Assignment 9 Solution - Recursion | DSA

Question 1

Given an integer `n`, return `true` if it is a power of two. Otherwise, return `false`.
An integer `n` is a power of two, if there exists an integer `x` such that `n == 2x`.

Example 1:
Input: n = 1
Output: true

Example 2:
Input: n = 16
Output: true

Example 3:
Input: n = 3
Output: false

Solution Code:

```
package in.ineuron.pptAssignment09;

public class IsPowerOfTwo {
    public static boolean isPowerOfTwo(int n) {
        if (n <= 0) {
            return false; // Negative numbers and zero are not powers of two
        }

        // Check if there is only one set bit in the binary representation of n
        // If so, n is a power of two
        return (n & (n - 1)) == 0;
    }

    public static void main(String[] args) {
        int n = 16;
        boolean result = isPowerOfTwo(n);
        System.out.println(result);
    }
}
```

Question 2

Given a number n, find the sum of the first natural numbers.

Example 1:
Input: n = 3
Output: 6

Example 2:
Input : 5
Output : 15

Solution Code:

```
package in.ineuron.pptAssignment09;

public class SumOfFirstNNumbers {
	public static int sumOfFirstNNumbers(int n) {
		return (n * (n + 1)) / 2;
	}

	public static void main(String[] args) {
		int n = 3;
		int sum = sumOfFirstNNumbers(n);
		System.out.println(sum);
	}
}
```

Question 3

Given a positive integer, N. Find the factorial of N.

Example 1:
Input: N = 5
Output: 120

Example 2:
Input: N = 4
Output: 24

Solution Code:

```
package in.ineuron.pptAssignment09;

public class Factorial {
    public static int factorial(int n) {
        if (n == 0 || n == 1) {
            return 1; // Base case: factorial of 0 or 1 is 1
        } else {
            return n * factorial(n - 1);
             // Recursive case: factorial of n is n multiplied by factorial of n-1
        }
    }

    public static void main(String[] args) {
        int N = 5;
        int result = factorial(N);
        System.out.println(result);
    }
}
```

Question 4

Given a number N and a power P, the task is to find the exponent of this number raised to the given power, i.e. N^P.

Example 1 :
Input: N = 5, P = 2
Output: 25

Example 2 :
Input: N = 2, P = 5
Output: 32

Solution Code:

```
package in.ineuron.pptAssignment09;

public class CalculateExponent {
        public static double calculateExponent(int N, int P) {
                return Math.pow(N, P);
        }

        public static void main(String[] args) {
                int N = 5;
                int P = 2;
                double result = calculateExponent(N, P);
                System.out.println(result);
        }
}
```

Question 5

Given an array of integers arr, the task is to find maximum element of that array using recursion.

Example 1:
Input: arr = {1, 4, 3, -5, -4, 8, 6};
Output: 8

Example 2:
Input: arr = {1, 4, 45, 6, 10, -8};
Output: 45

Solution Code:

```
package in.ineuron.pptAssignment09;

public class FindMax {

    public static int findMax(int[] arr, int start, int end) {
        if (start == end) {
            return arr[start]; // Base case: when there is only one element
        } else {
            int mid = (start + end) / 2;
            int max1 = findMax(arr, start, mid);
                // Maximum element in the first half of the array
            int max2 = findMax(arr, mid + 1, end);
                // Maximum element in the second half of the array
            return Math.max(max1, max2); // Return the maximum of the two halves
        }
    }

    public static void main(String[] args) {
        int[] arr = { 1, 4, 3, -5, -4, 8, 6 };
        int max = findMax(arr, 0, arr.length - 1);
        System.out.println(max);
    }
}
```

Question 6

Given first term (a), common difference (d) and a integer N of the Arithmetic Progression series, the task is to find Nth term of the series.

Example 1:
Input : a = 2 d = 1 N = 5
Output : 6
The 5th term of the series is : 6

Example 2:
Input : a = 5 d = 2 N = 10
Output : 23
The 10th term of the series is : 23

Solution Code:

```
package in.ineuron.pptAssignment09;

public class FindNthTerm {

    public static int findNthTerm(int a, int d, int N) {
        return a + (N - 1) * d;
    }

    public static void main(String[] args) {
        int a = 2;
        int d = 1;
        int N = 5;
        int nthTerm = findNthTerm(a, d, N);
        System.out.println("The " + N + "th term of the series is: " + nthTerm);
    }
}
```

Question 7

Given a string S, the task is to write a program to print all permutations of a given string.

Example 1:

Input:S = “ABC”

Output:“ABC”, “ACB”, “BAC”, “BCA”, “CBA”, “CAB”

Example 2:

Input:S = “XY”

Output:“XY”, “YX”

Solution Code:

```
package in.ineuron.pptAssignment09;
import java.util.ArrayList;
import java.util.List;
public class FindPermutations {

    public static List<String> findPermutations(String s) {
        List<String> permutations = new ArrayList<>();
        backtrack(permutations, s.toCharArray(), 0, s.length() - 1);
        return permutations;
    }

    public static void backtrack(List<String> permutations, char[] sArr, int start, int end) {
        if (start == end) {
            permutations.add(new String(sArr));
        } else {
            for (int i = start; i <= end; i++) {
                swap(sArr, start, i);
                backtrack(permutations, sArr, start + 1, end);
                swap(sArr, start, i);
            }
        }
    }
    public static void swap(char[] arr, int i, int j) {
        char temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }
    public static void main(String[] args) {
        String S = "ABC";
        List<String> permutations = findPermutations(S);
        for (String permutation : permutations) {
            System.out.println(permutation);
        }
    }
}
```

Question 8

Given an array, find a product of all array elements.

Example 1:
Input : arr[] = {1, 2, 3, 4, 5}
Output : 120

Example 2:
Input : arr[] = {1, 6, 3}
Output : 18

Solution Code:

```
package in.ineuron.pptAssignment09;

public class ProductOfArrayElements {

    public static long getProductOfArrayElements(int[] arr) {
        long product = 1;
        for (int num : arr) {
            product *= num;
        }
        return product;
    }

    public static void main(String[] args) {
        int[] arr = { 1, 2, 3, 4, 5 };
        long product = getProductOfArrayElements(arr);
        System.out.println(product);
    }
}
```