Assignment 3

**Computer Network**

**Name of the student: Swaraj Phand**

**Roll No. 7**

**PRN- 12211292**

**Class- TY – D -B1**

**Div: C Batch: 3**

**Date of performance: 24/01/2024**

**CRC**

**3.1: CRC**

**Code:**

#include<iostream>

using namespace std;

string xor1(string a, string b);

string mod2div(string data, string divisor);

void encodeData(string data, string key);

void receiver(string data, string key);

int main(){

    string data = "11010110";

    string key = "10011";

    cout <<"Data: "<< data << endl;

    cout <<"Key: "<< key << endl;

    cout << "\nSender" << endl;

    encodeData(data, key);

    cout << "\nReceiver" << endl;

    receiver(data+mod2div(data+std::string(key.size() - 1, '0'),key), key);

    return 0;

}

void encodeData(string data, string key)

{

    int l\_key = key.length();

    string appended\_data = (data + string(l\_key - 1, '0'));

    string remainder = mod2div(appended\_data, key);

    string codeword = data + remainder;

    cout << "Remainder : " << remainder << "\n";

    cout << "Encoded Data (Data + Remainder) :" << codeword

         << "\n";

}

string mod2div(string data, string divisor){

    int l = divisor.length();

    string temp = data.substr(0,l);

    int n = data.length();

    while(l < n){

        if(temp[0] == '1'){

            temp = xor1(divisor, temp) + data[l];

        }else{

            temp = xor1(string(l, '0'), temp)+ data[l];

        }

        l += 1;

    }

    if (temp[0] == '1')

        temp = xor1(divisor, temp);

    else

        temp = xor1(string(l, '0'), temp);

    return temp;

}

string xor1(string a, string b)

{

    string result = "";

    int n = b.length();

    for (int i = 1; i < n; i++) {

        if (a[i] == b[i])

            result += "0";

        else

            result += "1";

    }

    return result;

}

void receiver(string data, string key)

{

    string currxor = mod2div(data.substr(0, key.size()), key);

    int curr = key.size();

    while (curr != data.size()) {

        if (currxor.size() != key.size()) {

            currxor.push\_back(data[curr++]);

        }

        else {

            currxor = mod2div(currxor, key);

        }

    }

    if (currxor.size() == key.size()) {

        currxor = mod2div(currxor, key);

    }

    if (currxor.find('1') != string::npos) {

        cout << "there is some error in data" << endl;

    }

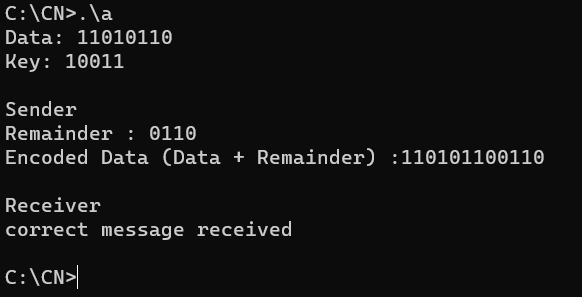
    else {

        cout << "correct message received" << endl;

    }

}

**Output:**

****