

```

1 // SL.No.: - 31
2 // Admission No.: - 21JE0269
3 // Name: - Chotaliya Zeel Vijaybhai
4
5 #include <iostream>
6 #include <string>
7 #include <bitset>
8 #include <limits>
9 #include <vector>
10
11 using namespace std;
12
13 bitset<8> mod2Division(const bitset<40>& dividend, const bitset<9>& divisor) {
14     bitset<40> dividendCopy = dividend; // Copy of the dividend for division
15
16     // Perform the division process, starting from the most significant bit
17     for (int i = 39; i >= 8; i--) {
18         // If the current bit is 1, perform XOR with the divisor
19         if (dividendCopy[i] == 1) {
20             // XOR each bit of the divisor with the corresponding bit in the dividend
21             for (int j = 0; j < 9; ++j) {
22                 dividendCopy[i - j] = dividendCopy[i - j] ^ divisor[8 - j]; // XOR operation
23             }
24         }
25     }
26
27     // The remainder is in the least significant 8 bits of the dividendCopy
28     bitset<8> remainder;
29     for (int i = 0; i < 8; ++i) {
30         remainder[i] = dividendCopy[i]; // Copying remainder
31     }
32
33     return remainder; // Returning the 8-bit remainder
34 }
35
36 string crc8encode(string bitstream){
37
38
39     // converting string to bitset variable.
40     bitset<40> dataword(bitstream);
41
42     // appending 8 zeros at last.
43     dataword <<= 8;
44
45     // creating divisor.
46     bitset<9> divisor("100011101");
47
48     // performing modulo - 2 - division.
49     bitset<8> remainder;
50     remainder = mod2Division(dataword, divisor);
51     string remainderString = remainder.to_string();
52
53     // returning encoded bitsream.
54     return bitstream + remainderString;
55 }
56
57 void crc8decode(string bitstream){
58     // converting string to bitset variable.
59     bitset<40> dataword(bitstream);
60
61     // creating divisor.
62     bitset<9> divisor("100011101");
63
64     // performing modulo - 2 - division.
65     bitset<8> remainder;
66     remainder = mod2Division(dataword, divisor);

```

```

67     string remainderString = remainder.to_string();
68
69     // checking for error
70     bool iserror = false;
71     for(char c : remainderString){
72         if(c == '1'){
73             iserror = true;
74             break;
75         }
76     }
77
78     cout << "Decoded Bitstream" << endl;
79
80     if(iserror){
81         cout << "Error Detected. Ask for Retransmission." << endl;
82     }else{
83         cout << "No error;" << endl;
84         cout << "Data: - " << bitstream.substr(0, bitstream.size() - 8) << endl;
85     }
86     return;
87 }
88
89 int main(){
90
91     // Checking, whether user wants to send or receive message.
92     cout << "Are you sender (1) or receiver (0) ?" << endl;
93     bool x;
94     cin >> x;
95
96     string bitstream;
97
98     if(x){
99         cout << "Enter bitstream to be encoded : " << endl;
100         cin >> bitstream;
101
102         string encoded_bitstream;
103         encoded_bitstream = crc8encode(bitstream);
104
105         cout << "Encoded bitstream : " << endl;
106         cout << encoded_bitstream << endl;
107     }else{
108
109         cout << "Enter bitstream to be decoded : " << endl;
110         cin >> bitstream;
111
112         crc8decode(bitstream);
113     }
114
115     return 0;
116 }

```