

Assignment – 2

Write a JavaScript program to find all the index positions of a given word within a given string

```
1. function findAllIndexesOfWord(sentence, word) {
2.     let indexes = [];
3.     let index = sentence.indexOf(word);
4.
5.     while (index !== -1) {
6.         indexes.push(index);
7.         index = sentence.indexOf(word, index + 1);
8.     }
9.
10.    return indexes;
11. }
12.
13.    const sentence = "Lorem, ipsum dolor sit amet
    consectetur adipisicing elit. Molestias aut, Lorem
    suscipit eaque?";
14.    const word = "Lorem";
15.
16.    const indexes = findAllIndexesOfWord(sentence,
    word);
17.    console.log(`Indexes of "${word}" in the sentence:
    ${indexes}`);
18.
O/P: Indexes of "Lorem" in the sentence: 0,74
```

2. Write a JavaScript program to find the first index of a given element in an array using the linear search algorithm.

```
function indexOF(arr,element){
    for(let x=0;x<arr.length;x++){
        if(arr[x]==element){return x;}
    }
}

const index=indexOF([1,3,5,6,5,4,3,5,9],3);
console.log(`index of the given element is ${index}`);
```

O/P:index of the given element is 2

3. Write a JavaScript program to sort a list of elements using Quick sort

```
function quickSort(arr) {
  if (arr.length <= 1) {
    return arr;
  }

  const pivot = arr[Math.floor(arr.length / 2)];
  const left = [];
  const right = [];

  for (let i = 0; i < arr.length; i++) {
    if (i === Math.floor(arr.length / 2)) {
      continue;
    }
    if (arr[i] < pivot) {
      left.push(arr[i]);
    } else {
      right.push(arr[i]);
    }
  }

  return [...quickSort(left), pivot, ...quickSort(right)];
}

const arr= [5, 3, 7, 2, 9, 1, 6, 4, 8];
const sortedArray = quickSort(arr);
console.log("Sorted array:", sortedArray);
O/P: Sorted array: [1, 2, 3, 4, 5,6, 7, 8, 9]
```

4. Write a JavaScript program to sort a list of elements using Merge sort.

```
function mergeSort(arr) {
  if (arr.length <= 1) {
    return arr;
  }

  const mid = Math.floor(arr.length / 2);
```

```

    const left = arr.slice(0, mid);
    const right = arr.slice(mid);

    return merge(mergeSort(left), mergeSort(right));
}

function merge(left, right) {
    let result = [];
    let leftIndex = 0;
    let rightIndex = 0;

    while (leftIndex < left.length && rightIndex <
right.length) {
        if (left[leftIndex] < right[rightIndex]) {
            result.push(left[leftIndex]);
            leftIndex++;
        } else {
            result.push(right[rightIndex]);
            rightIndex++;
        }
    }

    return
result.concat(left.slice(leftIndex)).concat(right.slice(rightIndex));
}

// Example usage:
const unsortedArray = [5, 3, 7, 2, 9, 1, 6, 4, 8];
const sortedArray = mergeSort(unsortedArray);
console.log("Sorted array using Merge Sort:", sortedArray);
O/P: Sorted array: [1, 2, 3, 4, 5,6, 7, 8, 9]

```

5. Write a JavaScript program to sort a list of elements using Heap sort.

```

function heapSort(arr) {
    buildMaxHeap(arr);

    for (let i = arr.length - 1; i > 0; i--) {

```

```

        swap(arr, 0, i);
        heapify(arr, 0, i);
    }

    return arr;
}

function buildMaxHeap(arr) {
    const mid = Math.floor(arr.length / 2);
    for (let i = mid; i >= 0; i--) {
        heapify(arr, i, arr.length);
    }
}

function heapify(arr, i, max) {
    let largest = i;
    const left = 2 * i + 1;
    const right = 2 * i + 2;

    if (left < max && arr[left] > arr[largest]) {
        largest = left;
    }

    if (right < max && arr[right] > arr[largest]) {
        largest = right;
    }

    if (largest !== i) {
        swap(arr, i, largest);
        heapify(arr, largest, max);
    }
}

function swap(arr, i, j) {
    const temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
}

```

```
const unsortedArray = [5, 3, 7, 2, 9, 1, 6, 4, 8];
const sortedArray = heapSort(unsortedArray);
console.log("Sorted array using Heap Sort:", sortedArray);
O/P: Sorted array: [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

6. Write a JavaScript program to sort a list of elements using Insertion sort.

