TITLE:-

Name:- KAUSHAL AGARWAL
Roll No:- 001810501051
Class:- BCSE Third year
Group:- A2

Assignment No:- 1
Date:-02.11.2020

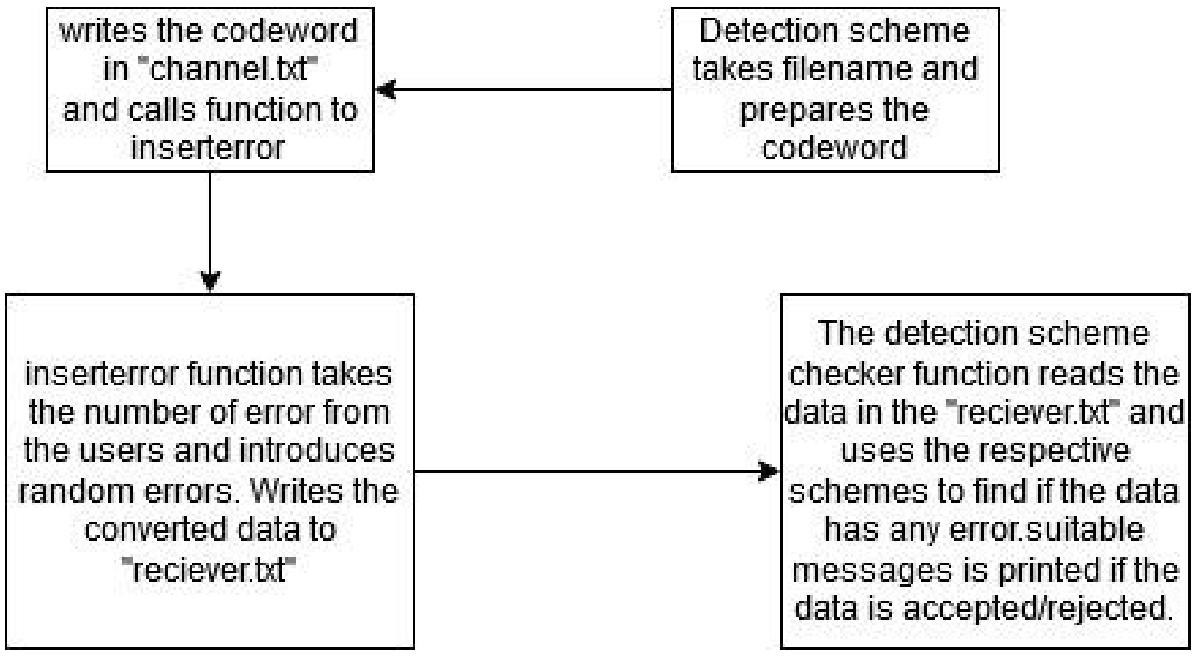
Problem Statement:-

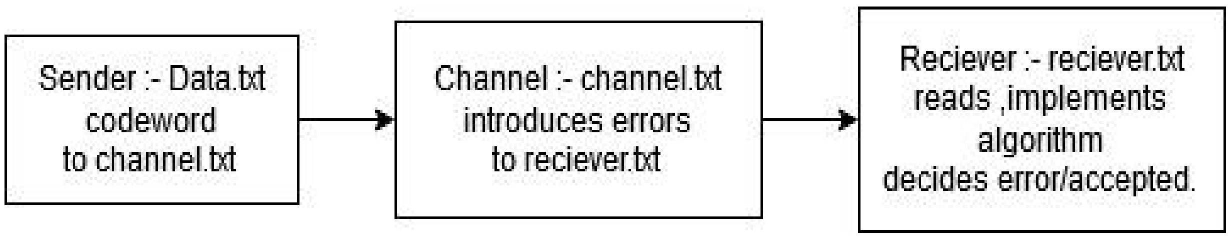
Design and implement an error detection module which has four schemes namely LRC, VRC, Checksum and CRC. The Sender program should accept the name of a test file (contains a sequence of 0,1) from the command line. Then it will prepare the data frame (decide the size of the frame) from the input. Based on the schemes, codeword will be prepared. Sender will send the codeword to the Receiver. Receiver will extract the dataword from codeword and show if there is any error detected. Test the same program to produce a PASS/FAIL result for following cases.
(a) Error is detected by all four schemes. Use a suitable CRC polynomial (list is given in next page).
(b) Error is detected by checksum but not by CRC.
(c) Error is detected by VRC but not by CRC.
[Note: Inject error in random positions in the input data frame. Write a separate method for that.]

DESIGN:-

Purpose:- The given assignment is to implement the various error detection schemes.The purpose of the program is to simulate the real implementation of these techniques. The User can visualize what actually happen in these techniques. All the code is written in c++ language and file handling is used to show the data flow Between sender and channel , channel and reciever.

