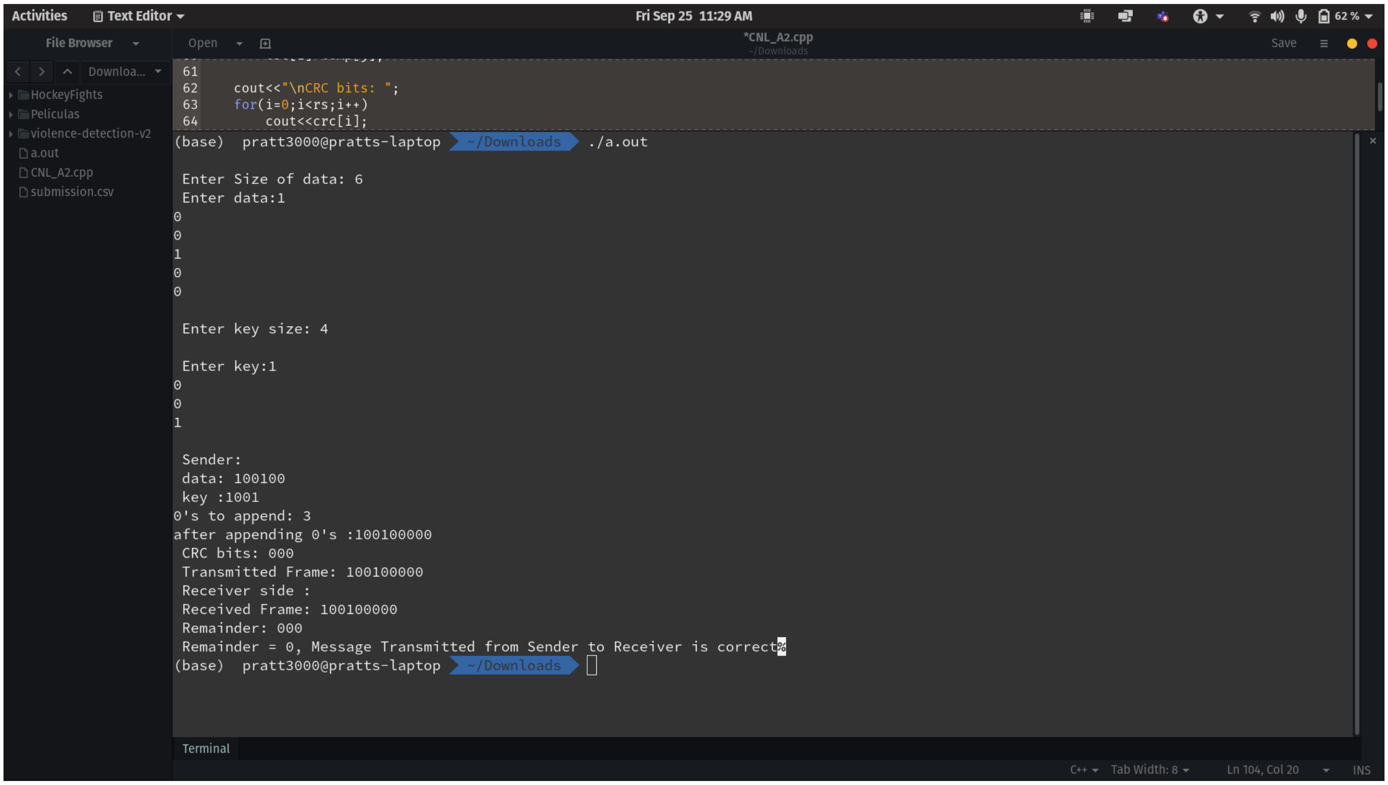
CODE

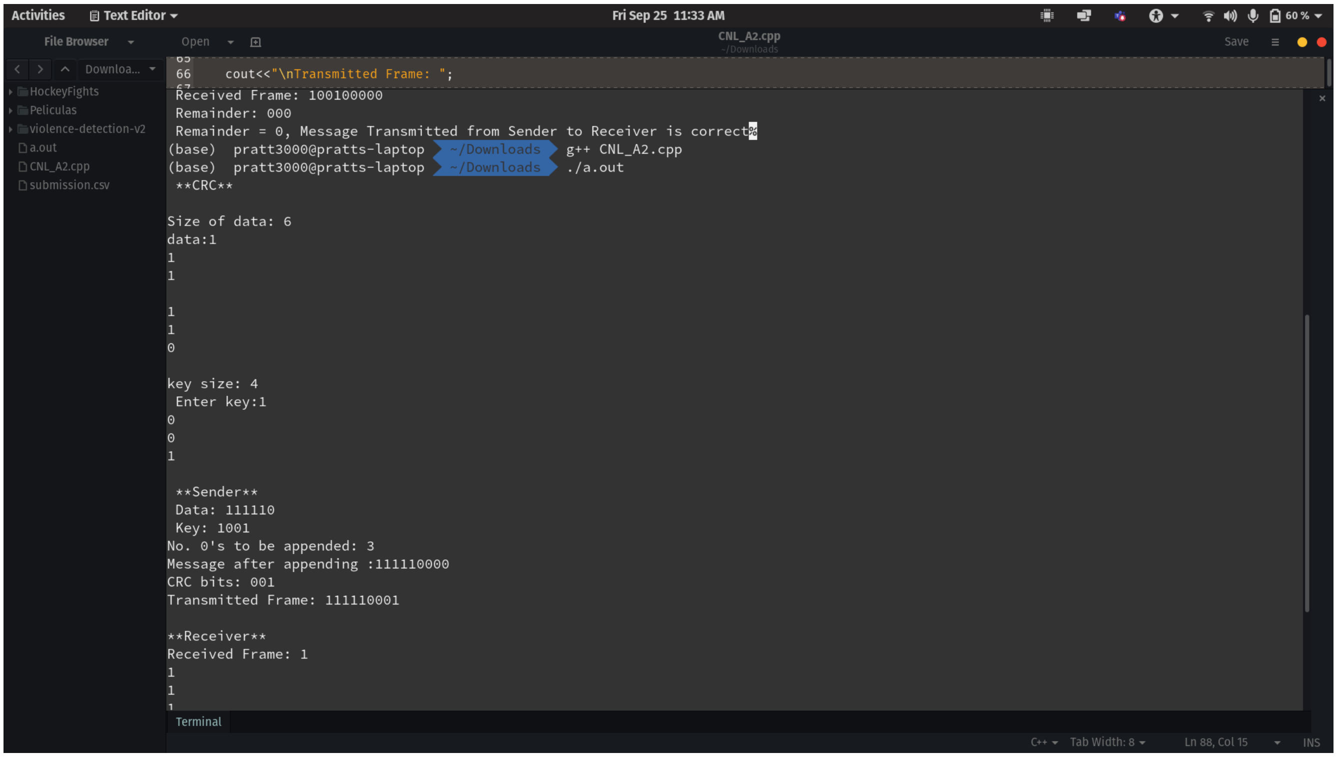
```
#include <bits/stdc++.h>

using namespace std;

void division(int temp[], int g[], int fs, int gs){
    for(int i=0;i<fs;i++){
        int j=0,k=i;
        if (temp[k]>=g[j]){
            for(j=0,k=i;j<gs;j++,k++){
                if((temp[k]==1 && g[j]==1) || (temp[k]==0 && g[j]==0))
                    temp[k]=0;
                else
                    temp[k]=1;
            }
        }
    }
}

int main(){
            cout<<" **CRC**\n";
    int i,j,k,l;
    int fs, f[20], gs, g[20], temp[20], tf[15], crc[15];
    cout<<"\nSize of data: ";
    cin>>fs;
    cout<<"data:";
    for(i=0;i<fs;i++)