```
function insertionSort(arr) {
    for (let i = 1; i < arr.length; i++) {
        let current = arr[i];
        let j = i - 1;

        while (j >= 0 && arr[j] > current) {
            arr[j + 1] = arr[j];
            j--;
        }

        arr[j + 1] = current;
    }

    return arr;
}

const unsortedArray = [5, 3, 7, 2, 9, 1, 6, 4, 8];
const sortedArray = insertionSort(unsortedArray);
console.log("Sorted array :", sortedArray);
O/P: Sorted array: [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

7. Write a JavaScript program to sort a list of elements using Bubble sort.

```
function bubbleSort(arr) {
    const len = arr.length;
    let swapped;

    do {
        swapped = false;
        for (let i = 0; i < len - 1; i++) {
            if (arr[i] > arr[i + 1]) {
                const temp = arr[i];
```

```

        arr[i] = arr[i + 1];
        arr[i + 1] = temp;
        swapped = true;
    }
}
} while (swapped);

return arr;
}
const unsortedArray = [5, 3, 7, 2, 9, 1, 6, 4, 8];
const sortedArray = bubbleSort(unsortedArray);
console.log("Sorted array using Bubble Sort:", sortedArray);
O/P: Sorted array: [1, 2, 3, 4, 5, 6, 7, 8, 9]

```

8. Write a JavaScript program to sort the characters in a string alphabetically.

```

function sortStringAlphabetically(str) {
    return str.split('').sort().join('');
}

const inputString = "hello world";
const sortedString = sortStringAlphabetically(inputString);
console.log("Sorted characters in the string:",
sortedString);
O/P: Sorted characters in the string:  dehlloorw

```

9. Write a JavaScript program to check if a numeric array is sorted or not.

```

function isArraySorted(arr) {
    for (let i = 1; i < arr.length; i++) {
        if (arr[i] < arr[i - 1]) {
            return false;
        }
    }
    return true;
}

const sortedArray = [1, 2, 3, 4, 5];

```

```
console.log(`${sortedArray ? "array is sorted": "array is not sorted"}`);
```

O/P: array is sorted

10. Write a JavaScript function to validate whether a given value type is null or not.

```
function isNull(value) {  
    return value === null;  
}  
const value1 = null;  
const value2 = 42;  
console.log("Is value1 null?", isNull(value1));  
console.log("Is value2 null?", isNull(value2));  
O/P: Is value1 null? true  
    Is value2 null? false
```

11. Write a JavaScript function to validate whether a given value is a number or not.

```
function isNumber(value) {  
    return typeof(value) === 'number' && !isNaN(value);  
}  
const value1 = 42;  
const value2 = "hello";  
console.log("Is value1 a number?", isNumber(value1));  
console.log("Is value2 a number?", isNumber(value2));  
O/P: Is value1 a number? true  
    Is value2 a number? false
```

12. Write a JavaScript function to validate whether a given value is RegExp or not.

```
function isRegExp(value) {  
    return value instanceof RegExp;  
}  
const value1 = /test/;  
const value2 = "hello";  
console.log("Is value1 a RegExp?", isRegExp(value1));  
console.log("Is value2 a RegExp?", isRegExp(value2));  
O/P: Is value1 a RegExp? true  
    Is value2 a RegExp? false
```

13. Write a JavaScript program to delete the rollno property from the following object. Also print the object before or after deleting the property. Sample object: var student = { name : "David Rayy", sclass : "VI", rollno : 12 };

```
var student = {
  name: "David Rayy",
  sclass: "VI",
  rollno: 12
};
console.log("Before deleting rollno property:", student);
delete student.rollno;
console.log("After deleting rollno property:", student);
O/P: Before deleting rollno property: { name: 'David Rayy',
sclass: 'VI', rollno: 12 }

After deleting rollno property: { name: 'David Rayy',
sclass: 'VI' }
```

14. Write a JavaScript program to display the reading status (i.e. display book name, author name and reading status) of the following books. var library = [{ author: 'Bill Gates', title: 'The Road Ahead', readingStatus: true }, { author: 'Steve Jobs', title: 'Walter Isaacson', readingStatus: true }, { author: 'Suzanne Collins', title: 'Mockingjay: The Final Book of The Hunger Games', readingStatus: false }];

```
var library = [
  {
    author: 'Bill Gates',
    title: 'The Road Ahead',
    readingStatus: true
  },
  {
    author: 'Steve Jobs',
    title: 'Walter Isaacson',
    readingStatus: true
  },
  {
    author: 'Suzanne Collins',
    title: 'Mockingjay: The Final Book of The Hunger
Games',
    readingStatus: false
  }
];
```



```

    }
  ];
  library.forEach(book => {
    console.log(`${book.title} by ${book.author} - Reading
status: ${book.readingStatus ? 'Read' : 'Not read'}`);
  });

```

O/P: "The Road Ahead" by Bill Gates - Reading status: Read
"Walter Isaacson" by Steve Jobs - Reading status: Read
"Mockingjay: The Final Book of The Hunger Games" by Suzanne
Collins - Reading status: Not read

15. Write a JavaScript program to create a clock. Note: The output will come every second. Expected Console Output : "14:37:42" "14:37:43" "14:37:44" "14:37:45" "14:37:46" "14:37:47"

```

function updateClock() {
  const now = new Date();
  console.log(`${(now.getHours()).toFixed(0)}:${now.getMin
utes()}:${now.getSeconds()}`);
}
setInterval(updateClock, 1000);

