1. Count digits in a number - Try yourself

i/p : 43235

o/p : 5

code:

public class count\_digit\_in\_num {

    //Loop

    static void f(int n){

        int count =0;

        int x=n;

        while(x!=0){

            int rem =x%10;

            if(rem!=0){

                count++;

            }

            x=x/10;

        }

        System.out.println(count);

    }

    //Recursion

    public static int recursion(int n){

        if(n==0){

            return 0;

        }

        int count =0;

        int rem = n%10;

        if(rem!=0){

            count+=1;

        }

        return count+recursion(n/10);

        }

    public static void main(String[] args) {

        f(43235);

        int num = 4332235;

        System.out.println(recursion(num));

    }

}

2.Count zeros in a number - Try yourself

i/p : 430203005

o/p : 4

Code:

public class Count\_Zero\_in\_Number {

    static void loop(int n){

        int count = 0;

        while(n!=0){

            int rem = n%10;

            if(rem==0){

                count++;

            }

            n= n/10;

        }

        System.out.println(count);

    }

    static int recursion(int n){

        int count =0;

        if(n==0){

            return 0;

        }

        int rem = n%10;

        if(rem==0){

            count+=1;

        }

        return count+recursion(n/10);

    }

    public static void main(String[] args) {

        loop(1230000321);

        System.out.println(recursion(210100000));

    }

}

3.Check the given number is Prime Number or not.

Code:

public class GivenNumber\_primeorNot {

    static void loop(int n){

        int x=n;

        if(x<=1){

           System.out.println(" Not Prime");

        }

        for(int i=2;i<=x;i++){

            if(x%i==0){

                System.out.print(n + " is not Prime Number");

                break;

            }

            else{

                System.out.println(n + " Is Prime");

                break;

            }

        }

        System.out.println();

    }

   public static void recursion(int n , int i){

       if (n < 2) {

           System.out.println(n +" Not Prime");

           return;

       }

       if (n == 2) {

           System.out.println(n + " is prime");

           return;}

       if (n % i == 0) {

           System.out.println(n +" Not Prime");

           return;

       }

       if (i \* i > n) {

           System.out.println(n +" Prime");

           return;

       }

       recursion(n, i + 1);

   }

    public static void main(String[] args) {

        loop(19);

        int number1 = 19;

        recursion(number1, 2);

    }

}

4.Generate all prime numbers in a given range.

Code:

public class allprimenumber {

    static void loop(int min , int max){

        int flag =0;

        if(min<=1){

            min=2;

        }

        for(int i=min ; i<=max;i++){

            for(int j=2;j<i;j++){

                if(i%j==0){

                    flag=1;

                    break;

                }

            }

            if(min<=1){

                flag=1;

            }

            if(flag==0){

                System.out.println(i+ " Prime Number");

            }

            flag=0;

        }

    }

    // Using Recursion

    public static void findrange(int s , int e){

        if(s>e){

            return;

        }

        if(isPrime(s)){

            System.out.println(s);

        }

        findrange(s+1,e);

    }

    public static boolean isPrime(int n){

        return isPrimeHelp(n,2);

    }

    public static boolean isPrimeHelp(int n , int i){

        if (n <= 2) {

            return (n == 2);

        }

        if (n % i == 0) {

            return false;

        }

        if (i \* i > n) {

            return true;

        }

        return isPrimeHelp(n, i + 1);

    }

    public static void main(String[] args) {

        loop(0,10);

        int s=1;

        int e= 10;

        System.out.println("Prime number between " + s + " and " + e + " is " + ":" );

        findrange(s,e);

    }

}

5.Check the given number is armstrong or not.

Code:

public class Armstrong\_or\_Not {

    static void loop(int n){

        int sum = 0;

        int x = n;

        while(x!=0){

            int rem = x%10;

            sum = sum + rem\*rem\*rem;

            x=x/10;

        }

        if(sum==n){

            System.out.println("Armstrong");

        }

        else{

            System.out.println("Not Armstrong");

        }

    }

    // Recursion

    public static int recursion(int n){

        int sum=0;

        if(n==0){

            return 0;

        }

        else{

            int rem = n%10;

            return (rem\*rem\*rem)+ recursion(n/10);

        }

    }

    public static void main(String[] args) {

        loop(152);

        int n =153;

        if(recursion(n)==n){

            System.out.println("Armstrong");

        }

        else{

            System.out.println("Not Armstrong");

        }

    }

}

6.Check if a string is palindrome or not.

Code:

public class is\_palindrom {

    public static void loop(String s){

        int j=s.length()-1;

        boolean ispal = true;

        for(int i=0;i<j;i++){

            if(s.charAt(i)!=s.charAt(j)){

                ispal=false;

               break;

            }

            j--;

        }

            if(ispal){

                System.out.println("Palindrom");

            }

            else{

                System.out.println("not palindrom");

            }

        }

    //Recursion

    public static boolean isPalindrome(String str) {

        return isPalindromeHelper(str, 0, str.length() - 1);

    }

    public static boolean isPalindromeHelper(String str, int start, int end) {

        if (start >= end) {

            return true;  // Base case, the substring has been checked

        }

        if (str.charAt(start) != str.charAt(end)) {

            return false;  // Characters at the corresponding positions are not equal

        }

        return isPalindromeHelper(str, start + 1, end - 1);

    }

    public static void main(String[] args) {

        loop("RohrthoR");

        String str = "aba";

        if (isPalindrome(str)) {

            System.out.println("Palindrome");

        } else {

            System.out.println("Not Palindrome");

        }

    }

}