**Lab – 6**

**Subject: NIS**

**Aim:** Write a Program to generate the Points on the Elliptical Curve E\_p(a,b)

**Program: -**

import java.util.\*;

import java.lang.\*;

public class ecc

{

    public static boolean isPerfactSquare(long n){

        double sqrt = Math.sqrt((double)n);

        return (sqrt - Math.floor(sqrt)==0);

    }

    public static long pow(long a,long b){

        if(b == 0){

            return 1;

        }

        else{

            return a \* pow(a,--b);

        }

    }

    public static long isCongruant(long a , long n){

        if((a+1) % n == 0){

            return -1;

        }

        else

        {

            return 1;

        }

    }

    public static long positiveInvers(long inverse,long n){

         while(inverse < 0){

            inverse = inverse + n;

         }

        return inverse;

     }

    public static void elipticalCurve(long a,long b,long p)

    {

        long x = 0;

        long w = 0;

        while(x<p){

            w = (pow(x,3) + a\*x + b) % p;

            long temp = pow(w,((p-1)/2)) % p;

            if(isCongruant(temp,p) == -1){

                System.out.println("No Solution For : " + x);

                x++;

                continue;

            }

            if(isCongruant(temp,p) == 1){

                while(!isPerfactSquare(w)){

                    if((p\*p) <= w){

                        break;

                    }

                    w =w + p;

                }

                w = (long)Math.sqrt(w);

                long pointA = ~(w-1);

                pointA = positiveInvers(pointA,p);

                System.out.println("( " + x +" , " + pointA +")");

                System.out.println("( " + x +" , " + w +")");

            }

            x++;

        }

    }

    public static void main(String[] args) {

        int i=0;

        long a=0,b=0;

        while(true){

            if(((int)Math.pow(i,3)\*4 + 27\*(int)Math.pow(i,2))!=0){

                a=i;

                b=i;

                break;

            }

            i++;

        }

        System.out.println("a is : " + a +" b is : " + b);

        long p = 13;

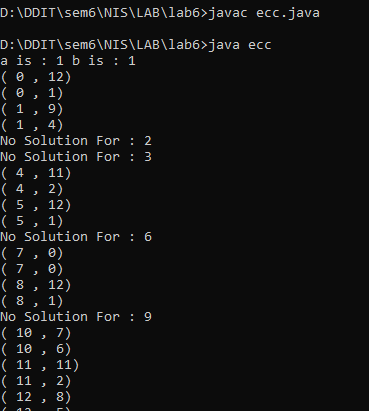
        elipticalCurve(a,b,p);

    }

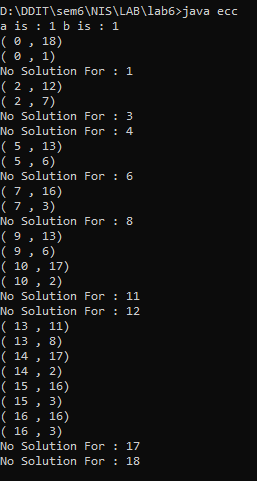
}

**Output: -**

**P=13**

****

**P = 17**

****