Control statements and Patterns

ASSIGNMENT-1:

1)//Take an integer **A** as input, you have to tell whether it is a prime number or not.

```
import java.util.Scanner;
public class PrimeExample{
public static void main(String args[]){
  int i,m=0,flag=0;

  Scanner sc=new Scanner(System.in);
  int n=sc.nextInt();
  m=n/2;
  if(n==0||n==1){
  System.out.println(n+" is not prime number");
  }else{
  for(i=2;i<=m;i++){
    if(n%i==0){
    System.out.println(n+" is not prime number");
    flag=1;
    break;
    }
  }
  if(flag==0)  {
System.out.println(n+" is prime number"); }
  }//end of else
}
}
```

2)//Write a program to input two integers **A & B** from user and print their HCF.

```
import java.util.Scanner;
public class GCDOfTwoNumbers {
   public static void main(String args[]){
      int a, b, i, hcf = 0;
      Scanner sc = new Scanner(System.in);
```

```
    System.out.println("Enter first number :: ");
    a = sc.nextInt();
    System.out.println("Enter second number :: ");
    b = sc.nextInt();

    for(i = 1; i <= a || i <= b; i++) {
      if( a%i == 0 && b%i == 0 )
      hcf = i;
    }
    System.out.println("HCF of given two numbers is ::"+hcf);
  }
}
```

3)//You are given an integer **N** you need to print all the Armstrong Numbers between **1** to **N**. (N inclusive).

```
import java.util.Scanner;
public class ArmstrongSeries {
   public static void main(String[] args) {
      double N;
      int i;
      Scanner scanner;
      scanner = new Scanner(System.in);
      System.out.println("Enter a Number");
      N = scanner.nextFloat();
      System.out.println("Armstrong Number between 0 to " + (int) N);

      for (i = 1; i < N; i++) {
         if (isArmstrongNumber(i)) {
            System.out.println(i + " ");
         }
      }
   }
   public static boolean isArmstrongNumber(int num) {
      int sum = 0, rightDigit, temp;
      temp = num;
      while (temp != 0) {
```

```
            rightDigit = temp % 10;
            sum = sum + (rightDigit * rightDigit * rightDigit);
            temp = temp / 10;
        }
        if (sum == num) {
            // N is armstrong number
            return true;
        } else {
            // N is not an armstrong number
            return false;
        }
    }
}
```

4)//You are given an **integer A** as input and you need to determine whether it is a **palindrome** or **not**.A **palindrome integer**

```
public int checkPalindrome(int original) {
  int reverseNum = 0;
  int tempOriginal = original;

  while (tempOriginal > 0) {

    int lastDigit = tempOriginal % 10;
    reverseNum = reverseNum * 10 + lastDigit;
    tempOriginal = tempOriginal / 10;
  }

  if (original == reverseNum) {
    return 1;
  } else {
    return 0;
  }
```

5)//Implement a program that takes two positive integers **A** and **B** in the input and prints their LCM.

```
import java.util.Scanner;
public class Main {
  public static void main(String[] args) {

    int  gcd = 1;
    Scanner sc=new Scanner(System.in);
    int n1=sc.nextInt();
    int n2=sc.nextInt();

    for(int i = 1; i <= n1 && i <= n2; ++i) {
      // Checks if i is factor of both integers
      if(n1 % i == 0 && n2 % i == 0)
        gcd = i;
    }

    int lcm = (n1 * n2) / gcd;
    System.out.printf("The LCM of %d and %d is %d.", n1, n2, lcm);
  }
}

}
```

6)//Take an integer **N** as input and print the count of its factors.**The factor** of a number is the number that divides it perfectly leaving no remainder

```
import java.util.*;
public class factors {
    public static void main(String[] args){
      Scanner in = new Scanner(System.in);
      System.out.print("Input an integer: ");
      int x = in.nextInt();

System.out.println(result(x));
}
public static int result(int num) {
int ctr = 0;
        for(int i=1; i<=(int)Math.sqrt(num); i++) {
```

```
            if(num%i==0 && i*i!=num) {
                ctr+=2;
            } else if (i*i==num) {
                ctr++;
            }
        }
        return ctr;
}
}
```

Assignment-2

1)//Write a program to print all Natural numbers from **1** to **N** where you have to take N as input from user

```
import java.util.Scanner;

public class Print_1_To_N_UsingWhile
{
public static void main(String[] args)
{
int i =1;

Scanner Sc = new Scanner(System.in);

System.out.print("Enter the value n : ");

int n = Sc.nextInt();

System.out.println("Numbers are : ");

while(i<=n)
{
System.out.println(i);
i++;
}
}
}
```

2)//Write a program to print all odd numbers from **1** to **N** where you have to take N as input from user. Here **N** is inclusive.

```
import java.util.Scanner;
public class DisplayOddNumbers {
    public static void main(String[] args) {
      Scanner sc=new Scanner(System.in);
      int n=sc.nextInt();
        //print all odd numbers <=n
        int odd=1;
        while (odd<=n) {
            System.out.print(odd+"  ");
            odd += 2;
        }
    }
}
```

3)//Write a program to find sum all Natural numbers from **1** to **N** where you have to take N as input from user.

```
import java.util.Scanner;
class sum {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int i=1;
        int sum=0;
        while(i<=n)
        {
            sum=sum+i;
            i++;

        }
        System.out.println(sum);
    }
}
```

4)//Write a program to print the multiplication table of the number entered by the user, **N**.

```
import java.util.Scanner;
class HelloWorld {
    public static void main(String[] args) {
Scanner sc=new Scanner(System.in);
int num=sc.nextInt();
int i=1;
while(i<=num)
{
    System.out.printf("%d * %d = %d \n", num, i, num * i);
            i++;
}
    }
}
```

5)//You are given two integers **A** and **B**. You have to find the value of $\mathbf{A^B}$.

```
import java.util.Scanner;
public class PowerOfNumber
{
public static void main(String args[])
{
int base, exponent, expo;
Scanner sc=new Scanner(System.in);
System.out.print("Enter the base: ");
base=sc.nextInt();
System.out.print("Enter the exponent: ");
exponent=sc.nextInt();
expo=exponent;
long power = 1;
//executes until the condition becomes false
while (exponent != 0)
{
```

```
//calculates power
power = power * base;
//decrements the power by 1
--exponent;
}
//prints result
System.out.println(base +" to the power " +expo + " is: "+power);
}
}
```

6) //Take an integer **A** as input. You have to print the sum of all odd numbers in the range **[1, A]**.

```
import java.util.Scanner;
class Sum_Odd_Number
{
public static void main(String[] args)
{
Scanner input = new Scanner(System.in);
System.out.print("Enter The Number of Limit : ");
int l =input.nextInt();
int sum = 0;
for(int s=1;s<=l;s++)
{
if(s%2==1)
sum = sum + s;
}
System.out.println("Sum of Odd Numbers :"+sum);
}
}
```