RECURSION

1. Binary String Problem

```java
import java.util.Scanner;

public class binaryStringsProblem {
    public static void printBinStrings (int n, int lastPlace, String str) {
        if (n == 0) {
            System.out.println(str);
            return;
        }
        printBinStrings(n-1, 0, str + "0");
        if (lastPlace == 0) {
            printBinStrings(n-1, 1, str + "1");
        }
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter no. of Places of Binary Number: ");
        int n = sc.nextInt();
        System.out.print("Enter what will be at Last place (0/1): ");
        int lp = sc.nextInt();
        printBinStrings(n, lp, " ");
    }
}
```

2. Contiguous Substring

```java
import java.util.Scanner;
public class countiguousSubstrings {
    public static int countSubstrs(String str, int i, int j, int n) {
        if (n == 1) {
            return 1;
        }
        if (n <= 0) {
            return 0;
        }
        int res = countSubstrs(str, i + 1, j, n - 1) +
                countSubstrs(str, i, j - 1, n - 1) -
                countSubstrs(str, i + 1, j - 1, n - 2);
        if (str.charAt(i) == str.charAt(j)) {
            res++;
        }
        return res;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter String: ");
        String str = sc.next();
        int n = str.length();
        System.out.print("Total no. of Contiguous substrings starting & ending with the same
character are: ");
        System.out.println(countSubstrs(str, 0, n-1, n));
    }
}
```

3. Factorial

```java
import java.util.Scanner;

public class Factorial {

    public static int fact(int n) {
        if (n == 0) {
            return 1;
        }
        int fnm1 = fact(n-1);
        int fn = n * fact(n-1);
        return fn;                                          // SPACE COMPLEXITY = O(n)
    }                                                       // TIME COMPLEXITY = O(n)
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter value of n: ");
        int n = sc.nextInt();
        System.out.print("Factorial of " +n+ " is: ");
        System.out.println(fact(n));
    }
}
```

4. Fibonacci Number

```java
import java.util.Scanner;

public class FibonacciNumber {

    public static int fib (int n) {
        if (n==0 || n==1) {
            return n;
        }
        int fnm1 = fib(n-1);
        int fnm2 = fib(n-2);
        int fn = fnm1 + fnm2;
        return fn;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter Nth Term: ");
        int n = sc.nextInt();
        System.out.print("The " +n+ "th term of Fibonacci Series is: ");
        System.out.println(fib(n));
    }
}
```

5. Friends Coupling Problem

```java
import java.util.Scanner;

public class friendsPairingProblem {
    public static int friendsPairing(int n) {
        if (n == 1 || n == 2) {
            return n;
        }

        // SINGLE
        int fnm1 = friendsPairing(n-1);
        //PAIR
        int fnm2 = friendsPairing(n-2);

        int pairWays = (n-1) * fnm2;
        int totWays = fnm1 + pairWays;
        return totWays;

        // from line no.7 to line no.14 can be also written in one line i.e.,
        // return friendsPairing(n-1) + (n-1) * friendsPairing(n-2);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter no. of Friends: ");
        int frnds = sc.nextInt();
        System.out.println("Total ways of Coupling " +frnds+ " friends is: "
+friendsPairing(frnds));
    }
}
```

6. Hanoi Problem

```java
public class HanoiProblem {
    // src-source, helper, dest-destination
    public static void towerOfHanoi (int n, String src, String helper, String dest) {
        if (n == 1) {
            System.out.println("Transfer disk " + n + " from " + src + " to " + dest);
            return;
        }
        // transfer top n-1 from src to helper using dest as helper
        towerOfHanoi(n-1, src, dest, helper);

        // transfer nth from src to destination
        System.out.println("Transfer disk " +n+ " from " +src+ " to " +dest);

        // transfer n-1 from helper to destination, using source as helper.
        towerOfHanoi(n-1, helper, src, dest);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter the no. of Disks: ");
        int n = sc.nextInt();
        towerOfHanoi(n, "A", "B", "C");
    }
}
```

