**Experiment No.03**

Name : Sanskruti Shinde. (139)

**Aim :** Study and implementation of the Restoring Division Algorithm.

**Code :**

import java.util.Scanner;

public class RestoringDiv {

static String toBinary(int num, int n) {

int mask = (1 << n) - 1;

num &= mask;

StringBuilder sb = new StringBuilder();

for (int i = n - 1; i >= 0; i--) sb.append((num >> i) & 1);

return sb.toString();

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the Divisor (M) = ");

int M = sc.nextInt();

System.out.print("Enter the Dividend (Q) = ");

int Q = sc.nextInt();

int n = (int)Math.ceil(Math.log(Math.max(Math.abs(M), Math.abs(Q)) + 1) / Math.log(2));

if (n < 2) n = 2;

int A = 0, quotient = 0;

for (int i = n - 1; i >= 0; i--) {

A = (A << 1) | ((Q >> i) & 1);

A = A - M;

if (A < 0) {

quotient = (quotient << 1);

A = A + M;

quotient = (quotient << 1) | 1;

}

}

System.out.println("\nBinary representation of Dividend (Q) = " + toBinary(Q, n));

System.out.println("Binary representation of Divisor (M) = " + toBinary(M, n));

System.out.println("Quotient in binary = " + toBinary(quotient, n));

System.out.println("Remainder in binary = " + toBinary(A, n));

sc.close();

}

**}**

**Output :**

