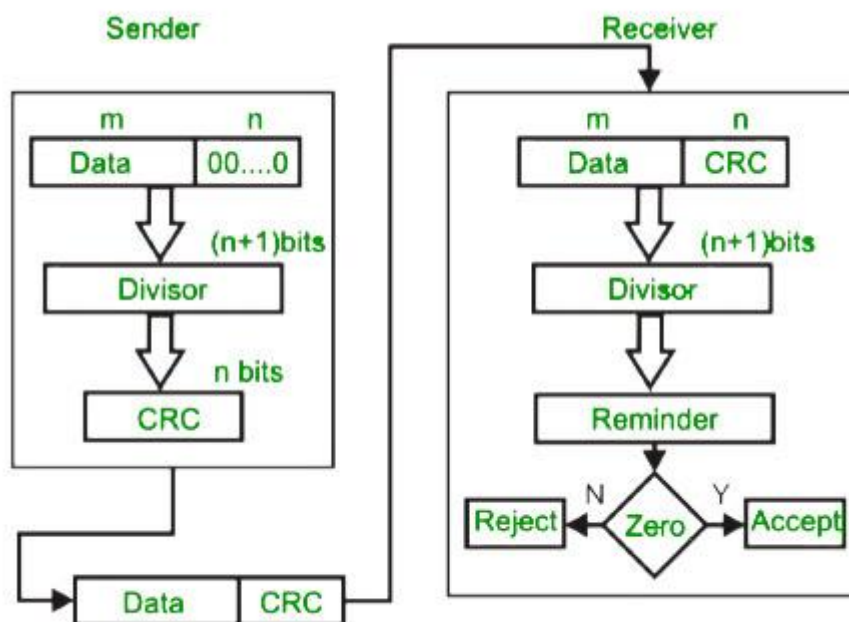


**EX.NO 10****Simulation of Error Correction Code(CRC)****AIM :**

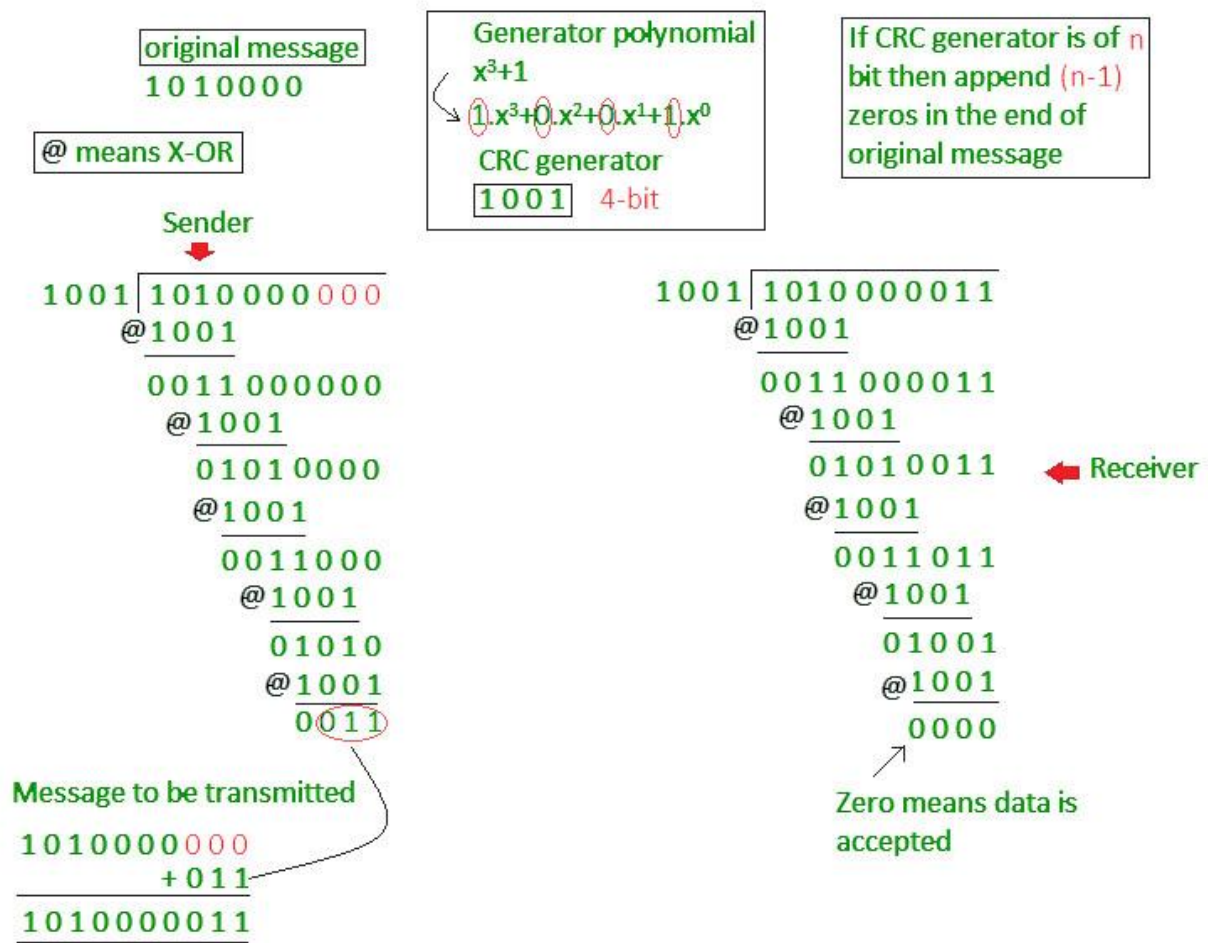
To Simulation of Error Correction Code using Java.

