Assignment-7

//Cyclic Redundancy Check (CRC) for error detection:

```
import java.util.*;

class Main {
	public static void main(String args[]) {
		Scanner scan = new Scanner(System.in);
		int n;

		System.out.println("Enter the size of the data:");
		n = scan.nextInt();
		int data[] = new int[n];
		System.out.println("Enter the data, bit by bit:");
		for(int i=0 ; i < n ; i++) {
			System.out.println("Enter bit number " + (n-i) + ":");
			data[i] = scan.nextInt();
		}
		System.out.println("Enter the size of the divisor:");
		n = scan.nextInt();
		int divisor[] = new int[n];
		System.out.println("Enter the divisor, bit by bit:");
		for(int i=0 ; i < n ; i++) {
			System.out.println("Enter bit number " + (n-i) + ":");
			divisor[i] = scan.nextInt();
		}
		int remainder[] = divide(data, divisor);
		for(int i=0 ; i < remainder.length-1 ; i++) {
			System.out.print(remainder[i]);
		}
		System.out.println("\nThe CRC code generated is:");

		for(int i=0 ; i < data.length ; i++) {
			System.out.print(data[i]);
		}
		for(int i=0 ; i < remainder.length-1 ; i++) {
			System.out.print(remainder[i]);
		}
		System.out.println();
		int sent_data[] = new int[data.length + remainder.length - 1];
		System.out.println("Enter the data to be sent:");
		for(int i=0 ; i < sent_data.length ; i++) {
			System.out.println("Enter bit number " + (sent_data.length-i)
								+ ":");
			sent_data[i] = scan.nextInt();
		}
		receive(sent_data, divisor);
	}
```

```
	static int[] divide(int old_data[], int divisor[]) {
		int remainder[] , i;
		int data[] = new int[old_data.length + divisor.length];
		System.arraycopy(old_data, 0, data, 0, old_data.length);
		remainder = new int[divisor.length];
		System.arraycopy(data, 0, remainder, 0, divisor.length);
		for(i=0 ; i < old_data.length ; i++) {
			System.out.println((i+1) + ".) First data bit is : "
								+ remainder[0]);
			System.out.print("Remainder : ");
			if(remainder[0] == 1) {
				for(int j=1 ; j < divisor.length ; j++) {
					remainder[j-1] = exor(remainder[j], divisor[j]);
					System.out.print(remainder[j-1]);
				}
			}
			else {
				for(int j=1 ; j < divisor.length ; j++) {
					remainder[j-1] = exor(remainder[j], 0);
					System.out.print(remainder[j-1]);
				}
			}
			remainder[divisor.length-1] = data[i+divisor.length];
			System.out.println(remainder[divisor.length-1]);
		}
		return remainder;
	}

	static int exor(int a, int b) {
		if(a == b) {
			return 0;
		}
		return 1;
	}

	static void receive(int data[], int divisor[]) {
		int remainder[] = divide(data, divisor);
		for(int i=0 ; i < remainder.length ; i++) {
			if(remainder[i] != 0) {
				System.out.println("There is an error in received
data...");
				return;
			}
		}
		System.out.println("Data was received without any error.");
	}
}
```

//Output:

```
1^Clinuxmint@jc623:~/Desktop/CNLAB/Ass7$ javac Main.java
linuxmint@jc623:~/Desktop/CNLAB/Ass7$ java Main
Enter the size of the data:
14
Enter the data, bit by bit:
Enter bit number 14:
1
Enter bit number 13:
1
Enter bit number 12:
0
Enter bit number 11:
1
Enter bit number 10:
0
Enter bit number 9:
1
Enter bit number 8:
1
Enter bit number 7:
0
Enter bit number 6:
1
Enter bit number 5:
1
Enter bit number 4:
0
Enter bit number 3:
0
Enter bit number 2:
0
Enter bit number 1:
0
Enter the size of the divisor:
5
Enter the divisor, bit by bit:
Enter bit number 5:
1
Enter bit number 4:
0
Enter bit number 3:
0
Enter bit number 2:
1
Enter bit number 1:
1
1.) First data bit is : 1
Remainder : 10011
2.) First data bit is : 1
Remainder : 00001
3.) First data bit is : 0
Remainder : 00010
4.) First data bit is : 0
Remainder : 00101
5.) First data bit is : 0
Remainder : 01011
6.) First data bit is : 0
Remainder : 10110
7.) First data bit is : 1
Remainder : 01010
8.) First data bit is : 0
Remainder : 10100
9.) First data bit is : 1
Remainder : 01110
10.) First data bit is : 0
Remainder : 11100
11.) First data bit is : 1
Remainder : 11110
12.) First data bit is : 1
Remainder : 11010
```

```
13.) First data bit is : 1
Remainder : 10010
14.) First data bit is : 1
Remainder : 00010
0001
The CRC code generated is:
110101101100000001
Enter the data to be sent:
Enter bit number 18:
1
Enter bit number 17:
1
Enter bit number 16:
0
Enter bit number 15:
1
Enter bit number 14:
0
Enter bit number 13:
1
Enter bit number 12:
1
Enter bit number 11:
0
Enter bit number 10:
1
Enter bit number 9:
1
Enter bit number 8:
0
Enter bit number 7:
0
Enter bit number 6:
0
Enter bit number 5:
0
Enter bit number 4:
0
Enter bit number 3:
0
Enter bit number 2:
0
Enter bit number 1:
1
1.) First data bit is : 1
Remainder : 10011
2.) First data bit is : 1
Remainder : 00001
3.) First data bit is : 0
Remainder : 00010
4.) First data bit is : 0
Remainder : 00101
5.) First data bit is : 0
Remainder : 01011
6.) First data bit is : 0
Remainder : 10110
7.) First data bit is : 1
Remainder : 01010
8.) First data bit is : 0
Remainder : 10100
9.) First data bit is : 1
Remainder : 01110
10.) First data bit is : 0
Remainder : 11100
11.) First data bit is : 1
Remainder : 11110
12.) First data bit is : 1
Remainder : 11010
13.) First data bit is : 1
Remainder : 10011
14.) First data bit is : 1
Remainder : 00000
15.) First data bit is : 0
Remainder : 00000
16.) First data bit is : 0
Remainder : 00000
17.) First data bit is : 0
Remainder : 00000
18.) First data bit is : 0
Remainder : 00000
Data was received without any error.
```