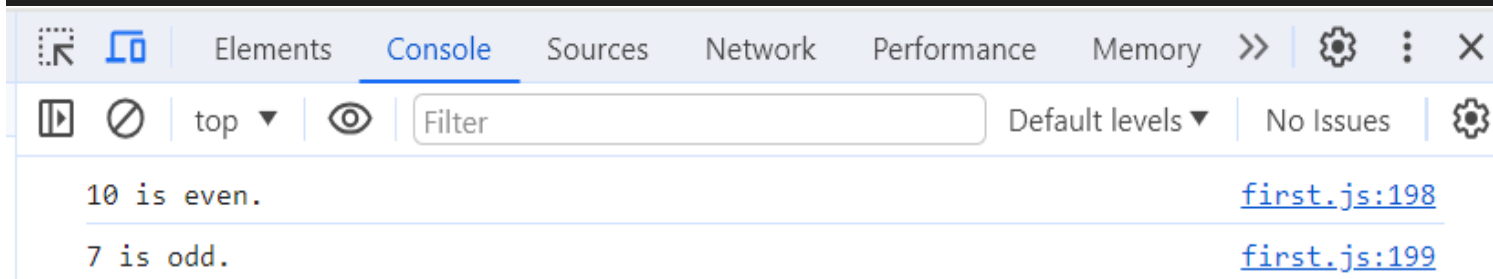
// Iterate through the array using a for-of loop
for (let element of myArray) {
  let doubledElement = element * 2;
  doubledArray.push(doubledElement);
}

console.log("Original Array:", myArray);
console.log("Doubled Array:", doubledArray);
```



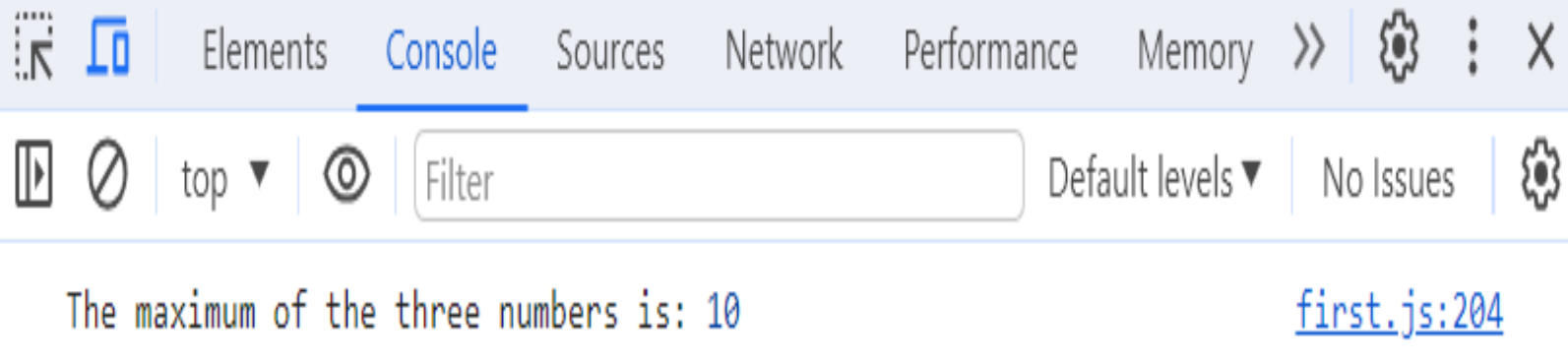
13. Question: Check if a number is even or odd and return a corresponding message.

```
function checkEvenOrOdd(number) {  
  if (number % 2 === 0) {  
    return `${number} is even.`;  
  } else {  
    return `${number} is odd.`;  
  }  
}  
  
console.log(checkEvenOrOdd(10));  
console.log(checkEvenOrOdd(7));
```



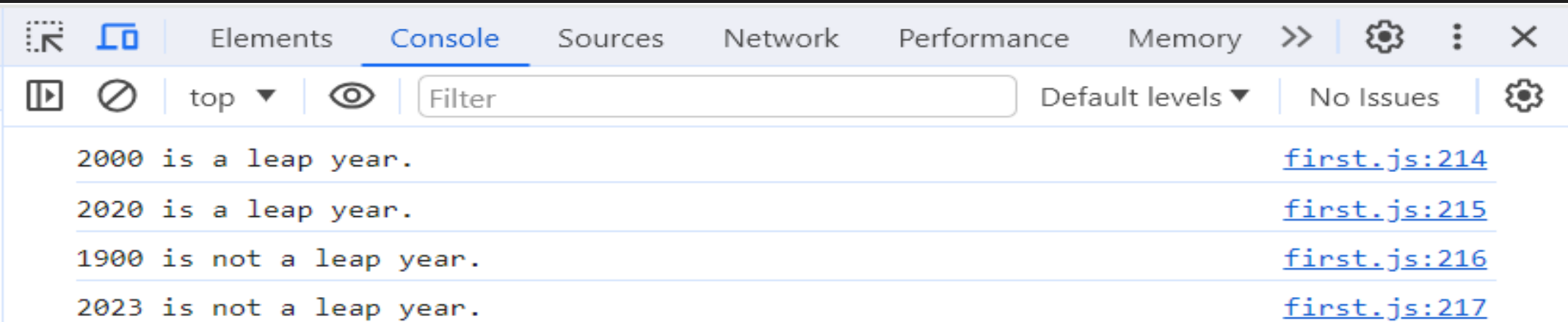
14. Question: Find the maximum of three numbers using nested ternary operators.

```
function findMaxOfThreeNumbers(a, b, c) {  
  return (a > b) ? ((a > c) ? a : c) : ((b > c) ? b : c);  
}  
  
const result = findMaxOfThreeNumbers(10, 5, 8);  
console.log("The maximum of the three numbers is:", result);
```



15. Question: Determine if a year is a leap year or not.

```
function isLeapYear(year) {  
  if ((year % 4 === 0 && year % 100 !== 0) || (year % 400 === 0)) {  
    return `${year} is a leap year.`;  
  } else {  
    return `${year} is not a leap year.`;  
  }  
}  
  
console.log(isLeapYear(2000));  
console.log(isLeapYear(2020));  
console.log(isLeapYear(1900));  
console.log(isLeapYear(2023));
```



The screenshot shows a web browser's developer console with the 'Console' tab selected. The console displays four log messages from the `isLeapYear` function, confirming its logic for leap years. The messages are: '2000 is a leap year.', '2020 is a leap year.', '1900 is not a leap year.', and '2023 is not a leap year.'. Each message is followed by a link to the source file and line number.

Message	Source
2000 is a leap year.	first.js:214
2020 is a leap year.	first.js:215
1900 is not a leap year.	first.js:216
2023 is not a leap year.	first.js:217