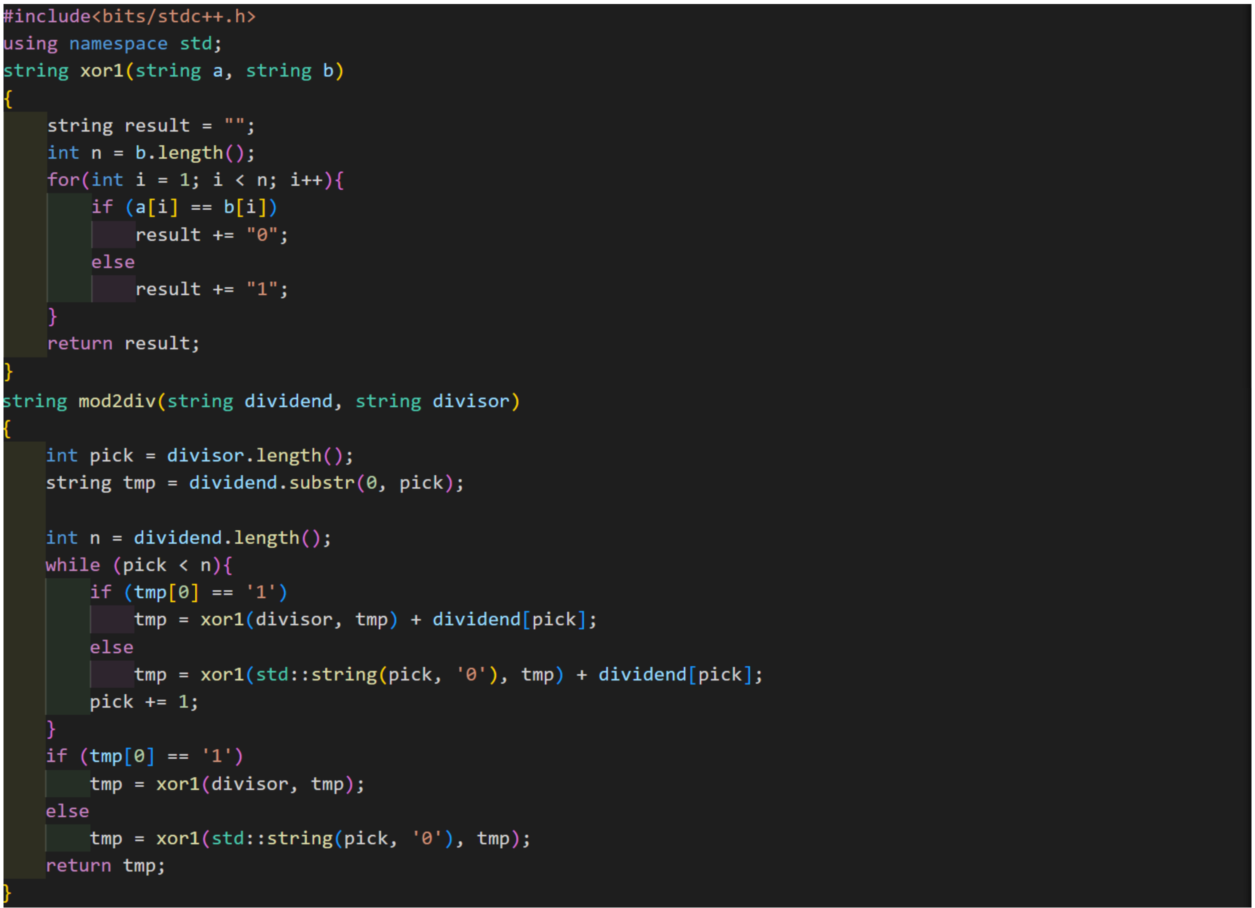
Data Communication and Computer Network
Lab3: CRC and Hamming code Error Detections

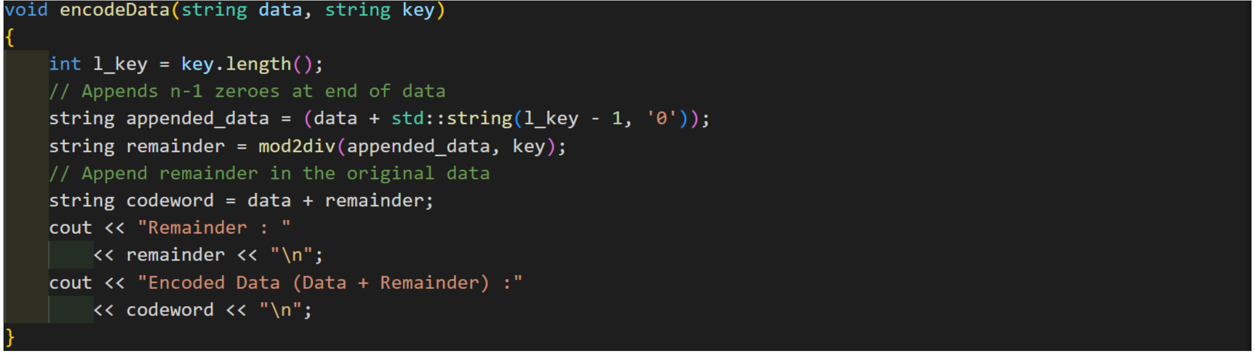
Ques1: Write a program for CRC and Hamming code. Output Format (CRC) Enter data to be transmitted: 1001101 Enter the Generating polynomial: 1011 --------------- Data padded with n-1 zeros : 1001101000 ------ CRC or Check value is : 101 --------- Final data to be sent : 1001101101 ---------Enter the received data: 1001101101 ---- Data received: 1001101101 No error detected.