```

O/P:
17:10:11
17:10:12
17:10:13
17:10:14
17:10:15
17:10:17

16. Write a JavaScript function to parse an URL.

```

const urlString = "https://www.google.com/index";
const url = new URL(urlString);
const hostname = url.hostname;
const pathname = url.pathname;
console.log(`Hostname: ${hostname}`);
console.log(`Pathname: ${pathname}`);

```

O/P: Hostname: www.google.com

Pathname: /index

17. Write a JavaScript function to split a string and convert it into an array of words.

```
function splitStringIntoWords(str) {  
    return str.split(" ").filter(word => word !== '');  
}  
const sentence = "Splitting the sentence."  
console.log(splitStringIntoWords(sentence));  
O/P: [ 'Splitting', 'the', 'sentence.' ]
```

18. Write a JavaScript function that takes a string with both lowercase and upper case letters as a parameter. It converts upper case letters to lower case, and lower case letters to upper case.

```
function convertCase(str) {  
    return str.replace(/[a-z]/ig, function(match) {  
        return match === match.toLowerCase() ?  
match.toUpperCase() : match.toLowerCase();  
    });  
}  
const upperCase= "uppercase";  
const lowerCase="LOWERCASE";  
console.log("Converted string:", convertCase(upperCase));  
console.log("Converted string:", convertCase(lowerCase));  
O/P:  
Converted string: UPPERCASE  
Converted string: lowercase
```

19. Write a JavaScript function that returns the number of minutes in hours and minutes. Input : console.log(timeConvert(200)); Output : "200 minutes = 3 hour(s) and 20 minute(s)."

```
function timeConvert(minutes) {  
    const hours = Math.floor(minutes / 60);  
    const remainingMinutes = minutes % 60;  
    return `${minutes} minutes = ${hours} hour(s) and  
${remainingMinutes} minute(s).`;  
}  
console.log(timeConvert(200));  
O/P: 200 minutes = 3 hour(s) and 20 minute(s).
```

20. Write a JavaScript program to implement a stack that checks if a given element is present or not in the stack.

```
. class Stack {  
    constructor() {  
        this.items = [];  
    }  
  
    push(element) {  
        this.items.push(element);  
    }  
  
    pop() {  
        if (this.isEmpty()) {  
            return "Underflow";  
        }  
        return this.items.pop();  
    }  
  
    peek() {  
        return this.items[this.items.length - 1];  
    }  
    isEmpty() {  
        return this.items.length === 0;  
    }  
    contains(element) {  
        return this.items.includes(element);  
    }  
}  
const stack = new Stack();  
stack.push(1);  
stack.push(2);  
stack.push(3);  
console.log("Is 2 present in the stack?",  
stack.contains(2));  
console.log("Is 4 present in the stack?",  
stack.contains(4));  
O/P: Is 2 present in the stack? true
```

Is 4 present in the stack? false

21. Write a JavaScript program to check whether a single linked list is empty or not. Return true otherwise false.

```
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }
}
class LinkedList {
  constructor() {
    this.head = null;
  }

  isEmpty() {
    return this.head === null;
  }
}

const list = new LinkedList();
console.log("Is the list empty?", list.isEmpty());
O/P: Is the list empty? true
```

22. Write a JavaScript program to create a class called 'Rectangle' with properties for width and height. Include two methods to calculate rectangle area and perimeter. Create an instance of the 'Rectangle' class and calculate its area and perimeter.

```
class Rectangle {
  constructor(width, height) {
    this.width = width;
    this.height = height;
  }
  area() {
    return this.width * this.height;
  }
  perimeter() {
    return 2 * (this.width + this.height);
  }
}
```

```

    }
}
const rectangle = new Rectangle(5, 3);
console.log("Area:", rectangle.area());
console.log("Perimeter:", rectangle.perimeter());
O/P: Area: 15
Perimeter: 16

```

23. Write a JavaScript program to create a slideshow that changes the displayed image when a next or previous button is clicked.

```

<script>
let slideIndex = 1;
showSlides(slideIndex);

function plusSlides(n) {
    showSlides(slideIndex += n);
}

function showSlides(n) {
    let i;
    const slides =
document.getElementsByClassName("mySlides");
    if (n > slides.length) { slideIndex = 1 }
    if (n < 1) { slideIndex = slides.length }
    for (i = 0; i < slides.length; i++) {
        slides[i].style.display = "none";
    }
    slides[slideIndex - 1].style.display = "block";
}
</script>
O/P:

```

24. Write a JavaScript program that uses a try-catch block to catch and handle a 'SyntaxError' when parsing an invalid JSON string.

```

const jsonString = '{"name": "John", "age": 30, "city": "New
York"}';

```

```
try {
    const jsonData = JSON.parse(jsonString);
    console.log("Parsed JSON data:", jsonData);
} catch (error) {
    if (error instanceof SyntaxError) {
        console.error("Invalid JSON:", error.message);
    } else {
        throw error;
    }
}
O/P: Invalid JSON: Unexpected end of JSON input
```

25. Write a JavaScript program to redirect to a specified URL.

```
function redirectTo(url) {
    window.location.href = url;
}
redirectTo("https://www.google.com");
```