

Assignment-3

Task:1

Problem-01:

A bit stream 1101011011 is transmitted using the standard CRC method. The generator polynomial is x^4+x+1 . What is the actual bit string transmitted?

```
case-1-lab.c++ > CRC(string, string)
1 // Case 2: Error in Transmission
2 // Data = 100100, Generator Polynomial (Key) =  $x^3 + x^2 + 1$  (1101)
3
4 // Receiver Side
5 // Let there be an error and code word received at the receiver side 100000001.
6
7 #include <iostream>
8 #include <string>
9 #include <cmath>
10 #include <algorithm>
11 using namespace std;
12 #define int long long int
13
14 // Function to convert integer to binary string
15 string toBin(int num) {
16     // Handle case when number is 0
17     if (num == 0) return "0";
18     string bin = "";
19     while (num) {
20         // Append '1' or '0' based on least significant bit
21         bin += (num & 1) ? '1' : '0';
22         // Shift right to process next bit
23         num = num >> 1;
24     }
25     // Reverse string since bits were added in reverse order
26     reverse(bin.begin(), bin.end());
27     return bin;
28 }
29
30 // Function to convert binary string to decimal integer
31 int toDec(string bin) {
32     int n = bin.size();
33     // Handle empty string
34     if (n == 0) return 0;
35     int num = 0;
36     for (int i = 0; i < n; i++) {
37         if (bin[i] == '1') {
38             // Compute power of 2 for each '1' in binary string
39             num += 1 << (n - i - 1);
40         }
41     }
42     return num;
43 }
44
45 // Function to compute CRC and print remainder and codeword
```

```

32 int toDec(string bin) {
33     return stoi(bin, nullptr, 2);
34 }
35
36 // Function to compute CRC and print remainder and codeword
37 void CRC(string data, string key) {
38     int n = key.length();
39     if (n == 0) {
40         cout << "Error: Key cannot be empty" << endl;
41         return;
42     }
43
44     // Convert binary strings to decimal integers
45     // Generator polynomial (key)
46     int gen = toDec(key);
47
48     // Original data
49     int code = toDec(data);
50
51     // Append (n - 1) zeros to the data to make space for CRC bits
52     int dividend = code << (n - 1);
53
54     // Calculate the position to start XOR (most significant bit position)
55     int shift;
56     while ((shift = (int)log2(dividend) - n + 1) >= 0) {
57         // Extract top 'n' bits of dividend, XOR with generator polynomial
58         int rem = (dividend >> shift) ^ gen;
59
60         // Replace top bits in dividend with XOR result (remainder)
61         dividend = (dividend & ((1 << shift) - 1)) | (rem << shift);
62     }
63
64     // Final codeword is the original data with the remainder appended
65     int codeword = (code << (n - 1)) | dividend;
66
67     // Print results
68     cout << "Remainder: " << toBin(dividend) << endl;
69     cout << "Codeword : " << toBin(codeword) << endl;
70 }
71
72 signed main() {
73     string data = "1101011011";
74     string key = "10011";
75     CRC(data, key);
76     return 0;
77 }

```

```

[Running] cd "/home/khaleepa/Desktop/Networking-Lab/lab4/" && g++ case-1-lab.cpp -o case-1-lab && "/home/khaleepa/Desktop/Networking-Lab/lab4/"case-1-lab
Remainder: 1110
Codeword : 11010110111110

[Done] exited with code=0 in 1.006 seconds

```