

Assignment-2

Q.1 Write a JavaScript program to find all the index positions of a given word within a given string.

```
function findAllIndexes(str, word) {
    let indexes = [];
    let index = -1;

    while ((index = str.indexOf(word, index + 1)) !== -1) {
        indexes.push(index);
    }

    return indexes;
}

let string = "hello world hello";
let word = "world";
let positions = findAllIndexes(string, word);
console.log("The word '" + word + "' appears at positions: " + positions.join(", "));
```

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

```
function linearSearch(arr, element) {
    for (let i = 0; i < arr.length; i++) {
        if (arr[i] === element) {
            return i;
        }
    }
    return -1;
}
```

Q.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)];
  }
}

```

```

// Example usage:
const arr = [5, 3, 7, 2, 8, 4, 1];
const sortedArr = quickSort(arr);
console.log("Sorted array:", sortedArr);

```

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

```

function merge_Arrays(left_sub_array, right_sub_array) {
  let array = []
  while (left_sub_array.length && right_sub_array.length) {
    if (left_sub_array[0] < right_sub_array[0]) {
      array.push(left_sub_array.shift())
    } else {
      array.push(right_sub_array.shift())
    }
  }
  return [ ...array, ...left_sub_array, ...right_sub_array ]
}

function merge_sort(unsorted_Array) {
  const middle_index = unsorted_Array.length / 2
  if(unsorted_Array.length < 2) {
    return unsorted_Array
  }
  const left_sub_array = unsorted_Array.splice(0, middle_index)
  return merge_Arrays(merge_sort(left_sub_array),merge_sort(unsorted_Array))
}

unsorted_Array = [39, 28, 44, 4, 10, 83, 11];
console.log("The sorted array will be: ",merge_sort(unsorted_Array));

```

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

```
function customSort(arr) {  
    var N = arr.length;  
  
    for (var i = Math.floor(N / 2) - 1; i >= 0; i--)  
        customHeapify(arr, N, i);  
  
    for (var i = N - 1; i > 0; i--) {  
        var temp = arr[0];  
        arr[0] = arr[i];  
        arr[i] = temp;  
        customHeapify(arr, i, 0);  
    }  
}
```

```
function customHeapify(arr, N, i) {  
    var largest = i;  
    var l = 2 * i + 1;  
    var r = 2 * i + 2;  
  
    if (l < N && arr[l] > arr[largest])  
        largest = l;  
  
    if (r < N && arr[r] > arr[largest])  
        largest = r;  
  
    if (largest != i) {  
        var swap = arr[i];  
        arr[i] = arr[largest];  
        arr[largest] = swap;  
        customHeapify(arr, N, largest);  
    }  
}
```

```
function customPrintArray(arr) {  
    var N = arr.length;  
    for (var i = 0; i < N; ++i)  
        console.log(arr[i]);  
}
```

```
var originalArr = [12, 11, 13, 5, 6, 7];  
customSort(originalArr);  
console.log("Sorted array:" + originalArr);
```