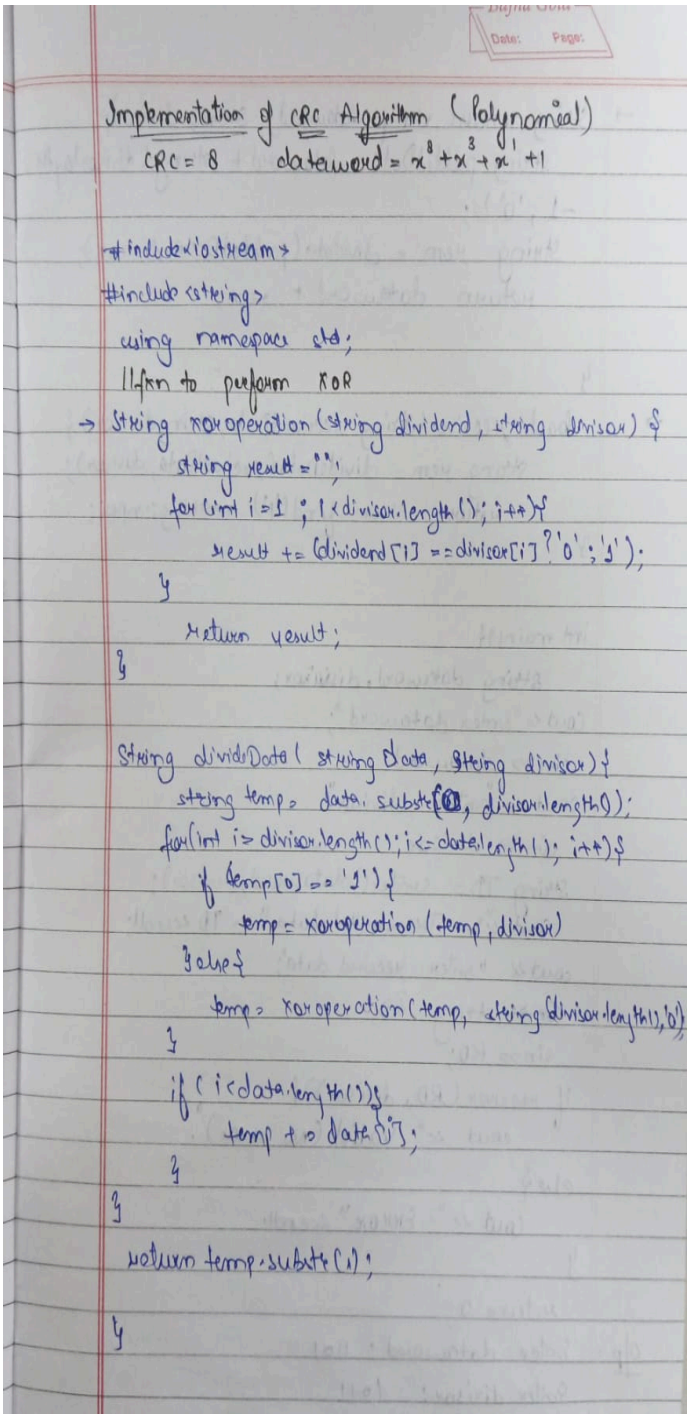
NAME : MANVI SHARMA USN: 1BM22CS149

CN LAB 9

AIM: IMPLEMENTATION OF CRC ALGORITHM

OBSERVATION:

