**NAME: TAKANKHAR SHUBHAM**

**CLASS: SY MCA**

**ROLL NO: 54**

**LAB ASSIGNMENT 3 :** Implement Producer Consumer Problem using multithreading

Create an Application to display the table of different numbers using multithreading.

Create an Application to display even and odd numbers till number X input by user, Using multithreading.

**Implement Producer Consumer Problem using multithreading**

**<<SOURCE\_CODE>>**

public class counter{

    int count=0;

    synchronized void producer(){

        if (count < 2 ){

        count++;

        System.out.println("produced count="+count);

        }

    }

    synchronized void consumer(){

        if ( count > 0){

        count--;

        System.out.println("consumed count="+count);

        }

    }

    public static void main(String args[]){

        counter c = new counter();

        Thread t1 = new Thread(new Runnable(){

            @Override

            public void run(){

                c.producer();

                c.producer();

            }

        });

        t1.start();

        Thread t2 = new Thread(new Runnable(){

            @Override

            public void run(){

                c.consumer();

            }

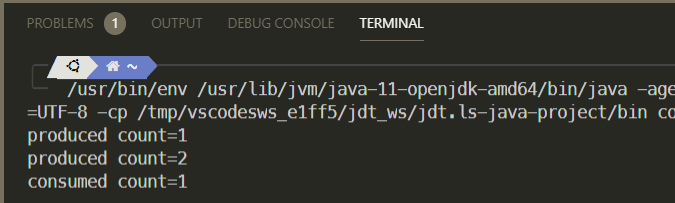
        });

        t2.start();

            }

}

**<<OUTPUT>>**



**Create an Application to display the table of different numbers using multithreading.**

**<<SOURCE\_CODE>>**

import java.util.Scanner;

class multiTables extends Thread{

    int x;

    multiTables(int num){

        x=num;

    }

    public void run(){

        printTable(x);

    }

    synchronized public void printTable(int x){

        for(int i=1;i<=10;i++){

        System.out.println(x+"x"+i+"="+x\*i);

        }

    }

    public static void main(String args[]){

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter First Table to print");

        int num1 = sc.nextInt();

        System.out.println("Enter Second Table to print:");

        int num2 = sc.nextInt();

        sc.close();

        multiTables t1 =new multiTables(num1);

        multiTables t2 =new multiTables(num2);

        t1.start();

        try{

              t1.join();

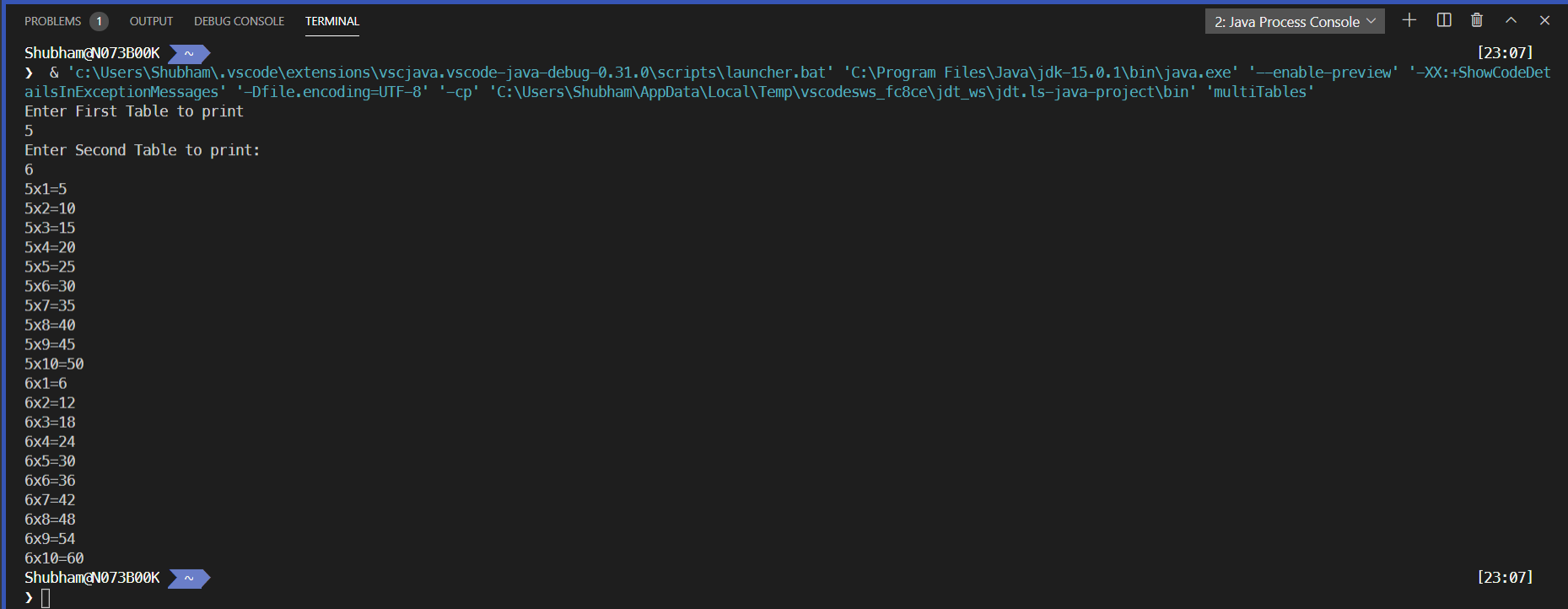
               }catch(Exception e){System.out.println(e);}

