Assignment No. :- 3

Problem:-1: Print 1 to N by using Recursion

```java
JavaProjects > J Print_1toN_usingrecursion.java > ...
1    public class Print_1toN_usingrecursion {
2        public static void print(int n){
3            if(n==1){
4                System.out.print(n+" ");
5                return;
6            }
7            print(n-1);
8            System.out.print(n+" ");
9        }
     Run | Debug
10       public static void main(String[] args) {
11           int n=10;
12           print(n);
13       }
14   }
15
```

Problem:-2: Print N to 1 by using Recursion

```java
JavaProjects > J Print_Nto1_usingrecursion.java > Print_Nto1_usingrecursion > main(String[])
1  v public class Print_Nto1_usingrecursion {
2  v     public static void print(int n){
3  v         if(n==1){
4                  System.out.print(n+" ");
5                  return;
6              }
7              System.out.print(n+" ");
8              print(n-1);
9          }
       Run | Debug
10 v       public static void main(String[] args) {
11             int n=10;
12             print(n);
13         }
14     }
15
```

Problem:-3: Sum of N numbers

```java
JavaProjects > J Sum_N_numbers.java > ⚡ Sum_N_numbers > ◉ main(String[])
1    public class Sum_N_numbers {
2        public static int sum(int n){
3            if(n==1){
4                return 1;
5            }
6            return n + sum(n-1);
7        }
     Run | Debug
8        public static void main(String[] args) {
9            int n=5;
10           System.out.println(sum(n));
11       }
12   }
13
```

Problem:-4: Reverse of String

```java
JavaProjects > J Reverse_String.java > ⚡ Reverse_String > ◉ reverse(String)
1    public class Reverse_String {
2        public static String reverse(String str){
3            if(str.isEmpty()){
4                return str;
5            }
6            return str.charAt(str.length()-1) + reverse(str.substring(beginIndex:0, str.length()-1));
7        }
     Run | Debug
8        public static void main(String[] args) {
9            String str="Ankit";
10           System.out.println(reverse(str));
11       }
12   }
13
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS                                          +

PS C:\Users\ankit\OneDrive\Desktop\JavaProjects> cd "c:\Users\ankit\OneDrive\Desktop\JavaProjects\" ; if ($?) { javac Reve
rse_String.java } ; if ($?) { java Reverse_String }
tiknA
PS C:\Users\ankit\OneDrive\Desktop\JavaProjects> |
```

Problem:-5: Sum of Array Elements

```java
JavaProjects > J TreasureBoxesSum.java > ...
1    import java.util.Scanner;
2
3    public class TreasureBoxesSum {
4        // sum array elements
5        static int sum(int[] arr, int n) {
6            if (n == 0) return 0;
7            return arr[n - 1] + sum(arr, n - 1);
8        }
9
     Run | Debug
10       public static void main(String[] args) {
11           Scanner sc = new Scanner(System.in);
12           int n = sc.nextInt();
13           int[] coins = new int[n];
14           for (int i = 0; i < n; i++) {
15               coins[i] = sc.nextInt();
16           }
17           System.out.println(sum(coins, n));
18       }
19   }
```

```
PROBLEMS 1    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Users\ankit\OneDrive\Desktop\JavaProjects> cd "c:\Users\ankit\OneDrive\Desktop\JavaProjects\" ; if ($?) { javac Trea
sureBoxesSum.java } ; if ($?) { java TreasureBoxesSum }
5
2
4
5
3
1
15
PS C:\Users\ankit\OneDrive\Desktop\JavaProjects> 
```

Problem:-6: Number of distinct ways to reach the top using recursion

```java
JavaProjects > J TravelersStep.java > TravelersStep > main(String[])
1    import java.util.Scanner;
2
3    public class TravelersStep {
4        // ways to climb stairs
5        static int countWays(int n) {
6            if (n == 0) return 1;
7            if (n < 0) return 0;
8            return countWays(n - 1) + countWays(n - 2);
9        }
10
     Run | Debug
11       public static void main(String[] args) {
12           Scanner sc = new Scanner(System.in);
13           int n = sc.nextInt();
14           System.out.println(countWays(n));
15       }
16   }
17
```

```
PROBLEMS 2    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Users\ankit\OneDrive\Desktop\JavaProjects> cd "c:\Users\ankit\OneDrive\Desktop\JavaProjects\" ; if ($?) { javac Trav
elersStep.java } ; if ($?) { java TravelersStep }
5
8
PS C:\Users\ankit\OneDrive\Desktop\JavaProjects> 
```

Problem:-7: Factorial of a Number

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

    Run | Debug
    public static void main(String[] args) {
        java.util.Scanner sc = new java.util.Scanner(System.in);
        int n = sc.nextInt();
        System.out.println(factorial(n));
    }
}
```

PROBLEMS 3   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
PS C:\Users\ankit\OneDrive\Desktop\JavaProjects> cd "c:\Users\ankit\OneDrive\Desktop\JavaProjects\" ; if ($?) { javac Fact
orial.java } ; if ($?) { java Factorial }
5
120
PS C:\Users\ankit\OneDrive\Desktop\JavaProjects>
```

Problem:-8: Fibonacci number at n.

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

    Run | Debug
    public static void main(String[] args) {
        java.util.Scanner sc = new java.util.Scanner(System.in);
        int n = sc.nextInt();
        System.out.println(fibonacci(n));
    }
}
```

PROBLEMS 4   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
PS C:\Users\ankit\OneDrive\Desktop\JavaProjects> cd "c:\Users\ankit\OneDrive\Desktop\JavaProjects\" ; if ($?) { javac Fibo
nacci.java } ; if ($?) { java Fibonacci }
6
8
PS C:\Users\ankit\OneDrive\Desktop\JavaProjects> cd "c:\Users\ankit\OneDrive\Desktop\JavaProjects\" ; if ($?) { javac Fibo
nacci.java } ; if ($?) { java Fibonacci }
2
1
PS C:\Users\ankit\OneDrive\Desktop\JavaProjects>
```