        cin>>f[i];

    cout<<"\nkey size: ";
    cin>>gs;
    cout<<" Enter key:";
    for(i=0;i<gs;i++)
        cin>>g[i];

    cout<<"\n **Sender**";
    cout<<"\n Data: ";
    for(i=0;i<fs;i++)
        cout<<f[i];

    cout<<"\n Key: ";
```

```
for(i=0;i<gs;i++)
    cout<<g[i];

int rs=gs-1;
cout<<"\nNo. 0's to be appended: "<<rs;
for (i=fs;i<fs+rs;i++)
    f[i]=0;

for(i=0;i<20;i++)
    temp[i]=f[i];

cout<<"\nMessage after appending :";
for(i=0; i<fs+rs;i++)
    cout<<temp[i];

division(temp,g,fs,gs);

for(i=0,j=fs;i<rs;i++,j++)
    crc[i]=temp[j];

cout<<"\nCRC bits: ";
for(i=0;i<rs;i++)
    cout<<crc[i];

cout<<"\nTransmitted Frame: ";

for(i=0;i<fs;i++)
    tf[i]=f[i];

for(i=fs,j=0;i<fs+rs;i++,j++)
    tf[i]=crc[j];

for(i=0;i<fs+rs;i++)
    cout<<tf[i];

cout<<"\n\n**Receiver**";
label:
cout<<"\nReceived Frame: ";

        for(i=0;i<fs+rs;i++)
    cin>>temp[i];

division(temp,g,fs,gs);

cout<<"Remainder: ";
int rrem[15];
for (i=fs,j=0;i<fs+rs;i++,j++)
    rrem[j]=temp[i];
    for(i=0;i<rs;i++)
        cout<<rrem[i];
  libr
    int flag=0;
    for(i=0;i<rs;i++){
        if(rrem[i]!=0)
            flag=1;
    }

    if(flag==0)
      cout<<"\n\nMessage transmitted from Sender to Receiver is correct";

    else{
        cout<<"\n\nMessage transmitted from Sender to Receiver contains ERROR!\n Re-transmit the
data ";
        goto label;
   }

return 0;
}
```



