**Design:**

I have implemented the program using Java and its multithreading concept.

In this code there is one sender and one receiver and implemented this code in different function like LRC, VRC , CRC and Checksum. Where each program have one main thread and one child thread.

Using synchronisation. First sender will send the input then by different error detection methods receiver will receive the data. Then according to the various error detection methods the data stream will received by the receiver.

Sender

Receiver

Data stream passes by

Error detection methods

**Fig 1.**

**Code snippet for LRC error detection method:**

**class** LRC **extends** Thread{

**public** **boolean** checkforerr(ArrayList<Integer>tocheck,ArrayList<Integer>check)

{

**for**(**int** i=0;i<tocheck.size();i++)

{

**int** f=tocheck.get(i);

**int** s=check.get(i);

**if**(f!=s)

**return** **false**;

}

**return** **true**;

}

**public** **void** run()

{

Scanner p=**new** Scanner(System.***in***);

**int** n=p.nextInt();

ArrayList<ArrayList<Integer>>arr=**new** ArrayList();

ArrayList<Integer>inp=**new** ArrayList();

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

{

**for**(**int** j=0;j<n;j++) {

**int** c;c=p.nextInt();

inp.add(c);}

arr.add(inp);

}

ArrayList<Integer>checkres=**new** ArrayList<>();

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

{

checkres.add(arr.get(0).get(i));

}

ArrayList<Integer> res=**new** ArrayList();

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

**int** d=0;

**for**(**int** j=0;j<n;j++)

{

**if**(arr.get(j).get(i)==1)

{

d++;

}

}

**if**(d%2==1)

{

res.add(1);

}

**else** {

res.add(0);

}

}

**if**(checkforerr(checkres,res))

{

System.***out***.println("No error");

}

**else** {

System.***out***.println("Error found");

}

}

}