DAY-7

1.String Anagram

```
import java.io.*;
import java.util.*;


public class Solution {
   public static  String sortString(String str) {
      char[] ch=str.toCharArray();
      Arrays.sort(ch);
      return String.valueOf(ch);
   }
   public static boolean isAnagram(String s1,String s2) {
      s1=sortString(s1);
      s2=sortString(s2);
      return s1.equals(s2);
   }
   public static void main(String[] args) {
      Scanner sc=new Scanner(System.in);
      String s1=sc.nextLine();
      String s2=sc.nextLine();
      if(isAnagram(s1, s2)) {
         System.out.println("The given strings are an anagram");
      } else {
         System.out.println("The given strings are not an anagram");
      }
   }
}
```

2.Alternating code

```
import java.io.*;
import java.util.*;
import java.util.*;
```

```
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String s = sc.nextLine();
        if (s.length() < 2) {
            System.out.println("No");
            return;
        }

        char first = s.charAt(0);
        char second = '\0';
        for (int i = 1; i < s.length(); i++) {
            if (s.charAt(i) != first) {
                second = s.charAt(i);
                break;
            }
        }
        if (second == '\0') {
            System.out.println("No");
            return;
        }
        boolean isAlternating = true;
        for (int i = 0; i < s.length(); i++) {
            char expected = (i % 2 == 0) ? first : second;
            if (s.charAt(i) != expected) {
                isAlternating = false;
                break;
            }
        }
        Set<Character> unique = new HashSet<>();
```

```
        for (char c : s.toCharArray()) {
            unique.add(c);
        }

        if (isAlternating && unique.size() == 2)
            System.out.println("Yes");
        else
            System.out.println("No");
    }
}
```

3.Recursion-Natural numbers

```
import java.io.*;
import java.util.*;

public class Main {
    static void printNumbers(int n) {
        if (n == 0)
            return;
        printNumbers(n - 1);
        System.out.print(n + " ");
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        if (n < 1 || n > 50) {
            System.out.println("Enter a Valid Input!!!!!!!!!!!!!!!!!!!!!!!!");
        } else {
            System.out.println("The first " + n + " Natural Numbers are:");
            printNumbers(n);
        }
```

```
    }
}
```

3.Recursion-sum of Numbers

```
import java.io.*;
import java.util.*;


public class Main {
    static int sumOfNumbers(int n) {
        if (n == 1)
            return 1;
        return n + sumOfNumbers(n - 1);
    }


    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        if (n < 1 || n > 50) {
            System.out.println("Enter a Valid Input!!!!!!!!!!!!!!!!!!");
        } else {
            int sum = sumOfNumbers(n);
            System.out.printf("The sum of numbers from 1 to %d : %04d", n, sum);
        }
    }
}
```

4.Recursion-Sum of digits

```
import java.io.*;
import java.util.*;




public class Main {
```

```
    static int sumOfDigits(int n) {
        if (n == 0)
            return 0;
        return (n % 10) + sumOfDigits(n / 10);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        if (n >= 1000) {
            System.out.println("Enter a Valid Input!!!!!!!");
        } else {
            int sum = sumOfDigits(n);
            System.out.printf("Sum of Digit:%04d", sum);
        }
    }
}
```

5. Recursion-Count the digit

```
import java.io.*;
import java.util.*;
import java.util.Scanner;

public class Main {
    static int countDigits(int num) {
        if (num == 0)
            return 0;
        return 1 + countDigits(num / 10);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
```

```
        int num = sc.nextInt();
        if (num == 0)
            System.out.println("The Count the digits: 1");
        else
            System.out.println("The Count the digits: " + countDigits(num));
    }
}
```

5.Armstrong number or not

```
import java.io.*;
import java.util.*;
import java.util.Scanner;

public class Main {

    public static boolean isArmstrong(int num) {
        int original = num;
        int digits = 0, sum = 0;

        int temp = num;
        while (temp > 0) {
            digits++;
            temp /= 10;
        }

        temp = num;
        while (temp > 0) {
            int rem = temp % 10;
            sum += Math.pow(rem, digits);
            temp /= 10;
        }
```

```
    return sum == original;
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int n = sc.nextInt();
    if (n < 0 || n >= 100000) {
        System.out.println("Invalid Input");
    } else {
        if (isArmstrong(n))
            System.out.println("yes");
        else
            System.out.println("no");
    }
}
```