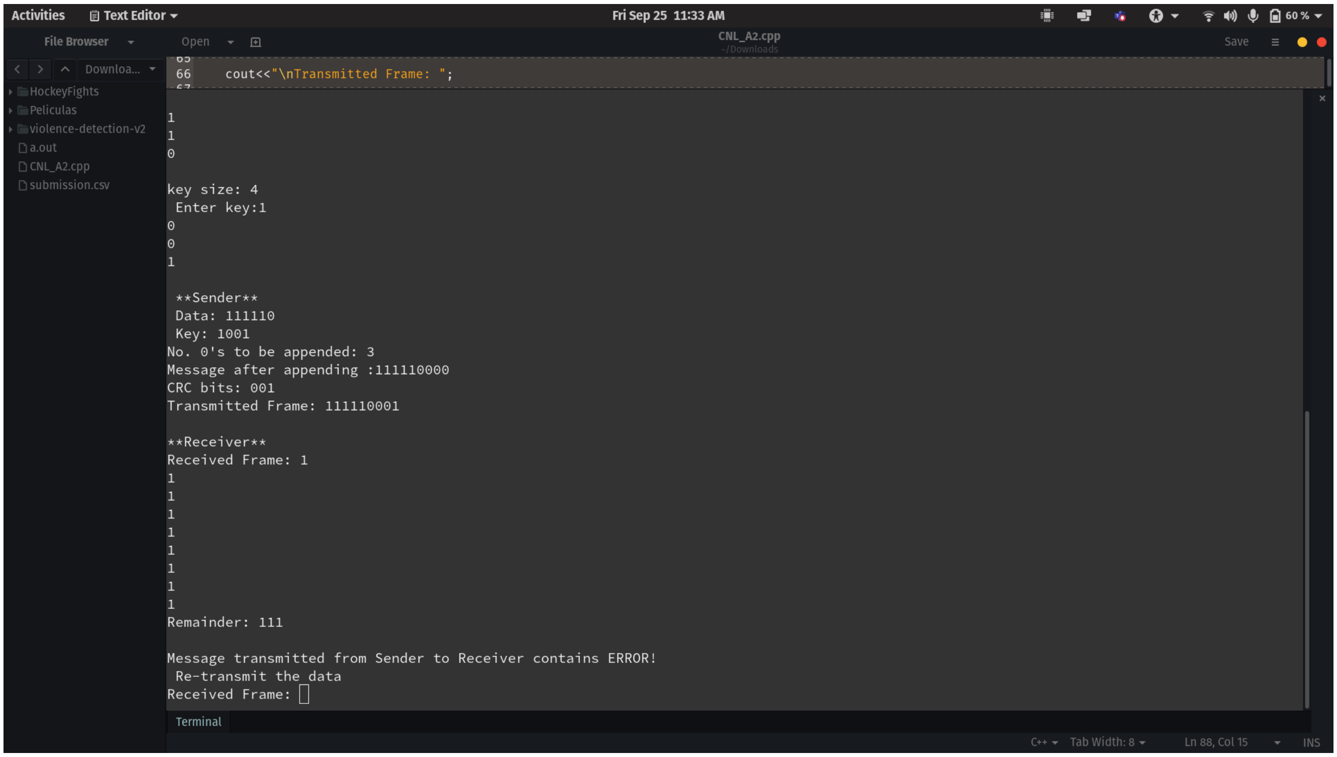
```
    cout<<"\nCRC bits: ";
    for(i=0;i<rs;i++)
        cout<<crc[i];
```

```
(base)  pratt3000@pratts-laptop  ~/Downloads  ./a.out

 Enter Size of data: 6
 Enter data:1
0
0
1
0
0


 Enter key size: 4

 Enter key:1
0
0
1


 Sender:
 data: 100100
 key :1001
0's to append: 3
after appending 0's :100100000
 CRC bits: 000
 Transmitted Frame: 100100000
 Receiver side :
 Received Frame: 100100000
 Remainder: 000
 Remainder = 0, Message Transmitted from Sender to Receiver is correct%
(base)  pratt3000@pratts-laptop  ~/Downloads
```



```
    cout<<"\nTransmitted Frame: ";
```

```
 Received Frame: 100100000
 Remainder: 000
 Remainder = 0, Message Transmitted from Sender to Receiver is correct%
(base)  pratt3000@pratts-laptop  ~/Downloads  g++ CNL_A2.cpp
(base)  pratt3000@pratts-laptop  ~/Downloads  ./a.out
 **CRC**

Size of data: 6
data:1
1
1

1
1
0

key size: 4
 Enter key:1
0
0
1

 **Sender**
 Data: 111110
 Key: 1001
No. 0's to be appended: 3
Message after appending :111110000
CRC bits: 001
Transmitted Frame: 111110001

**Receiver**
Received Frame: 1
1
1
1
```



```
    cout<<"\nTransmitted Frame: ";
```

```
1
1
0

key size: 4
 Enter key:1
0
0
1

 **Sender**
 Data: 111110
 Key: 1001
No. 0's to be appended: 3
Message after appending :111110000
CRC bits: 001
Transmitted Frame: 111110001

**Receiver**
Received Frame: 1
1
1
1
1
1
1
1
1
Remainder: 111

Message transmitted from Sender to Receiver contains ERROR!
 Re-transmit the data
Received Frame: 
```