Source Code



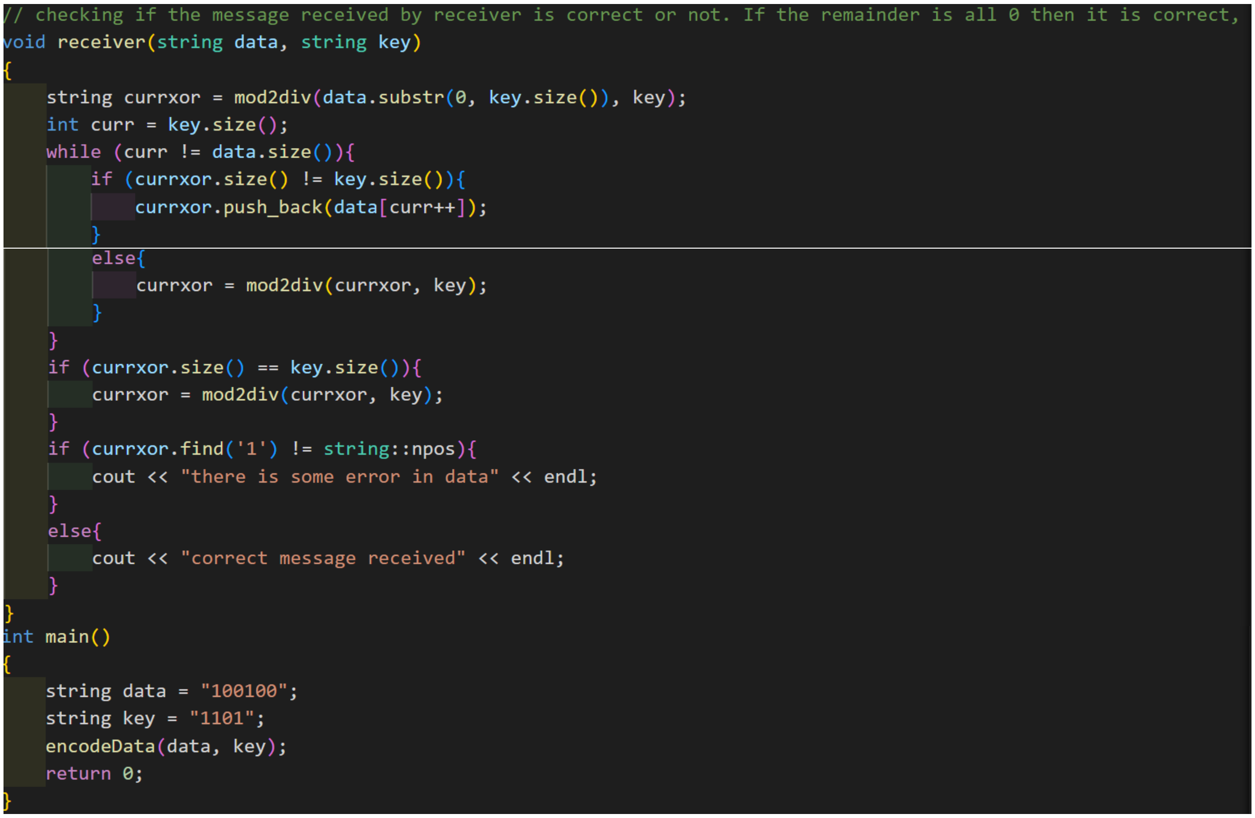
```
#include<bits/stdc++.h>
using namespace std;
string xor1(string a, string b)
{
    string result = "";
    int n = b.length();
    for(int i = 1; i < n; i++){
        if (a[i] == b[i])
            result += "0";
        else
            result += "1";
    }
    return result;
}
string mod2div(string dividend, string divisor)
{
    int pick = divisor.length();
    string tmp = dividend.substr(0, pick);

    int n = dividend.length();
    while (pick < n){
        if (tmp[0] == '1')
            tmp = xor1(divisor, tmp) + dividend[pick];
        else
            tmp = xor1(std::string(pick, '0'), tmp) + dividend[pick];
        pick += 1;
    }
    if (tmp[0] == '1')
        tmp = xor1(divisor, tmp);
    else
        tmp = xor1(std::string(pick, '0'), tmp);
    return tmp;
}
```



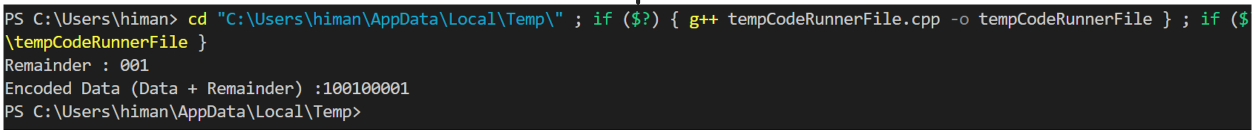
```
void encodeData(string data, string key)
{
    int l_key = key.length();
    // Appends n-1 zeroes at end of data
    string appended_data = (data + std::string(l_key - 1, '0'));
    string remainder = mod2div(appended_data, key);
    // Append remainder in the original data
    string codeword = data + remainder;
    cout << "Remainder : "
        << remainder << "\n";
    cout << "Encoded Data (Data + Remainder) :"
        << codeword << "\n";
}
```

Data Communication and Computer Network
Lab3: CRC and Hamming code Error Detections



```cpp
// checking if the message received by receiver is correct or not. If the remainder is all 0 then it is correct,
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;
    }
}
int main()
{
    string data = "100100";
    string key = "1101";
    encodeData(data, key);
    return 0;
}
```

Output



```
PS C:\Users\himan> cd "C:\Users\himan\AppData\Local\Temp\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($
\tempCodeRunnerFile }
Remainder : 001
Encoded Data (Data + Remainder) :100100001
PS C:\Users\himan\AppData\Local\Temp>
```