Answer 1)

```
public class HelloWorld {
    public static boolean isPowerOfThree(int n) {
        if (n == 1) {
            return true;
        } else if (n <= 0 || n % 3 != 0) {
            return false;
        } else {
            return isPowerOfThree(n / 3);
        }
    }

    public static void main(String[] args) {
        int n1 = 27;
        System.out.println(n1 + " is a power of three? " + isPowerOfThree(n1));

        int n2 = 45;
        System.out.println(n2 + " is a power of three? " + isPowerOfThree(n2));

        int n3 = 9;
        System.out.println(n3 + " is a power of three? " + isPowerOfThree(n3));
    }
}
```

Answer 2)

```
public class HelloWorld {
    public static int getLastRemaining(int n) {
        return findLastRemaining(n, true);
    }

    private static int findLastRemaining(int n, boolean isLeftToRight) {
        if (n == 1) {
            return 1;
        }

        if (isLeftToRight) {
            return 2 * findLastRemaining(n / 2, false);
        } else {
            if (n % 2 == 1) {
                return 2 * findLastRemaining(n / 2, true);
            } else {
                return 2 * findLastRemaining(n / 2, true) - 1;
            }
        }
    }

    public static void main(String[] args) {
        int n1 = 9;
        System.out.println("Last remaining number: " + getLastRemaining(n1));

        int n2 = 16;
        System.out.println("Last remaining number: " + getLastRemaining(n2));

        int n3 = 7;
        System.out.println("Last remaining number: " + getLastRemaining(n3));
    }
}
```

Answer 3)

```
import java.util.ArrayList;
import java.util.List;

public class HelloWorld {
    public static void printSubsets(String set) {
        List<String> subsets = new ArrayList<>();
        generateSubsets(set, "", 0, subsets);
        System.out.println(subsets);
    }

    private static void generateSubsets(String set, String currentSubset, int index, List<String> subsets) {
        if (index == set.length()) {
            subsets.add(currentSubset);
            return;
        }

        generateSubsets(set, currentSubset, index + 1, subsets);
        generateSubsets(set, currentSubset + set.charAt(index), index + 1, subsets);
    }

    public static void main(String[] args) {
        String set1 = "abc";
        System.out.println("Subsets of \"" + set1 + "\":");
        printSubsets(set1);

        String set2 = "abcd";
        System.out.println("Subsets of \"" + set2 + "\":");
        printSubsets(set2);
    }
}
```

Answer 4)

```
public class HelloWorld {
    public static int calculateLength(String str) {
        if (str.isEmpty()) {
            return 0;
        } else {
            return 1 + calculateLength(str.substring(1));
        }
    }

    public static void main(String[] args) {
        String str1 = "abcd";
        System.out.println("Length of \"" + str1 + "\": " + calculateLength(str1));

        String str2 = "GEEKSFORGEEKS";
        System.out.println("Length of \"" + str2 + "\": " + calculateLength(str2));
    }
}
```

Answer 5)

```
// Online Java Compiler
// Use this editor to write, compile and run your Java code online

public class HelloWorld {
    public static int countContiguousSubstrings(String str) {
        return countSubstrings(str, 0, str.length() - 1);
    }

    private static int countSubstrings(String str, int start, int end) {
        if (start > end) {
            return 0;
        }

        int count = countSubstrings(str, start, end - 1) + countSubstringsWithSameStartEnd(str,
start, end);
        return count;
    }

    private static int countSubstringsWithSameStartEnd(String str, int start, int end) {
        if (start == end) {
            return 1;
        }

        int count = 0;
        if (str.charAt(start) == str.charAt(end)) {
            count = 1 + countSubstringsWithSameStartEnd(str, start + 1, end - 1);
        }

        return count;
    }

    public static void main(String[] args) {
        String str1 = "abcab";
        System.out.println("Count of contiguous substrings in \"" + str1 + "\": " +
countContiguousSubstrings(str1));

        String str2 = "aba";
        System.out.println("Count of contiguous substrings in \"" + str2 + "\": " +
countContiguousSubstrings(str2));
    }
}
```

Answer 6)

```
// Online Java Compiler
// Use this editor to write, compile and run your Java code online

public class TowerOfHanoi {
    public static void moveDisks(int n, String source, String auxiliary, String destination) {
        if (n == 1) {
            System.out.println("Move disk 1 from rod " + source + " to rod " + destination);
            return;
        }

        moveDisks(n - 1, source, destination, auxiliary);
        System.out.println("Move disk " + n + " from rod " + source + " to rod " + destination);
        moveDisks(n - 1, auxiliary, source, destination);
    }

    public static int countMoves(int n) {
        if (n == 1) {
            return 1;
        }
        return 2 * countMoves(n - 1) + 1;
    }

    public static void main(String[] args) {
        int n = 2;
        System.out.println("Steps to solve Tower of Hanoi with " + n + " disks:");
        moveDisks(n, "1", "2", "3");
        System.out.println("Total moves: " + countMoves(n));
    }
}
```

Answer 7)

```
public class StringPermutations {
    public static void printPermutations(String str) {
        permute(str, 0, str.length() - 1);
    }

    private static void permute(String str, int left, int right) {
        if (left == right) {
            System.out.println(str);
        } else {
            for (int i = left; i <= right; i++) {
                str = swap(str, left, i);
                permute(str, left + 1, right);
                str = swap(str, left, i);
            }
        }
    }
    private static String swap(String str, int i, int j) {
        char[] charArray = str.toCharArray();
        char temp = charArray[i];
        charArray[i] = charArray[j];
        charArray[j] = temp;
        return String.valueOf(charArray);
    }

    public static void main(String[] args) {
        String str1 = "cd";
        System.out.println("Permutations of \"" + str1 + "\":");
        printPermutations(str1);

        String str2 = "abb";
        System.out.println("Permutations of \"" + str2 + "\":");
        printPermutations(str2);
    }
}
```

Answer 8)

```
public class ConsonantCounter {
    public static int countConsonants(String str) {
        return countConsonantsHelper(str.toLowerCase(), 0);
    }

    private static int countConsonantsHelper(String str, int index) {
        if (index == str.length()) {
            return 0;
        }

        char currentChar = str.charAt(index);
        if (isConsonant(currentChar)) {
            return 1 + countConsonantsHelper(str, index + 1);
        } else {
            return countConsonantsHelper(str, index + 1);
        }
    }

    private static boolean isConsonant(char ch) {
        return ch >= 'a' && ch <= 'z' && !isVowel(ch);
    }
    private static boolean isVowel(char ch) {
        return ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u';
    }

    public static void main(String[] args) {
        String str1 = "abc de";
        System.out.println("Number of consonants in \"" + str1 + "\": " + countConsonants(str1));

        String str2 = "geeksforgeeks portal";
        System.out.println("Number of consonants in \"" + str2 + "\": " + countConsonants(str2));
    }
}
```