Assignment 2

Title: CRC (Cyclic Redundancy Check)

Problem Statement:
Write a program for error detection & correction for 7/8 bits ASCII codes using CRC. Demonstrate the packets captured using wireshark Packet Analyzer Tool for peer to peer mode

Software & Hardware Req.:
C++ Compiler, Wireshark Packet Analyzer Tool, IDE, processor.

Theory:
In digital systems, the analog signals will change into digital sequence (in form of bits). This sequence of bits is called a "Data Stream". The change in position of single bit also leads to catastropic error in data output.

Error:
1) The data can be corrupted during transmission.
2) It may be affected by external noise.
3) This mismatch is caller "Error"

* Types of Error:

1) Single Bit Error:

a) The change in one bit in whole data Sequence is called Single Bit Error.

b) Occurence of single bit error is very rare in serial communication system.

2) Multiple Bit Data Errors

a) It there is change in two or more bits of data sequence of transmitter to receiver, it is called 'Multiple Bit Error'

b) This type of error occurs both in seriol type & parallel type data communication networks.

3) Burst Errors:

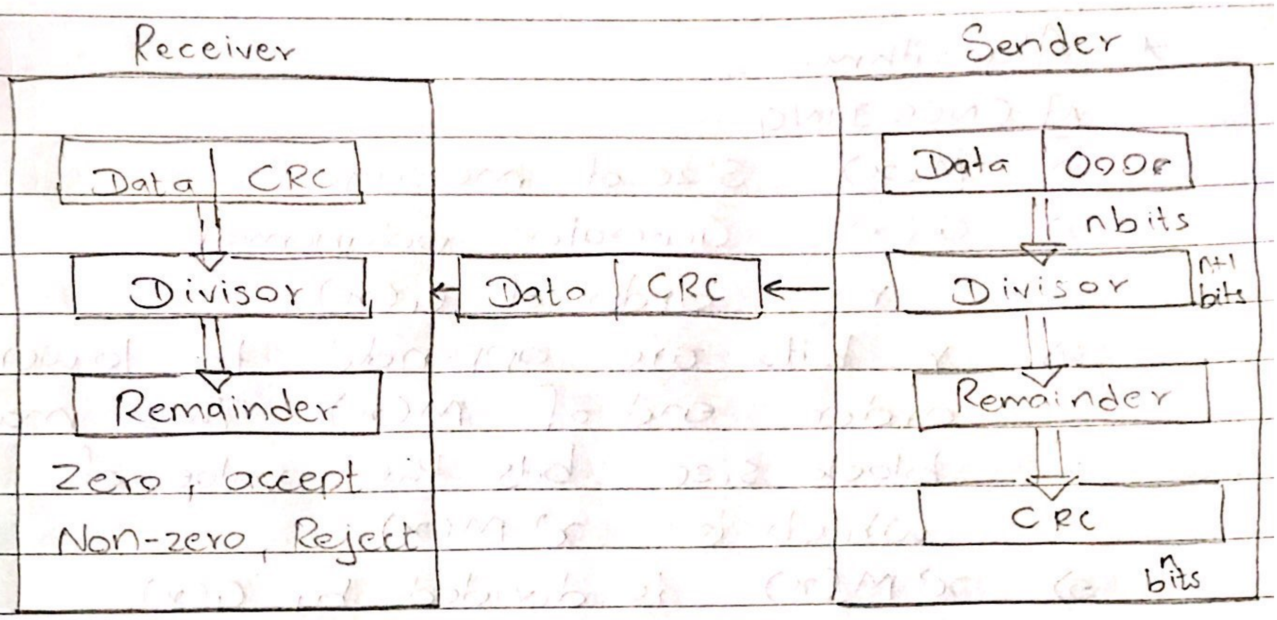
a) The change of set of bits in data sequence is called Burst Error.

b) The burst error is calculated in form first bit change to last bit change.

* Error Detecting Codes:

1) Errors are transferred from one comm. system to another.

2) If these are not detected & corrected then data will be lost.



CRC

★ Error Detecting Codes:

1) Parity Checking
2) CRC
3) LRC
4) Check Sum

- CRC (Cyclic Redundancy Check)

1) A cyclic code is a linear (n, k) block code with the property that every cyclic shift of a code word results in another code word.
2) k: length of message
3) n: length of message after adding check bits.
4) These are called CRC codes.
5) These are implemented using shift registers with feedback connections.

* Algorithm:

A] ENCODING:

1) $M(x)$: Size of message
2) $G(x)$: Generator polynomial
3) r : order of $G(x)$
4) r bits are appended to lower order end of $M(x)$. This makes - block size bits the value of which is $x^r M(x)$
5) $x^r M(x)$ is divided by $G(x)$ [Modulo 2 division]
6] The remainder is added to $x^r M(x)$
7) The result is frame to be transmitted $T(x)$

B] DECODING

1) $T(x)$ is divided by $G(x)$ [modulo 2 div.]
2) If there is no error then frame is accepted
3) The remainder indicates error $E(x)$ in which case frame is rejected.

• Testing:

	Sender Side	Receiver Side	Accept/Reject
1)	Data : 1010101 CRC : 001 Divisor : 100 Transmitted: 101010100	Received: 101010100 Remainder: 0	Accept

Data: 10011	Received:	Rejected
Divisor: 1101	10011001	
CRC : 011	Remainder:	
Transmitted: 10011011	001	

Conclusion:

Error Detection & correction using CRC was studied & implemented successfully.