1.Print Hello my dear N times using Recursion

```
public class HelloRecursion {
    static void printHello(int n) {
        if (n == 0) return;
        System.out.println("Hello my dear");
        printHello(n - 1);
    }
    public static void main(String[] args) {
        int N = 5;
        printHello(N);
    }
}
```

2.Print Numbers from N to 1

```
public class PrintNTo1 {
    static void print(int n) {
        if (n == 0) return;
        System.out.println(n);
        print(n - 1);
    }
    public static void main(String[] args) {
        int N = 5;
        print(N);
    }
}
```

3.Print Numbers from 1 to N

```
public class Print1ToN {
    static void print(int n) {
        if (n == 0) return;
        print(n - 1);
        System.out.println(n);
    }
    public static void main(String[] args) {
        int N = 5;
        print(N);
    }
}
```

4.Find a factorial of a Number

```
public class Factorial {
    static int factorial(int n) {
        if (n == 0 || n == 1) return 1;
        return n * factorial(n - 1);
    }

    public static void main(String[] args) {
        int N = 5;
        System.out.println("Factorial of " + N + " is " + factorial(N));
    }
}
```

5.Find nth fibonacci Number

```
public class Fibonacci {
    static int fibonacci(int n) {
        if (n == 0) return 0;
        if (n == 1) return 1;
        return fibonacci(n - 1) + fibonacci(n - 2);
    }

    public static void main(String[] args) {
        int N = 6;
        System.out.println("Fibonacci number at position " + N + " is " + fibonacci(N));
    }
}
```

6.Sum of first N natural Numbers

```
public class SumNatural {
    static int sum(int n) {
        if (n == 0) return 0;
        return n + sum(n - 1);
    }

    public static void main(String[] args) {
        int N = 5;
        System.out.println("Sum of first " + N + " natural numbers is " + sum(N));
    }
}
```

7.Find the sum of digits of a number

```
public class SumOfDigits {
    static int sumDigits(int n) {
        if (n == 0) return 0;
        return (n % 10) + sumDigits(n / 10);
    }

    public static void main(String[] args) {
        int num = 1234;
        System.out.println("Sum of digits of " + num + " is " + sumDigits(num));
    }
}
```

8.Count the Number of digits of a number

```
public class CountDigits {
    static int countDigits(int n) {
        if (n == 0) return 0;
        return 1 + countDigits(n / 10);
    }

    public static void main(String[] args) {
        int num = 12345;
        if (num == 0)
            System.out.println("Number of digits: 1");
        else
            System.out.println("Number of digits: " + countDigits(num));
    }
}
```

9.Print Even Numbers from 2 to N

```
public class RecursionPractice {
    public static void printHello(int n) {
        if (n == 0)
            return;
        System.out.println("Hello my dear");
        printHello(n - 1);
    }

    public static void main(String[] args) {
        int n = 5;
        printHello(n);
    }
}
```

10.Print a number in reverse

```
public class ReverseNumber {
    static void reverse(int n) {
        if (n == 0) return;
        System.out.print(n % 10);
        reverse(n / 10);
    }

    public static void main(String[] args) {
        int num = 1234;
        System.out.print("Reversed number: ");
        if (num == 0)
            System.out.print(0);
        else
            reverse(num);
    }
}
```

11.Calculate product of first N natural Numbers

```
public class ProductOfN {
    static int product(int n) {
        if (n == 0 || n == 1) return 1;
        return n * product(n - 1);
    }

    public static void main(String[] args) {
        int N = 5;
        System.out.println("Product of first " + N + " natural numbers is " + product(N));
    }
}
```

12.Print sum of an array using Recursion

```
public class ArraySum {
    static int sum(int[] arr, int n) {
        if (n == 0) return 0;
        return arr[n - 1] + sum(arr, n - 1);
    }

    public static void main(String[] args) {
        int[] arr = {1, 2, 3, 4, 5};
        System.out.println("Sum of array: " + sum(arr, arr.length));
    }
}
```

13.Print all elements of an array using recursion

```
public class PrintArray {
    static void printElements(int[] arr, int index) {
        if (index == arr.length) return;
        System.out.println(arr[index]);
        printElements(arr, index + 1);
    }

    public static void main(String[] args) {
        int[] arr = {10, 20, 30, 40, 50};
        printElements(arr, 0);
    }
}
```

14.Calculate sum of squares of first N numbers

```
public class SumOfSquares {
    static int sumSquares(int n) {
        if (n == 0) return 0;
        return (n * n) + sumSquares(n - 1);
    }
    public static void main(String[] args) {
        int N = 4;
        System.out.println("Sum of squares of first " + N + " numbers is " + sumSquares(N));
    }
}
```

15.Check if number is palindrome

```
public class PalindromeCheck {
    static int reverse(int n, int rev) {
        if (n == 0) return rev;
        return reverse(n / 10, rev * 10 + (n % 10));
    }

    public static void main(String[] args) {
        int num = 121;
        int reversed = reverse(num, 0);

        if (num == reversed)
            System.out.println(num + " is a palindrome");
        else
            System.out.println(num + " is not a palindrome");
    }
}
```