7. First Occurrence

```java
import java.util.Scanner;

public class IstOccurence {
    public static int firstOccurence (int arr[], int key, int i) {
        if (i == arr.length) {
            return -1;
        }
        if (arr[i] == key) {
            return i;
        }
        return firstOccurence(arr, key, i+1);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter size of Array: ");
        int size = sc.nextInt();
        int arr[] = new int[size];

        System.out.println("Enter the Elements:");
        for (int i = 0; i < size; i++) {
            arr[i] = sc.nextInt();
        }
        System.out.print("Enter the Key value: ");
        int key = sc.nextInt();
        System.out.print("Enter Index i: ");
        int i = sc.nextInt();
        System.out.println("First Occurence of " +key+ " is at Index: " +firstOccurence(arr, key,
i));
    }
}
```

8. Key Occurrence

```java
import java.util.Scanner;

public class keyOccurences {
    public static void allOccurences (int arr[], int key, int i) {
        if (i == arr.length) {
            return;
        }
        if (arr[i] == key) {
            System.out.print(i + " ");
        }
        allOccurences(arr, key, i+1);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter size of an Array: ");
        int size = sc.nextInt();
        int arr[] = new int[size];
        System.out.print("Enter ELements: ");
        for (int i = 0; i < size; i++) {
            arr[i] = sc.nextInt();
        }
```

```
        System.out.print("Enter Key: ");
        int key = sc.nextInt();
        System.out.print("Key " +key+ " is at Index: ");
        allOccurences(arr, key, 0);
        System.out.println();
    }
}
```

9. Last Occurrence

```
import java.util.Scanner;

public class LastOccurence {

    public static int lastOccurence (int arr[], int key, int i) {
        if (i == arr.length) {
            return -1;
        }
        int isFound = lastOccurence(arr, key, i+1);
        if (isFound == -1 && arr[i] == key) {
            return i;
        }
        return isFound;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter size of Array: ");
        int size = sc.nextInt();
        int arr[] = new int[size];

        System.out.println("Enter the Elements:");
        for (int i = 0; i < size; i++) {
            arr[i] = sc.nextInt();
        }
        System.out.print("Enter the Key value: ");
        int key = sc.nextInt();
        System.out.print("Enter Index i: ");
        int i = sc.nextInt();
        System.out.println("Last Occurence of " +key+ " is at Index: " +lastOccurence(arr, key,
i));
    }
}
```

10. Length of String

```java
import java.util.Scanner;

public class lengthOfString {
    public static int length(String str) {
        if (str.length() == 0) {
            return 0;
        }
        return length(str.substring(1)) + 1;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a String: ");
        String str = sc.next();
        System.out.println("Length of a String is: " +length(str));
        if (length(str) == 7) {
            System.out.println("Thala for a Reason");
        }
    }
}
```

11.Numbers into Digit

```java
import java.util.Scanner;

public class numbersIntoDigit {
    static String digits[] = {"zero", "one", "two", "three", "four", "five", "six", "seven",
"eight", "nine"};
    public static void printDigits(int number) {
        if (number == 0) {
            return;
        }
        int lastDigit = number % 10;
        printDigits(number/10);
        System.out.print(digits[lastDigit]+ " ");
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter the Number: ");
        int num = sc.nextInt();
        System.out.print(+ num + " in Digits is: ");
        printDigits(num);
        System.out.println();
    }
}
```

12.Number 1 to n

```java
import java.util.Scanner;

public class numFrom1toN {

    public static void printInc (int n) {
        if (n == 1) {
            System.out.print(n+ " ");
            return;
        }
        printInc(n-1);
        System.out.print(n+ " ");
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter value of n: ");
        int n = sc.nextInt();
        printInc(n);
    }
}
```

13. Number n to 1

```java
import java.util.Scanner;

public class numFromNto1 {

    public static void printDec(int n) {
        if (n == 1) {
            System.out.print(n);
            return;
        }
        System.out.print(n+ " ");
        printDec(n-1);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter the value of n: ");
        int n = sc.nextInt();
        printDec(n);
    }
}
```

