Assignment 5: Recursion Based

Question - 1

Write a program to find the number of digits in a number using recursion.

Input Format

Input consists of a non-negative integer.

Constraints

NA

Output Format

Refer sample output for formatting specifications.

Sample Input

432

Sample Output

The number of digits in 432 is 3

Program:

```
import java.util.Scanner;
public class Main {
   public static int countDig(int n) {
      if (n==0) {
         return 0; }
      return 1 + countDig(n/ 10);    }
  public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int n = sc.nextInt();
      int d;
      if (n== 0) {
         d = 1;
```

```
} else {
        d= countDig(n);
      }
      System.out.println(n);
      System.out.println(d);   }}
```

Question - 2

Check if a String is Palindrome (Using Recursion)

Description: Use recursion to check whether a given string is a palindrome.

Input:
Input: madam
Output:
Output: Yes

Program:

```
import java.util.Scanner;
public class Main{
   public static boolean Pal(String s,int start,int end) {
      if (start >= end)
         return true;
      if (s.charAt(start) != s.charAt(end))
         return false;
      return Pal(s, start + 1, end - 1);
   }
 public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      String a = sc.nextLine();
```

```
    if (Pal(a,0,a.length()- 1))
      System.out.println("Yes");
    else
      System.out.println("No");    }}
```

Question 3

Calculate Power of a Number

Description: Write a recursive method to calculate a^b.

Input:
Input: 2 5
Output:
Output: 32

Program:

```
import java.util.Scanner;
public class Main {
  public static int pow(int a, int b) {
    if (b == 0)
      return 1;
    return a*pow(a, b - 1);
  }
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int a = sc.nextInt();
    int b = sc.nextInt();
    int res = pow(a, b);
    System.out.println(res);}}
```

Question – 4

Count Occurrences of a Character in String

Description: Use recursion to count how many times a character appears in a string.

Input:
Input: "programming", 'g'
Output:
Output: 2

Program:

```
import java.util.Scanner;

public class Main {
  public static int countChar(String s, char c, int i) {
    if (i == s.length()) {
      return 0;
    }
    if (s.charAt(i) == c) {
      return 1 + countChar(s, c, i + 1);
    } else {
      return countChar(s, c, i + 1);
    }
  }


  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    String s = sc.nextLine();
    char c = sc.next().charAt(0);
    int count = countChar(s, c, 0);
    System.out.println(count);}}
```