**INPUT:**

package expt4;

import java.util.Scanner;

public class CRCChecker {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        //Sender Side

        System.out.print("Enter the binary data: ");

        String binaryData = scanner.nextLine();

        System.out.print("Enter the generator g(x): ");

        String generator = scanner.nextLine();

        //Validate binary input

        if (!binaryData.matches("[01]+") || !generator.matches("[01]+")) {

            System.out.println("Invalid input! Please enter binary values only.");

        }

        int generatorLength = generator.length();

        String appendedData = binaryData + "0".repeat(generatorLength - 1);

        System.out.println("\nAppended Data: " + appendedData);

        String[] senderResult = binaryDivision(appendedData, generator);

        String remainder = senderResult[1];

        String transmittedCodeword = binaryData + remainder;

        System.out.println("\nFinal Remainder at Sender: " + remainder);

        System.out.println("Transmitted Codeword: " + transmittedCodeword);

        //Receiver Side

        System.out.print("\nEnter the received codeword: ");

        String receivedCodeword = scanner.nextLine();

        System.out.print("Enter the generator g(x) at receiver: ");

        String receiverGenerator = scanner.nextLine();

        //DO NOT append zeros at receiver side

        String[] receiverResult = binaryDivision(receivedCodeword, receiverGenerator);

        String receiverRemainder = receiverResult[1];

        System.out.println("\nFinal Remainder at Receiver: " + receiverRemainder);

        if (receiverRemainder.matches("0+")) {

            System.out.println("No error detected.");

        } else {

            System.out.println("Error detected. Discarding codeword.");

        }

        scanner.close();

    }

    //Performs binary division (CRC division)

    public static String[] binaryDivision(String dividend, String divisor) {

        int dividendLen = dividend.length();

        int divisorLen = divisor.length();

        StringBuilder remainder = new StringBuilder(dividend.substring(0, divisorLen));

        StringBuilder quotient = new StringBuilder();

        for (int i = divisorLen; i <= dividendLen; i++) {

            if (remainder.charAt(0) == '1') {

                remainder = xor(remainder.toString(), divisor);

                quotient.append('1');

            } else {

                quotient.append('0');

            }

            if (i < dividendLen) {

                remainder.append(dividend.charAt(i));

            }

            remainder.deleteCharAt(0); // shift left

        }

        return new String[]{quotient.toString(), remainder.toString()};

    }

    //XOR operation between two binary strings

    public static StringBuilder xor(String a, String b) {

        StringBuilder result = new StringBuilder();

        for (int i = 0; i < b.length(); i++) {

            result.append(a.charAt(i) == b.charAt(i) ? '0' : '1');

        }

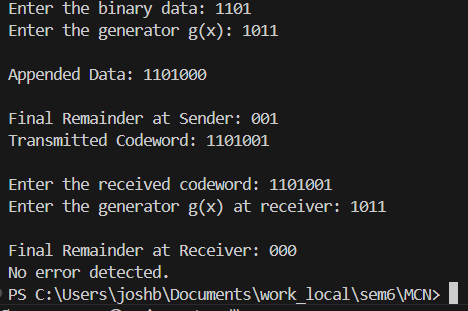
        return result;

    }

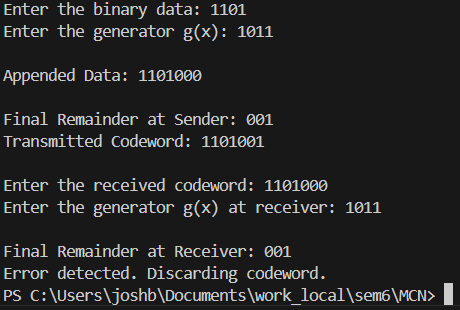
}

**OUTPUT:**

1. **Without error:**

****

1. **With error (flip a bit):**

****