

## **1 Write a program for error detecting code using CRC-CCITT (16- bits).**

```
import java.util.*;

public class CRC {

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

        System.out.print("Enter the size of the data: ");
        int n = scan.nextInt();
        int[] data = new int[n];
        System.out.println("Enter the data bits (space separated):");
        for (int i = 0; i < n; i++) {
            data[i] = scan.nextInt();
            if (data[i] != 0 && data[i] != 1) data[i] = 0;
        }

        System.out.print("Enter the size of the divisor: ");
        int m = scan.nextInt();
        if (m < 2) {
            System.out.println("Divisor must be at least 2 bits long.");
            scan.close();
            return;
        }
        int[] divisor = new int[m];
        System.out.println("Enter the divisor bits (space separated):");
        for (int i = 0; i < m; i++) divisor[i] = scan.nextInt();

        // Prepare padded data for generating CRC: data + (m-1) zeros
        int[] dataPadded = new int[data.length + m - 1];
        System.arraycopy(data, 0, dataPadded, 0, data.length);

        // Compute remainder (CRC)
        int[] remainder = mod2Division(dataPadded, divisor);

        System.out.print("Remainder: ");
        for (int b : remainder) System.out.print(b);
        System.out.println();
    }
}
```

```

// Form the codeword (data + remainder)
int[] codeword = new int[data.length + remainder.length];
System.arraycopy(data, 0, codeword, 0, data.length);
System.arraycopy(remainder, 0, codeword, data.length, remainder.length);

System.out.print("The CRC code generated is: ");
for (int b : codeword) System.out.print(b);
System.out.println();

// Ask user for received codeword or simulate correct transmission
System.out.println("Enter the received codeword bits (space separated) OR press
Enter to simulate correct transmission:");
scan.nextLine(); // consume endline
String line = scan.nextLine().trim();

int[] received;
if (line.isEmpty()) {
    received = codeword; // simulate perfect transmission
    System.out.print("Simulating correct transmission: ");
    for (int b : received) System.out.print(b);
    System.out.println();
} else {
    String[] toks = line.split("\\s+");
    if (toks.length != codeword.length) {
        System.out.println("Received codeword length mismatch. Expected length: " +
codeword.length);
        scan.close();
        return;
    }
    received = new int[toks.length];
    for (int i = 0; i < toks.length; i++) received[i] = Integer.parseInt(toks[i]);
}

int[] recvRemainder = mod2Division(received, divisor);
boolean error = false;
for (int b : recvRemainder) if (b != 0) { error = true; break; }

if (error) System.out.println("There is an error in received data... ");
else System.out.println("Data was received without any error.");

```

```

        scan.close();
    }

    // Performs mod-2 division on 'data' using 'divisor' and returns remainder (length =
    divisor.length - 1)
    static int[] mod2Division(int[] data, int[] divisor) {
        int[] temp = Arrays.copyOf(data, data.length);
        int n = data.length;
        int m = divisor.length;

        // For each bit position where the divisor fits
        for (int i = 0; i <= n - m; i++) {
            if (temp[i] == 1) {
                for (int j = 0; j < m; j++) {
                    temp[i + j] = exor(temp[i + j], divisor[j]);
                }
            }
        }

        int remLen = m - 1;
        int[] remainder = new int[remLen];
        int start = n - remLen;
        for (int i = 0; i < remLen; i++) remainder[i] = temp[start + i];
        return remainder;
    }

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

```

Output:

Case : 1

Enter the size of the data: 7

Enter the data bits (space separated):

1 0 1 1 0 0 1

Enter the size of the divisor: 3

Enter the divisor bits (space separated):

1 0 1

Remainder: 11

The CRC code generated is: 101100111

Enter the received codeword bits (space separated) OR press Enter to simulate correct transmission:

1 0 1 1 0 0 1 1 1

Data was received without any error.

Case :2

Enter the size of the data: 7

Enter the data bits (space separated):

1 0 1 1 0 0 1

Enter the size of the divisor: 3

Enter the divisor bits (space separated):

1 0 1

Remainder: 11

The CRC code generated is: 101100111

Enter the received codeword bits (space separated) OR press Enter to simulate correct transmission:

1 0 1 1 0 0 1 1 0

There is an error in received data...