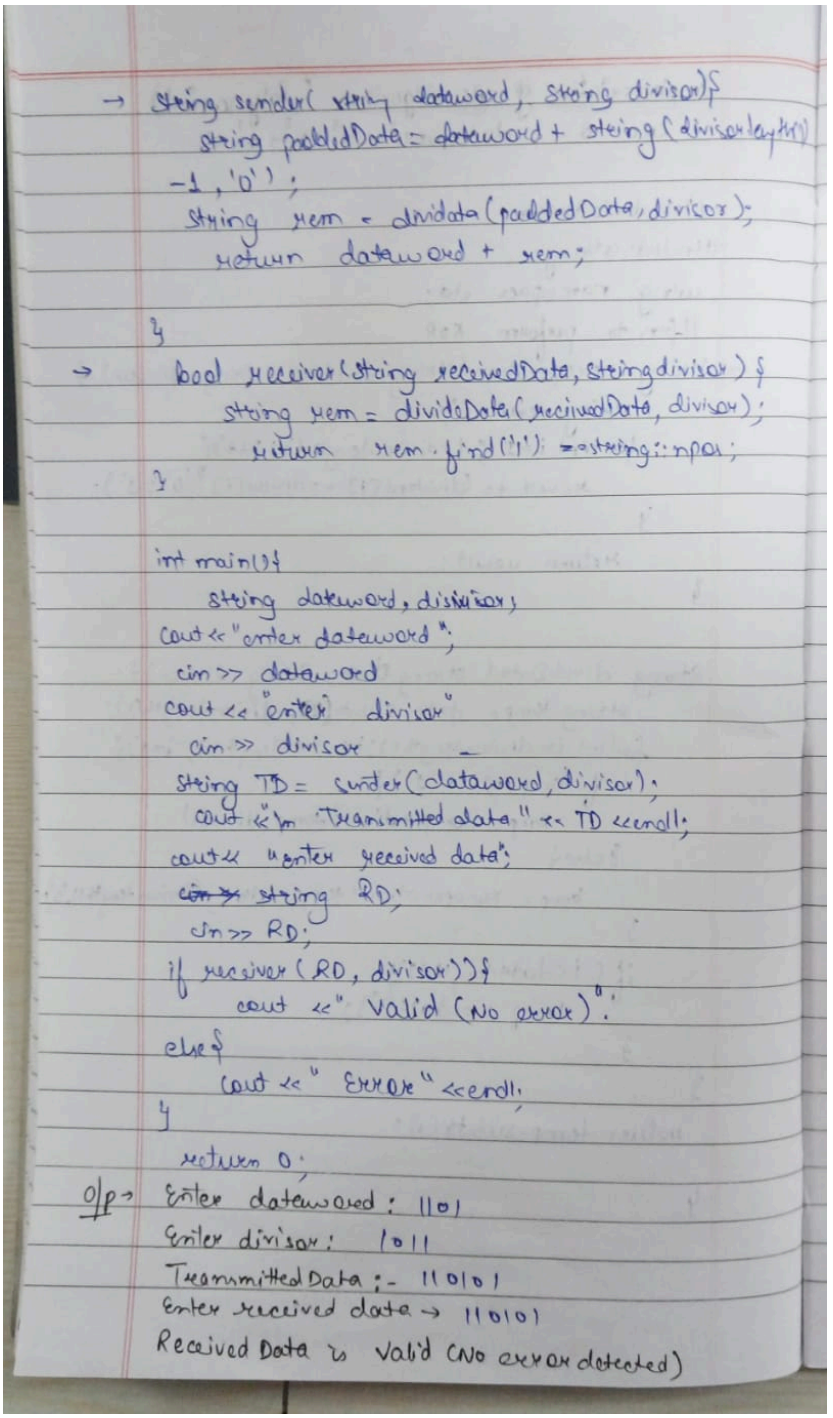


Implementation of CRC Algorithm (Polynomial)

