

## Worksheet 7 :

### 1) STACKS

```
const prompt = require("prompt-sync")({ sigint: true });
let myStack = [];

function menuStack() {
    process.stdout.write("\n===== STACK MENU =====\n");
    process.stdout.write("1. Push element onto stack\n");
    process.stdout.write("2. Pop element from stack\n");
    process.stdout.write("3. Find element and its index\n");
    process.stdout.write("4. Print number of elements in stack\n");
    process.stdout.write("5. Count total integers and floats\n");
    process.stdout.write("6. Fix decimal places of floats\n");
    process.stdout.write("0. Reduce to sum (default)\n");
    process.stdout.write("Enter your choice: ");
    let choice = prompt();
    return choice;
}

function pushStack() {
    let input = prompt("Enter a number (int or float) to push: ");
    let num = parseFloat(input);
    if (!isNaN(num)) {
        myStack.push(num);
        console.log("Pushed:", num);
    } else {
        console.log("Only numeric values are allowed!");
    }
}

function popStack() {
    if (myStack.length === 0) {
        console.log("Stack is empty!");
    } else {
        let popped = myStack.pop();
        console.log("Popped:", popped);
    }
}

function findStack() {
```

```
let val = parseFloat(prompt("Enter number to find: "));
let found = myStack.find(e => e === val);
let idx = myStack.findIndex(e => e === val);
if (found !== undefined) {
    console.log(`Element ${found} found at index ${idx}`);
} else {
    console.log("Element not found!");
}

function countTypes() {
    let intCount = 0, floatCount = 0;
    myStack.forEach(num => {
        if (Number.isInteger(num)) intCount++;
        else floatCount++;
    });
    console.log(`Integers: ${intCount}, Floats: ${floatCount}`);
}

function fixDecimals() {
    let n = parseInt(prompt("Enter number of decimal places: "));
    myStack = myStack.map(num => Number.isInteger(num) ? num :
parseFloat(num.toFixed(n)));
    console.log("Stack after fixing decimals:", myStack);
}

function reduceSum() {
    let sum = myStack.reduce((acc, val) => acc + val, 0);
    console.log("Sum of stack elements:", sum);
}

// Menu loop
while (true) {
    let choice = menuStack();
    switch (choice) {
        case "1": pushStack(); break;
        case "2": popStack(); break;
        case "3": findStack(); break;
        case "4": console.log("Number of elements:", myStack.length); break;
        case "5": countTypes(); break;
        case "6": fixDecimals(); break;
        case "0": reduceSum(); break;
        default: reduceSum(); break;
    }
}
```

## 2) Queue

```
const prompt = require("prompt-sync")({ sigint: true });
let myQueue = [];

function menuQueue() {
    process.stdout.write("\n===== QUEUE MENU =====\n");
    process.stdout.write("1. Insert element at end\n");
    process.stdout.write("2. Remove element from front\n");
    process.stdout.write("3. Print total number of elements\n");
    process.stdout.write("4. Sort elements into new array\n");
    process.stdout.write("5. Remove n elements from ith position and add 2 new
elements\n");
    process.stdout.write("6. Divide queue into two queues\n");
    process.stdout.write("7. Join all elements into string\n");
    process.stdout.write("0. Convert all elements to string (map)\n");
    process.stdout.write("Enter your choice: ");
    let choice = prompt();
    return choice;
}

function insertQueue() {
    let val = prompt("Enter element to insert (int or string): ");
    if (!isNaN(val)) val = Number(val); // convert numbers to number type
    myQueue.push(val);
    console.log("Queue after insertion:", myQueue);
}

function removeQueue() {
    if (myQueue.length === 0) console.log("Queue is empty!");
    else console.log("Removed element:", myQueue.shift());
}

function printCount() {
    console.log("Total elements in queue:", myQueue.length);
}

function sortQueue() {
    let sorted = [...myQueue].sort(); // store in new array
    console.log("Sorted queue array:", sorted);
}

function spliceQueue() {
    let i = parseInt(prompt("Enter starting index i: "));
    let n = parseInt(prompt("Enter number of elements to remove: "));
    let start = i - n;
    let end = i;
    let removedElements = myQueue.splice(start, end - start);
    myQueue.push(...removedElements);
    myQueue.push(2, 3);
    console.log("Queue after splicing:", myQueue);
}
```

```
let n = parseInt(prompt("Enter number of elements to remove n: "));
let new1 = prompt("Enter first new element to insert: ");
let new2 = prompt("Enter second new element to insert: ");
let removed = myQueue.splice(i, n, new1, new2);
console.log("Removed elements:", removed);
console.log("Updated queue:", myQueue);
}

function divideQueue() {
    let mid = Math.ceil(myQueue.length / 2);
    let first = myQueue.slice(0, mid);
    let second = myQueue.slice(mid);
    console.log("First queue:", first);
    console.log("Second queue:", second);
}

function joinQueue() {
    let joined = myQueue.join(" ");
    console.log("Joined string:", joined);
}

function mapQueue() {
    myQueue = myQueue.map(e => String(e));
    console.log("Queue after converting to string:", myQueue);
}

// Menu loop
while (true) {
    let choice = menuQueue();
    switch (choice) {
        case "1": insertQueue(); break;
        case "2": removeQueue(); break;
        case "3": printCount(); break;
        case "4": sortQueue(); break;
        case "5": spliceQueue(); break;
        case "6": divideQueue(); break;
        case "7": joinQueue(); break;
        case "0": mapQueue(); break;
        default: mapQueue(); break;
    }
}
```