14. Removing Duplicates

```
import java.util.Scanner;

public class RemovingDuplicates {
    public static void removeDuplicates(String str, int index, StringBuilder newStr, boolean
map[]) {
        if (index == str.length()) {
            System.out.println(newStr);
            return;
        }
        char currChar = str.charAt(index);
        if (map [currChar - 'a'] == true) {
            removeDuplicates(str, index+1, newStr, map);
        } else {
            map[currChar - 'a'] = true;
            removeDuplicates(str, index+1, newStr.append(currChar), map);
        }
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter the String: ");
        String str = sc.next();
        removeDuplicates(str, 0, new StringBuilder(""), new boolean[26]);
    }
}
```

15. Sorted or Not

```
import java.util.Scanner;
public class SortedorNot {

    public static boolean isSorted (int arr[], int i) {
        if (i == arr.length-1) {
            return true;
        }
        if (arr[i] > arr[i+1]) {
            return false;
        }
        return isSorted(arr, i+1);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter size of Array: ");
        int size = sc.nextInt();
        int arr[] = new int[size];

        System.out.println("Enter the Elements:");
        for (int i = 0; i < size; i++) {
            arr[i] = sc.nextInt();
        }
        System.out.print("Enter value of Index i: ");
        int i = sc.nextInt();
        System.out.println("Array is Sorted. " + isSorted(arr, i));
    }
}
```

16. Sum of N

```java
import java.util.Scanner;

public class SumOfN {

    public static int sum(int n) {
        if (n == 1) {
            return 1;
        }
        int Snm1 = sum(n-1);
        int Sn = n + Snm1;
        return Sn;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter value of N: ");
        int n = sc.nextInt();
        System.out.println(sum(n));
    }
}
```

17.Tilling Problem

```java
import java.util.Scanner;

public class TillingProblem {
    public static int tillingProblem(int n) { // 2 * n

        // BASE CASE
        if (n == 0 || n == 1) {
            return 1;
        }
        // VERTICAL CHOICE
        int fnm1 = tillingProblem(n-1);
        // HORIZONTAL CHOICE
        int fnm2 = tillingProblem(n-2);

        int totWays = fnm1 + fnm2;
        return totWays;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter value of n: ");
        int n = sc.nextInt();
        System.out.println("Total ways of Tilling: " +tillingProblem(n));
    }
}
```

18. X to power n

```java
import java.util.Scanner;

public class xToPowern {  // TIME COMPLEXITY - O(n)
    public static int power (int x, int n) {
        if (n == 0) {
            return 1;
        }
        int xnm1 = power(x, n-1);
        int xn = x * xnm1;
        return xn;
            // OR for above 3 lines you can write the below 1 line.
        // return x * power(x, n-1);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter the base value: ");
        int base = sc.nextInt();
        System.out.print("Enter the power value: ");
        int pow = sc.nextInt();
        System.out.println(base+ "^" +pow+ " is: " +power(base, pow));
    }
}
```

19. X to power n (2^{nd} method)

```java
import java.util.Scanner;
public class xToPowern2nd {
    public static int optimizedPower(int a, int n) {
        if (n == 0) {
            return 1;
        }
        // 1. Time Complexity - O(n)
        // int halfPowerSq = optimizedPower(a, n/2) * optimizedPower(a, n/2);

                    // OR

        // 2. Time Complexity - O(log n)
        int halfPower = optimizedPower(a, n/2);
        int halfPowerSq = halfPower * halfPower;

        if (n % 2 != 0) {
            halfPowerSq = a * halfPowerSq;
        }
        return halfPowerSq;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter the base value: ");
        int base = sc.nextInt();
        System.out.print("Enter the power value: ");
        int pow = sc.nextInt();
        System.out.println(base+ "^" +pow+ " is: " + optimizedPower(base, pow));
    }
}
```