CRC = 8 dataword = $x^8+x^3+x^1+1$

```
#include<iostream>
#include<string>
using namespace std;
//fxn to perform XOR
string xoroperation(string dividend, string divisor){
    string result="";
    for(int i=1; i<divisor.length(); i++){
        result += (dividend[i]==divisor[i]?'0':'1');
    }
    return result;
}

string divideData(string data, string divisor){
    string temp= data.substr(0, divisor.length());
    for(int i=divisor.length(); i<=data.length(); i++){
        if (temp[0]=='1'){
            temp= xoroperation(temp, divisor)
        }else{
            temp= xoroperation(temp, string(divisor.length(),'0'));
        }
        if (i<data.length()){
            temp += data[i];
        }
    }
    return temp.substr(1);
}
```



```
string sender(string dataword, string divisor){
    string paddedData= dataword + string(divisor.length()
    -1,'0');
    string rem = dividata(paddedData, divisor);
    return dataword + rem;
}

bool receiver(string receivedData, string divisor){
    string rem = divideData(recivedData, divisor);
    return rem.find('1') == string::npos;
}

int main(){
    string dataword, disivisor;
    cout<<"enter dataword";
    cin>> dataword
    cout<<"enter divisor"
    cin>> divisor
    string TD= sender(dataword, divisor);
    cout<<"\n Transmitted data"<< TD <<endl;
    cout<< "enter received data";
    string RD;
    cin>> RD;
    if (resciver(RD, divisor)){
        cout<<" valid (No error)";
    }else{
        cout<<" Error"<<endl;
    }
    return 0;
}
```

o/p→ Enter dataword : 1101
Enter divisor : 1011
Transmitted Data :- 110101
Enter received data → 110101
Received Data is valid (No error detected)

PROGRAM:

```
#include <iostream>
#include <string>

using namespace std;

// Function to perform XOR operation

string xorOperation(string dividend, string divisor) {
    string result = "";
    for (int i = 1; i < divisor.length(); i++) {
        result += (dividend[i] == divisor[i] ? '0' : '1');
    }
    return result;
}


// Function to perform binary division

string divideData(string data, string divisor) {
    string temp = data.substr(0, divisor.length());
    for (int i = divisor.length(); i <= data.length(); i++) {
        if (temp[0] == '1') {
            temp = xorOperation(temp, divisor);
        } else {
            temp = xorOperation(temp, string(divisor.length(), '0'));
        }
        if (i < data.length()) {
            temp += data[i];
        }
    }
    return temp.substr(1); // Return the remainder
}


// Sender side: Generate transmitted data

string sender(string dataword, string divisor) {
    string paddedData = dataword + string(divisor.length() - 1, '0'); // Append zeros equal to
(degree of divisor - 1)
    string remainder = divideData(paddedData, divisor);             // Calculate CRC
    return dataword + remainder;                                    // Concatenate dataword and CRC
remainder
```

```
}

// Receiver side: Verify received data
bool receiver(string receivedData, string divisor) {
    string remainder = divideData(receivedData, divisor);
    return remainder.find('1') == string::npos; // If remainder is all zeros, data is valid
}

int main() {
    string dataword, divisor;

    // Input dataword and divisor in binary form
    cout << "Enter the dataword (binary polynomial, e.g., 1101 for x^3 + x^2 + 1): ";
    cin >> dataword;
    cout << "Enter the divisor (binary polynomial, e.g., 1011 for x^3 + x + 1): ";
    cin >> divisor;

    // Sender Side
    string transmittedData = sender(dataword, divisor);
    cout << "\nTransmitted Data (Dataword + CRC): " << transmittedData << endl;

    // Receiver Side
    cout << "Enter the received data (binary polynomial): ";
    string receivedData;
    cin >> receivedData;

    if (receiver(receivedData, divisor)) {
        cout << "Received data is valid (No error detected)." << endl;
    } else {
        cout << "Received data contains errors." << endl;
    }

    return 0;
}
```

OUTPUT:

CASE 1 -> NO ERROR

```
Enter the dataword (binary polynomial, e.g., 1101 for x^3 + x^2 + 1): 1101
Enter the divisor (binary polynomial, e.g., 1011 for x^3 + x + 1): 1011

Transmitted Data (Dataword + CRC): 110101
Enter the received data (binary polynomial): 110101
Received data is valid (No error detected).

=== Code Execution Successful ===
=== Session Ended. Please Run the code again ===
```

CASE 2 -> ERROR

```
Enter the dataword (binary polynomial, e.g., 1101 for x^3 + x^2 + 1): 1101
Enter the divisor (binary polynomial, e.g., 1011 for x^3 + x + 1): 1011

Transmitted Data (Dataword + CRC): 110101
Enter the received data (binary polynomial): 110100
Received data contains errors.


=== Code Execution Successful ===
```