Function in JavaScript

Q 1. Create an Arrow Function to Calculate the Square of a Number

Arrow function:

```javascript
const square = (num) => num * num;

// Example usage:
const number = 5;
console.log(`Square of ${number} is:`, square(number)); // Output: Square of 5 is: 25
```

Q 2. Create a Function to Generate a Personalized Greeting

Function to generate a greeting:

```javascript
function generateGreeting(name) {
    return `Hello, ${name}! Welcome!`;
}

// Example usage:
console.log(generateGreeting("Alice"));
console.log(generateGreeting("Bob"));
console.log(generateGreeting("Charlie"));

// Output:
// Hello, Alice! Welcome!
// Hello, Bob! Welcome!
// Hello, Charlie! Welcome!
```

Q 3. Create an IIFE to Calculate the Square of a Number

Immediately Invoked Function Expression (IIFE):

```
(function(num) {
    console.log(`Square of ${num} is:`, num * num);
})(7);

// Output: Square of 7 is: 49
```

Q 4. Create a Function to Calculate Tax with a Closure for Different Tax Rates

Function with closure for tax rates:

```
function calculateTax(income) {
    return function() {
        if (income <= 10000) {
            return income * 0.1; // 10% tax for incomes <= 10,000
        } else if (income <= 50000) {
            return income * 0.2; // 20% tax for incomes <= 50,000
        } else {
            return income * 0.3; // 30% tax for incomes above 50,000
        }
    }
}

// Example usage:
const taxForLowIncome = calculateTax(8000)(); // Output: 800
const taxForMidIncome = calculateTax(30000)(); // Output: 6000
const taxForHighIncome = calculateTax(100000)(); // Output: 30000

console.log(taxForLowIncome, taxForMidIncome, taxForHighIncome);
```

Q 5. Create a Recursive Function to Calculate Factorial

Recursive factorial function:

```
function factorial(n) {
    if (n === 0 || n === 1) {
        return 1;
    } else {
        return n * factorial(n - 1);
    }
}

// Example usage:
console.log(factorial(5)); // Output: 120
console.log(factorial(7)); // Output: 5040
console.log(factorial(0)); // Output: 1
```

Q 6. Create a Curry Function

Curry function:

```
function curry(func) {
    return function curried(...args) {
        if (args.length >= func.length) {
            return func(...args);
        } else {
            return function(...nextArgs) {
                return curried(...args, ...nextArgs);
            };
        }
    };
}

// Example function to add two numbers
function add(a, b) {
    return a + b;
}

// Curried version of add
const curriedAdd = curry(add);

// Example usage:
console.log(curriedAdd(5)(3)); // Output: 8
```