        t2.start();

    }

}

**<<OUTPUT>>**

****

**Create an Application to display even and odd numbers till number X input by user, Using multithreading.**

**<<SOURCE\_CODE>>**

import java.util.Scanner;

public class evenOdd {

    void findEvens(int x) {

        System.out.print("Even Nos:");

        for (int i = 1; i <= x; i++) {

            if (i % 2 == 0) {

                System.out.print(i + " ");

            }

        }

        System.out.println();

    }

    void findOdds(int x) {

        System.out.print("Odd Nos:");

        for (int i = 1; i <= x; i++) {

            if (i % 2 != 0) {

                System.out.print(i + " ");

            }

        }

        System.out.println();

    }

    public static void main(String args[]) throws InterruptedException {

        System.out.print("Enter Range 1-");

        Scanner sc = new Scanner(System.in);

        int num = sc.nextInt();

        evenOdd obj = new evenOdd();

        threadOne even = new threadOne(obj, num);

        even.start();

        threadTwo odd = new threadTwo(obj, num);

        odd.start();

        sc.close();

    }

}

class threadOne extends Thread {

    evenOdd e;

    int x;

    threadOne(evenOdd e, int x) {

        this.e = e;

        this.x = x;

    }

    public void run() {

        e.findEvens(x);

    }

}

class threadTwo extends Thread {

    evenOdd o;

    int x;

    threadTwo(evenOdd o, int x) {

        this.o = o;

        this.x = x;

    }

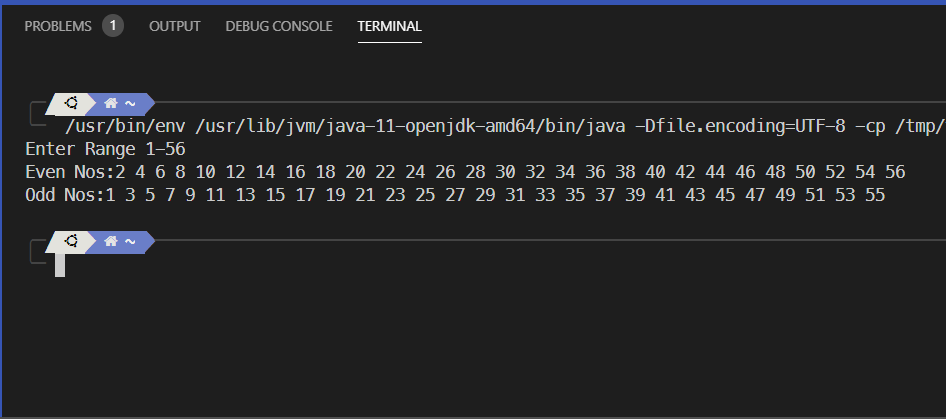
    public void run() {

        o.findOdds(x);

    }

}

**<<OUTPUT>>**

****