1. Write a Javascript function to check whether a triangle is equilateral, isosceles or scalene

```
function triangleType(side1, side2, side3) {
  if (side1 <= 0 || side2 <= 0 || side3 <= 0) {
    return "Invalid sides: All sides must be positive numbers.";
  }

  if (side1 + side2 <= side3 || side1 + side3 <= side2 || side2 + side3 <= side1) {
    return "Invalid sides: The sum of any two sides must be greater than the third side.";
  }

  if (side1 === side2 && side2 === side3) {
    return "Equilateral triangle";
  } else if (side1 === side2 || side1 === side3 || side2 === side3) {
    return "Isosceles triangle";
  } else {
    return "Scalene triangle";
  }
}

const side1 = 5;
const side2 = 5;
const side3 = 5;
console.log(`The triangle with sides ${side1}, ${side2}, and ${side3} is an ${triangleType(side1,
side2, side3)}.`);

const side4 = 5;
const side5 = 5;
const side6 = 8;
console.log(`The triangle with sides ${side4}, ${side5}, and ${side6} is an ${triangleType(side4,
side5, side6)}.`);

const side7 = 3;
const side8 = 4;
const side9 = 5;
```

```
console.log(`The triangle with sides ${side7}, ${side8}, and ${side9} is a ${triangleType(side7,
side8, side9)}.`);
```

2. Write a function using switch case to find the grade of a student based on marks obtained
a. “S grade” if the marks are between 90 and 100.
b. “A grade” if the marks are between 80 and 90.
c. “B grade” if the marks are between 70 and 80.
d. “C grade” if the marks are between 60 and 70.
e. “D grade” if the marks are between 50 and 60.
f. “E grade” if the marks are between 40 and 50.
g. “Student has failed” if the marks are between 0 and 40.
h. Else output “Invalid marks”.

```
function findGrade(marks) {
  let grade;

  switch (true) {
    case (marks >= 90 && marks <= 100):
      grade = "S grade";
      break;
    case (marks >= 80 && marks < 90):
      grade = "A grade";
      break;
    case (marks >= 70 && marks < 80):
      grade = "B grade";
      break;
    case (marks >= 60 && marks < 70):
      grade = "C grade";
      break;
    case (marks >= 50 && marks < 60):
      grade = "D grade";
      break;
    case (marks >= 40 && marks < 50):
      grade = "E grade";
      break;
    case (marks >= 0 && marks < 40):
      grade = "Student has failed";
      break;
    default:
      grade = "Invalid marks";
      break;
  }

  return grade;
}

const marks1 = 95;
console.log(`Marks: ${marks1}, Grade: ${findGrade(marks1)}`);

const marks2 = 85;
console.log(`Marks: ${marks2}, Grade: ${findGrade(marks2)}`);

const marks3 = 75;
console.log(`Marks: ${marks3}, Grade: ${findGrade(marks3)}`);

const marks4 = 65;
console.log(`Marks: ${marks4}, Grade: ${findGrade(marks4)}`);

const marks5 = 55;
console.log(`Marks: ${marks5}, Grade: ${findGrade(marks5)}`);

const marks6 = 45;
console.log(`Marks: ${marks6}, Grade: ${findGrade(marks6)}`);

const marks7 = 35;
console.log(`Marks: ${marks7}, Grade: ${findGrade(marks7)}`);

const marks8 = 105;
console.log(`Marks: ${marks8}, Grade: ${findGrade(marks8)}`);
```

3. Write a JavaScript program to find the sum of the multiples of 3 and 5 under 1000

```
function sumOfMultiples(limit) {
  let sum = 0;
  for (let i = 1; i < limit; i++) {
    if (i % 3 === 0 || i % 5 === 0) {
      sum += i;
    }
  }
  return sum;
}

const limit = 1000;
const result = sumOfMultiples(limit);
console.log(`The sum of the multiples of 3 and 5 under ${limit} is ${result}`);
```

4. Write a program to find the factorial of all prime numbers between a given range . Range will be passed as 2 values in the function parameters. eg- if it is needed to find the values for numbers 1-100, then function declaration can look like - function prime(1,100).

```
function isPrime(num) {
  if (num <= 1) return false;
  if (num <= 3) return true;

  if (num % 2 === 0 || num % 3 === 0) return false;

  for (let i = 5; i * i <= num; i += 6) {
    if (num % i === 0 || num % (i + 2) === 0) return false;
  }

  return true;
}

function factorial(n) {
  if (n < 0) {
    return -1; // Factorial of a negative number is not defined
  } else if (n === 0 || n === 1) {
    return 1; // Factorial of 0 or 1 is 1
  } else {
    let result = 1;
    for (let i = n; i > 1; i--) {
      result *= i;
    }
    return result;
  }
}

function primeFactorials(rangeStart, rangeEnd) {
  let result = [];

  for (let i = rangeStart; i <= rangeEnd; i++) {
    if (isPrime(i)) {
      result.push({ prime: i, factorial: factorial(i) });
    }
  }

  return result;
}


const rangeStart = 1;
const rangeEnd = 100;
const primeFactResults = primeFactorials(rangeStart, rangeEnd);

primeFactResults.forEach(item => {
  console.log(`Prime number: ${item.prime}, Factorial: ${item.factorial}`);
});
```