

# CodeSpeedy

**AIM:** To implement CRC.

**THEORY:** Cyclic Redundancy Check(CRC) is a way of identifying unintended alterations or errors in a communication system. CRC employs the Generator Polynomial, which is present on both the transmitter and recipient sides. An example generator polynomial is in the form of  $x^5+x^3 + x + 1$ . This generator polynomial represents key 101011. Another example is  $x^2 +x+ 1$  that represents key 111.

On the Sender side, the binary division is done to get CRC.

Receiver's side Checks for any errors that may have been introduced while transmission. After performing modulo-2 division repeatedly if the remainder is 0, then no errors are there.

## IMPLEMENTATION

```
#include <iostream>

using namespace std;

class CRC
{
public:

    int nf, ng, frame[20], gen[10], temp[20], b;

    char a;
    int* divide(int n, int g, int temp[10], int gen[10])
    {

        for(int i=0; i<n; i++)
        {

            if(gen[0]==temp[i])
            {
```

```

        for(int j=0, k=i; j<g+1; j++, k++)
        {
            if(temp[k]^gen[j]==1)

                temp[k]=1;

            else

                temp[k]=0;

        }

    }

    return temp;
}

void input()
{
    cout<<"Enter length of your frame:";
    cin>>nf;

    cout<<"Enter your frame:";

    for(int i=0; i<nf; i++)
    {

        cin>>frame[i];

        temp[i]=frame[i];

    }

    cout<<"Enter length of your generator:";
    cin>>ng;

```

```

        cout<<"Enter your generator:";

        for(int i=0; i<ng; i++)

        {

            cin>>gen[i];

        }

        ng--;

        for(int i=0; i<ng; i++)

        {

            temp[nf+i]=0;

        }

    }

    void sender_side()

    {

        int* sender;

        sender = divide(nf,ng,temp,gen);

        cout<<endl<<"-----Senders Side-----\n"<<"CRC:";

        for(int i=0; i<ng; i++)

        {

            frame[nf+i]=sender[nf+i];

            cout<<sender[nf+i]<<' ';

        }

```

```

        cout<<endl<<"Transmitted frame:";

        for(int i=0; i<nf+ng; i++)

            cout<<frame[i]<<' ';

        cout<<endl;
    }

    int receiver_side()
    {

        int* receiver;

        cout<<"\n-----Receivers Side-----\n"<<"Received
message:";

        for(int i=0; i<nf+ng; i++)

            cout<<frame[i]<<' ';

        cout<<endl;

        cout<<"Enter which bit you want to change (from
0-"<<nf+ng<<") - ";
        cin>>b;

        if (frame[b]==1)

            frame[b]=0;

        else

            frame[b]=1;

        receiver = divide(nf,ng,frame,gen);
    }

```

```

        cout<<"Error:";

        for(int i=0; i<nf+ng; i++)
        {
            if(receiver[i]!=0)
            {
                cout<<"Error Detected!!"<<endl;

                return 0;
            }
        }

        cout<<"No error detected!"<<endl;
    }
};

```

```

int main()
{
    CRC o;
    o.input();

    o.sender_side();

    o.receiver_side();

    return 0;
}

```

## OUTPUT:

```
C:\Users\91870\Desktop\cd\crc.exe
Enter length of your frame: 6
Enter your frame: 1 0 0 1 0 0
Enter length of your generator: 4
Enter your generator: 1 1 0 1

-----Senders Side-----
CRC:0 0 1
Transmitted frame:1 0 0 1 0 0 0 0 1

-----Receivers Side-----
Received message:1 0 0 1 0 0 0 0 1
Enter which bit you want to change(from 0-9)- 3
Error:Error Detected!!

-----
Process exited after 48.2 seconds with return value 0
Press any key to continue . . .
```