Aim:

To write a Java program to simulate Error Correction Code (CRC) in Java

Algorithm:

1.Define a function that does bit to bit Xor operation .Concatenate the bits into one Single String

2.Define another function to Divide the Polynomial with the CRC Polynomial and Obtain the Remainder using repeated Division .Ignore the leading zeros while division.

3.Take a Transmission String and CRC Polynomial. Divide and obtain CRC Remainder

4.Pad the CRC Remainder to the Transmission String and Again Divide Using CRC Polynomial If no Error the is Transmitted properly Function returns True

5.Introduce another Error to Transmission String and Divide Using CRC Polynomial .This time the isTransmittedProperly Function should return false

Code:

CRC.java:

```
package Exercise8;

public class CRC {
	public  String Xor(String s1,String s2) {
		String res="";
		for(int i=0;i<s2.length();i++) {
			if(s1.charAt(i)==s2.charAt(i)) {
				res+='0';
			}
			else {
				res+='1';
			}
		}
		return res;
	}
```

```
public String  padBit(String s1,String s2) {
        int d2=s2.length();
        for (int i=0;i<d2-1;i++) {
                s1+='0';
        }
        return s1;
}

public String Divide(String s1,String s2) {
        int d1=s1.length();
        String tmp="";
        int track=0;
        String tmp2="";
        tmp=Xor(s1.substring(0, s2.length()),s2);
        track+=tmp.length();
        while(track<d1) {
                        if(tmp.charAt(0)=='0') {
                                tmp=tmp.substring(1);
                                tmp+=String.valueOf(s1.charAt(track++));
                        }

                tmp2=tmp;
                tmp="";
                tmp=Xor(tmp2,s2);
        }

        return tmp2;

}
```

```
	public boolean isTransmittedProperly(String originalBits,String CRCRemainder,String divisor)
{


		String s2=originalBits+CRCRemainder.substring(1);


		if(new CRC().Divide(s2, divisor).equals("0000")) {

			return true;

		}

		return false;

	}

	public static void main(String [] args) {

		String s1="10111011";

		System.out.println("No Error While Transmission: ");

		System.out.println(" Original BitArrangement:"+s1);


		String chk=s1;

		String s2="1001";

		System.out.println(" CRC Polynomial:"+s2);

		s1=new CRC().padBit(s1,s2);

		String s3=new CRC().Divide(s1, s2);

		chk+=s3.substring(1);

		/*Check for proper Transmission*/

		System.out.println("CRC Match :->"+new
CRC().isTransmittedProperly("10111011","0110","1001"));

		/*Flip a bit in the original bit to simulate Error*/

		System.out.println("\n\nError Introduced into Bits:\n ErrenousBit
Arrangement:10111111\n CRC Polynomial:1001\n");

		System.out.println("CRC Match :->"+new
CRC().isTransmittedProperly("10111111","0110","1001"));


	}

}
```

Output:

```
No Error While Transmission:
 Original BitArrangement:10111011
 CRC Polynomial:1001
CRC Match :->true


Error Introduced into Bits:
 ErrenousBit Arrangement:10111111
 CRC Polynomial:1001

CRC Match :->false
```

Result:

Thus the CRC Error Correction/Detection Algorithm was implemented in Java