

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?

```
C case-1-lab.c++ > CRC(string, string)
1 // Case 2: Error in Transmission
2 // Data = 100100, Generator Polynomial (Key) = x3 + x2 + 1 (1101)
3
4 // Receiver Side
5 // Let there be an error and code word received at the receiver side 1000000001.
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
Windsurf:Refactor|Explain|X
15 string toBin(int num) {
16
17     // Handle case when number is 0
18     if (num == 0) return "0";
19     string bin = "";
20     while (num) {
21         // Append '1' or '0' based on least significant bit
22         bin += (num & 1) ? '1' : '0';
23         // Shift right to process next bit
24         num = num >> 1;
25     }
26     // Reverse string since bits were added in reverse order
27     reverse(bin.begin(), bin.end());
28     return bin;
29 }
30
31 // Function to convert binary string to decimal integer
Windsurf:Refactor|Explain|X
32 int toDec(string bin) {
33     int n = bin.size();
34
35     // Handle empty string
36     if (n == 0) return 0;
37     int num = 0;
38     for (int i = 0; i < n; i++) {
39         if (bin[i] == '1') {
40             // Compute power of 2 for each '1' in binary string
41             num += 1 << (n - i - 1);
42         }
43     }
44     return num;
45 }
46
47 // Function to compute CRC and print remainder and codeword
```

```
case-1-lab.c++ > CRC(string, string)
32 int toDec(string bin) {
33     vector<char> binary;
34 }
35
36 // Function to compute CRC and print remainder and codeword
37 // Windurf: Refactor | Explain | X
38 void CRC(string data, string key) {
39     int n = key.length();
40     if (n == 0) {
41         cout << "Error: Key cannot be empty" << endl;
42         return;
43     }
44
45     // Convert binary strings to decimal integers
46     // Generator polynomial (key)
47     int gen = toDec(key);
48
49     // Original data
50     int code = toDec(data);
51
52     // Append (n - 1) zeros to the data to make space for CRC bits
53     int dividend = code << (n - 1);
54
55     // Calculate the position to start XOR (most significant bit position)
56     int shft;
57     while ((shft = (int)log2(dividend) - n + 1) >= 0) {
58         // Extract top 'n' bits of dividend, XOR with generator polynomial
59         int rem = (dividend >> shft) ^ gen;
60
61         // Replace top bits in dividend with XOR result (remainder)
62         dividend = (dividend & ~(1 << shft) - 1) | (rem << shft);
63     }
64
65     // Final codeword is the original data with the remainder appended
66     int codeword = (code << (n - 1)) | dividend;
67
68     // Print results
69     cout << "Remainder: " << toBin(dividend) << endl;
70     cout << "Codeword : " << toBin(codeword) << endl;
71 }
72
73 // Windurf: Refactor | Explain | Generate Function Comment | X
74 signed main() {
75     string data = "11010110111";
76     string key  = "10011";
77     CRC(data, key);
78     return 0;
79 }
```

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

[Done] exited with code=0 in 1.006 seconds
```

Ln 77, Col 1 Spaces: 4 UTF-8 LF {} C++ Go Live Windsurf: Login Linux ✎ Prettier