Array Logical Questions in Javascript Solution

Q-1.

Solution :

```
<script>
function calSumUtil(a, b, n, m)
{
	// array to store sum.
	let sum = new Array(n);
	let i = n - 1, j = m - 1, k = n - 1;
	let carry = 0, s = 0;
	// Until we reach beginning of array. we are comparing only for second array
	// because we have already compare the size of array in wrapper function.
	while (j >= 0) {
		// find sum of corresponding element of both arrays.
		s = a[i] + b[j] + carry;
		sum[k] = (s % 10);
		// Finding carry for next sum.
		carry = Math.floor(s / 10);
		k--;
		i--;
		j--;
	}
	// If second array size is less the first array size.
	while (i >= 0) {
		// Add carry to first array elements.
		s = a[i] + carry;
		sum[k] = (s % 10);
```

```
            carry = Math.floor(s / 10);
            i--;
            k--;
        }
        let ans = 0;
        // If there is carry on adding 0 index elements. append 1 to total sum.
        if (carry){
            ans = 10;
        }
        // Converting array into number.
        for (let i = 0; i <= n - 1; i++) {
            ans += sum[i];
            ans *= 10;
        }
        return ans / 10;
}
// Wrapper Function
function calSum(a, b, n, m)
{
        // Making first array which have greater number of element
        if (n >= m)
            return calSumUtil(a, b, n, m);
        else
            return calSumUtil(b, a, m, n);
}
// Driven Program
        let a = [ 9, 3, 9 ];
        let b = [ 6, 1 ];
```

```
	let n = a.length;
	let m = b.length;
	document.write(calSum(a, b, n, m) + "<br>");
</script>
```

Q-2.

Solution :

```
<script>
	function largest(arr) {
		let i;
		// Initialize maximum element
		let max = arr[0];
		// Traverse array elements from second and compare every element with current max
		for (i = 1; i < arr.length; i++) {
			if (arr[i] > max)
				max = arr[i];
		}
	return max;
}
	// Driver code
	let arr = [10, 324, 45, 90, 9808];
	document.write("Largest in given array is " + largest(arr));
</script>
```

Q-3.

Solution :

```
<script>
	function sum(arr) {
		let sum = 0; // initialize sum Iterate through all elements and add them to sum
		for (let i = 0; i < arr.length; i++)
			sum += arr[i];
		return sum;
	}
	// Driver code
	let arr = [12, 3, 4, 15];
	document.write("Sum of given array is " + sum(arr));
</script>
```

Q-4.

Solution :

```
<script>
function sortArrays(arr)
{
	// Finding the length of array 'arr'
		let length = arr.length;
		// Sorting using a single loop
		for (let j = 0; j < length - 1; j++) {
			// Checking the condition for two simultaneous elements of the array
			if (arr[j] > arr[j + 1]) {
				// Swapping the elements.
				let temp = arr[j];
				arr[j] = arr[j + 1];
```

```
                                arr[j + 1] = temp;
                                // updating the value of j = -1 so after getting updated for j++
                                // in the loop it becomes 0 and the loop begins from the start.
                                j = -1;
                        }
                }
                return arr;
}
// Declaring main method
let arr=[1, 2, 99, 9, 8,7, 6, 0, 5, 4, 3];
document.write("Original array: ["+ (arr).join(", ")+"]<br>");
// Sorting the array using a single loop
arr = sortArrays(arr);
// Printing the sorted array.
document.write("Sorted array: ["+ arr.join(", ")+"]<br>");
</script>
```

Q-5.

Solution :

```
<script>
function findDuplicates(arr, len) {
        // initialize ifPresent as false
        let ifPresent = false;      // ArrayList to store the output
        let al = new Array();
        for (let i = 0; i < len - 1; i++) {
                for (let j = i + 1; j < len; j++) {
                        if (arr[i] == arr[j]) {
                        // checking if element is present in the ArrayList or not if present then break
```

```
if (al.includes(arr[i])) {
                        break;
                        }
        // if element is not present in the ArrayList then add it to ArrayList
        // and make ifPresent at true
                                else {
                                        al.push(arr[i]);
                                        ifPresent = true;
                                }
                        }
                }
        }
        // if duplicates is present then print ArrayList
        if (ifPresent == true) {
                document.write(`[${al}]`);
        }
        else {
                document.write("No duplicates present in arrays");
        }
}
// Driver Code
let arr = [12, 11, 40, 12, 5, 6, 5, 12, 11];
let n = arr.length;
findDuplicates(arr, n);
</script>
```

Q-6.

Solution :

```
<script>
	function minMax(arr)
	{
		// Initialize the min_value and max_value to 0
		let min_value = 0;
		let max_value = 0;
		let n = arr.length;
		// Sort array before calculating min and max value
		arr.sort(function(a,b){return a-b;});
		for (let i = 0, j = n - 1;
			i < n - 1; i++, j--)
		{
			// All elements except rightmost will be added
			min_value += arr[i];
			// All elements except leftmost will be added
			max_value += arr[j];
		}
		// Output: min_value and max_value
		document.write(
			min_value + " " + max_value+"<br>");
	}
	// Driver Code
	let arr=[10, 9, 8, 7, 6, 5];
	let arr1=[100, 200, 300, 400, 500 ];
	minMax(arr);
	minMax(arr1);
</script>
```

Q-7.

Solution :

```
<script>
	let n = 1234;	//Function to get sum of digits
	function getSum(n) {
		let sum = 0;
		while (n > 0 || sum > 9) {
			if(n == 0) {
				n = sum;
				sum = 0;
			}
			sum = sum + n % 10;
			n = Math.floor(n / 10);
		}
		return sum;
	}
//function call
	document.write(getSum(n));
</script>
```

Q-8.

Solution :

```
<script>
function reverse(arr, n, k)
{
	for(let i = 0; i < n; i += k)
	{
		let left = i;
```

```
                // To handle case when k is not multiple of n
                let right = Math.min(i + k - 1, n - 1);
                let temp;
                // Reverse the sub-array [left, right]
                while (left < right)
                {
                        temp = arr[left];
                        arr[left] = arr[right];
                        arr[right] = temp;
                        left += 1;
                        right -= 1;
                }
        }
        return arr;
}
// Driver Code
let arr = new Array(1, 2, 3, 4, 5, 6, 7, 8);
let k = 3;
let n = arr.length;
let arr1 = reverse(arr, n, k);
for(let i = 0; i < n; i++)
        document.write(arr1[i] + " ");
</script>
```

Q-9.

Solution :

```
<script>
	function print2largest(arr, arr_size) {
		let i, first, second;
		// There should be atleast two elements
		if (arr_size < 2) {
			document.write(" Invalid Input ");
			return;
		}
		// sort the array
		arr.sort();
		// start from second last element as the largest element is at last
		for (i = arr_size - 2; i >= 0; i--) {
			// if the element is not equal to largest element
			if (arr[i] != arr[arr_size - 1]) {
				document.write("The second largest element is " + arr[i]);
				return;
			}
		}
		document.write("There is no second largest element<br>");
	}
	// Driver program to test above function
	let arr= [ 12, 35, 1, 10, 34, 1 ];
	let n = arr.length;
	print2largest(arr, n);
</script>
```

Q-10.

Solution :

```
const arrayOfNumbers = [1, 2, 3, 4];

arrayOfNumbers.reduce((accumulator, currentValue, index, array) => array[index] = array[index] * 2);
```