**Theory:****Cyclic redundancy check (CRC)**

- Unlike checksum scheme, which is based on addition, CRC is based on binary division.
- In CRC, a sequence of redundant bits, called cyclic redundancy check bits, are appended to the end of data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number.
- At the destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit is assumed to be correct and is therefore accepted.
- A remainder indicates that the data unit has been damaged in transit and therefore must be rejected.



### Example :



:

### Procedure:

1. Open the editor and type the program for error detection
2. Get the input in the form of bits.
3. Append the redundancy bits.
4. Divide the appended data using a divisor polynomial.
5. The resulting data should be transmitted to the receiver.
6. At the receiver the received data is entered.
7. The same process is repeated at the receiver.
8. If the remainder is zero there is no error otherwise there is some error in the received bits
9. Run the program.

### Program:

```
import java.io.*;

class CRC
{
    public static void main(String args[]) throws IOException
```

```

{
    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
    System.out.println("Enter Generator:");
    String gen = br.readLine();
    System.out.println("Enter Data:");
    String data = br.readLine();
    String code = data;
    while(code.length() < (data.length() + gen.length() - 1))
        code = code + "0";
    code = data + div(code,gen);
    System.out.println("The transmitted Code Word is: " + code);
    System.out.println("Please enter the received Code Word: ");
    String rec = br.readLine();
    if(Integer.parseInt(div(rec,gen)) == 0)
        System.out.println("The received code word contains no errors.");
    else
        System.out.println("The received code word contains errors.");
}

```

```

static String div(String num1,String num2)
{
    int pointer = num2.length();
    String result = num1.substring(0, pointer);
    String remainder = "";
    for(int i = 0; i < num2.length(); i++)
    {
        if(result.charAt(i) == num2.charAt(i))
            remainder += "0";
        else
            remainder += "1";
    }
}

```

```

    }
    while(pointer < num1.length())
    {
        if(remainder.charAt(0) == '0')
        {
            remainder = remainder.substring(1, remainder.length());
            remainder = remainder + String.valueOf(num1.charAt(pointer));
            pointer++;
        }
        result = remainder;
        remainder = "";
        for(int i = 0; i < num2.length(); i++)
        {
            if(result.charAt(i) == num2.charAt(i))
                remainder += "0";
            else
                remainder += "1";
        }
    }
    return remainder.substring(1,remainder.length());
}
}

```

**Result:**

Thus the error detection and error correction is implemented successfully