

ex. 1

```
function printDiamond(rows) {
  for (let i = 1; i <= rows; i++) {
    let spaces = ' '.repeat(rows - i);
    let stars = '*'.repeat(2 * i - 1);
    console.log(spaces + stars);
  }
  for (let i = rows - 1; i >= 1; i--) {
    let spaces = ' '.repeat(rows - i);
    let stars = '*'.repeat(2 * i - 1);
    console.log(spaces + stars);
  }
}

function printSquare(rows) {
  for (let i = 1; i <= rows; i++) {
    let stars = '*'.repeat(rows);
    console.log(stars);
  }
}

function printTriangle(rows) {
  for (let i = 1; i <= rows; i++) {
    let stars = '*'.repeat(i);
    console.log(stars);
  }
}
```

```
var numberOfRows = parseInt(prompt("Enter the number of rows:"));
```

```
var choice = prompt("Choose a geometric figure: \n1. Diamond\n2. Square\n3. Triangle");
```

```
switch (choice) {
  case '1':
    printDiamond(numberOfRows);
    break;
  case '2':
    printSquare(numberOfRows);
    break;
  case '3':
    printTriangle(numberOfRows);
    break;
  default:
    console.log("Invalid choice. Please enter a valid option.");
}
```

ex. 2

```
function printEmptyDiamond(rows) {
  for (let i = 1; i <= rows; i++) {
    let spaces = ' '.repeat(rows - i);
    let stars = (i === 1 || i === rows) ? '*' : '*' + ' '.repeat(2 * (i - 1)
- 1) + '*';
    console.log(spaces + stars);
  }
  for (let i = rows - 1; i >= 1; i--) {
    let spaces = ' '.repeat(rows - i);
    let stars = (i === 1 || i === rows) ? '*' : '*' + ' '.repeat(2 * (i - 1)
- 1) + '*';
    console.log(spaces + stars);
  }
}
```

```

}

function printEmptySquare(rows) {
  for (let i = 1; i <= rows; i++) {
    let stars = (i === 1 || i === rows) ? '*'.repeat(rows) : '*' + '
'.repeat(rows - 2) + '*';
    console.log(stars);
  }
}

function printEmptyTriangle(rows) {
  for (let i = 1; i <= rows; i++) {
    let stars = (i === 1 || i === rows) ? '*'.repeat(i) : '*' + ' '.repeat(i
- 2) + '*';
    console.log(stars);
  }
}

var numberOfRows = parseInt(prompt("Enter the number of rows:"));

var choice = prompt("Choose a geometric figure: \n1. Empty Diamond\n2. Empty
Square\n3. Empty Triangle");

switch (choice) {
  case '1':
    printEmptyDiamond(numberOfRows);
    break;
  case '2':
    printEmptySquare(numberOfRows);
    break;
  case '3':
    printEmptyTriangle(numberOfRows);
    break;
  default:
    console.log("Invalid choice. Please enter a valid option.");
}

```

ex. 3

```

// Traditional
let number = prompt("Enter number")
function calculateFactorialTraditional(number) {
  let result = 1;
  for (let i = 2; i <= number; i++) {
    result *= i;
  }
  return result;
}

// Recursive
function calculateFactorialRecursive(number) {
  if (number === 0 || number === 1) {
    return 1;
  } else {
    return number * calculateFactorialRecursive(number - 1);
  }
}

var userInput = parseInt(prompt("Enter a number to calculate its factorial:"));

if (!isNaN(userInput)) {

```

```

    var traditionalResult = calculateFactorialTraditional(userInput);
    console.log("Factorial (Traditional): " + traditionalResult);

    var recursiveResult = calculateFactorialRecursive(userInput);
    console.log("Factorial (Recursive): " + recursiveResult);
} else {
    console.log("Please enter a valid number.");
}

```

ex. 4

```

// Function to calculate factorial
function calculateFactorial(number) {
    if (number === 0 || number === 1) {
        return 1;
    } else {
        return number * calculateFactorial(number - 1);
    }
}

// Function to calculate the odds of winning a lottery
function calculateLotteryOdds(n, r) {
    if (n >= r && n >= 0 && r >= 0) {
        // Calculate n!/(r!(n-r)!)
        const numerator = calculateFactorial(n);
        const denominator = calculateFactorial(r) * calculateFactorial(n - r);
        return numerator / denominator;
    } else {
        console.log("Invalid input. Make sure n is greater than or equal to r, and both are non-negative.");
        return null;
    }
}

var n = parseInt(prompt("Enter the total number of possible numbers (n):"));
var r = parseInt(prompt("Enter the number of numbers chosen (r):"));

if (!isNaN(n) && !isNaN(r)) {
    var odds = calculateLotteryOdds(n, r);
    if (odds !== null) {
        console.log("Odds of winning the lottery: 1 in " + odds.toFixed(0));
    }
} else {
    console.log("Please enter valid numbers for n and r.");
}

```

ex. 5

```

function calculateSumRecursive(number) {
    // Base case: sum of 0 to 0 is 0
    if (number === 0) {
        return 0;
    } else {
        // Recursive case: sum of 1 to n is n + sum of 1 to (n-1)
        return number + calculateSumRecursive(number - 1);
    }
}

var userInput = parseInt(prompt("Enter a number to calculate the sum from 1 to that number recursively:"));

```

```

if (!isNaN(userInput) && userInput >= 0) {
    // Calculate and display the sum recursively
    var sum = calculateSumRecursive(userInput);
    console.log("Sum from 1 to " + userInput + " is: " + sum);
} else {
    console.log("Please enter a valid non-negative number.");
}

```

ex. 6

```

function countDigitsRecursive(number) {
    if (number < 10) {
        return 1;
    } else {
        return 1 + countDigitsRecursive(Math.floor(number / 10));
    }
}

var userInput = parseInt(prompt("Enter a number to calculate the number of
digits recursively:"));

if (!isNaN(userInput)) {
    var digitCount = countDigitsRecursive(Math.abs(userInput));
    console.log("Number of digits in " + userInput + " is: " + digitCount);
} else {
    console.log("Please enter a valid number.");
}

```

ex. 7

```

function reverseNumber(number) {
    return parseInt(number.toString().split('').reverse().join(''));
}

var userInput = parseInt(prompt("Enter a number to reverse its order:"));

if (!isNaN(userInput)) {
    var reversedNumber = reverseNumber(userInput);
    console.log("Reversed number: " + reversedNumber);
} else {
    console.log("Please enter a valid number.");
}

```