Diagram:-


writes the codeword in "channel.txt" and calls function to inserterror
Detection scheme takes filename and prepares the codeword
inserterror function takes the number of error from the users and introduces random errors. Writes the converted data to "reciever.txt"
The detection scheme checker function reads the data in the "reciever.txt" and uses the respective schemes to find if the data has any error.suitable messages is printed if the data is accepted/rejected.


Sender :- Data.txt codeword to channel.txt
Channel :- channel.txt introduces errors to reciever.txt
Reciever :- reciever.txt reads ,implements algorithm decides error/accepted.

Input/Output format:-

The data (binary) is written in any text file and the name of that text file is to be provided. Output is the different stages of execution and whether the data has error, it is accepted or rejected.Some inputs is provided by the user for better visualization like the number of errors the channel would introduce and the polynomial code in cyclic redundancy code detection technique.

Implementation

void inserterror();
Reads the data from “channel.txt” and takes the number of errors to produce during the channel transmission and introduces random error.Writes to “reciever.txt”

void truncdata(string filename,int size);
Checks if the data in the file can be divided into groups of size and if not add the zero’s in the front.

void vrc(string filename);
Calls truncdata(filename,8);
To encode VRC scheme and writes the data to channel.txt. Calls inserterror() and checkvrc()

void checkvrc();
Checks by VRC scheme if the data recieved in “reciever.txt” has error and displays suitable messages.

void lrc(string filename);
Calls truncdata(filename,32);
To encode LRC scheme and writes the data to channel.txt. Calls inserterror() and checklrc()

void checklrc();
Checks by LRC scheme if the data recieved in "reciever.txt" has error and displays suitable messages.

void addbits(char sum[],char word[]);
Adds two binary of length 8.

void checksum(string filename);
Calls truncdata(filename,8);
To encode CHECKSUM scheme and writes the data to channel.txt. Calls inserterror() and checksumChecker().

void checksumChecker();
Checks by CHECKSUM scheme if the data recieved in "reciever.txt" has error and displays suitable messages.

void division(int temp[],int gen[],int n,int r);
performs the division for crc, takes the data as integer array of size n,generator polynomial of size r.

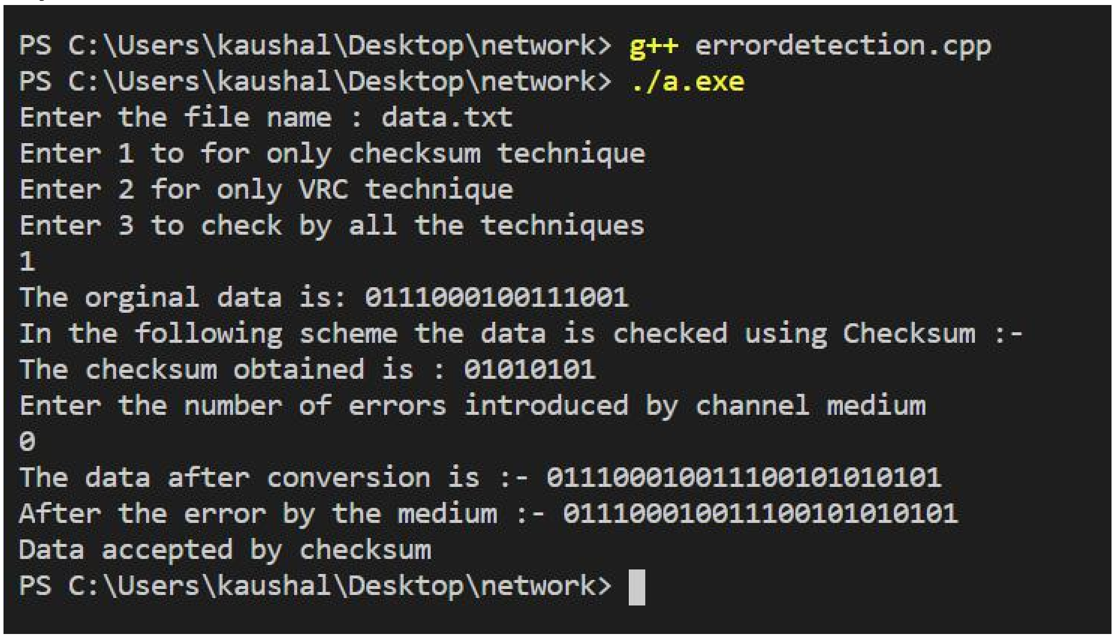
void crc(string filename);
To encode LRC scheme and writes the data to channel.txt. Calls inserterror() and checkcrc()

void checkcrc(int gen[],int r);

Checks by CRC scheme if the data recieved in “reciever.txt” has error and displays suitable messages.

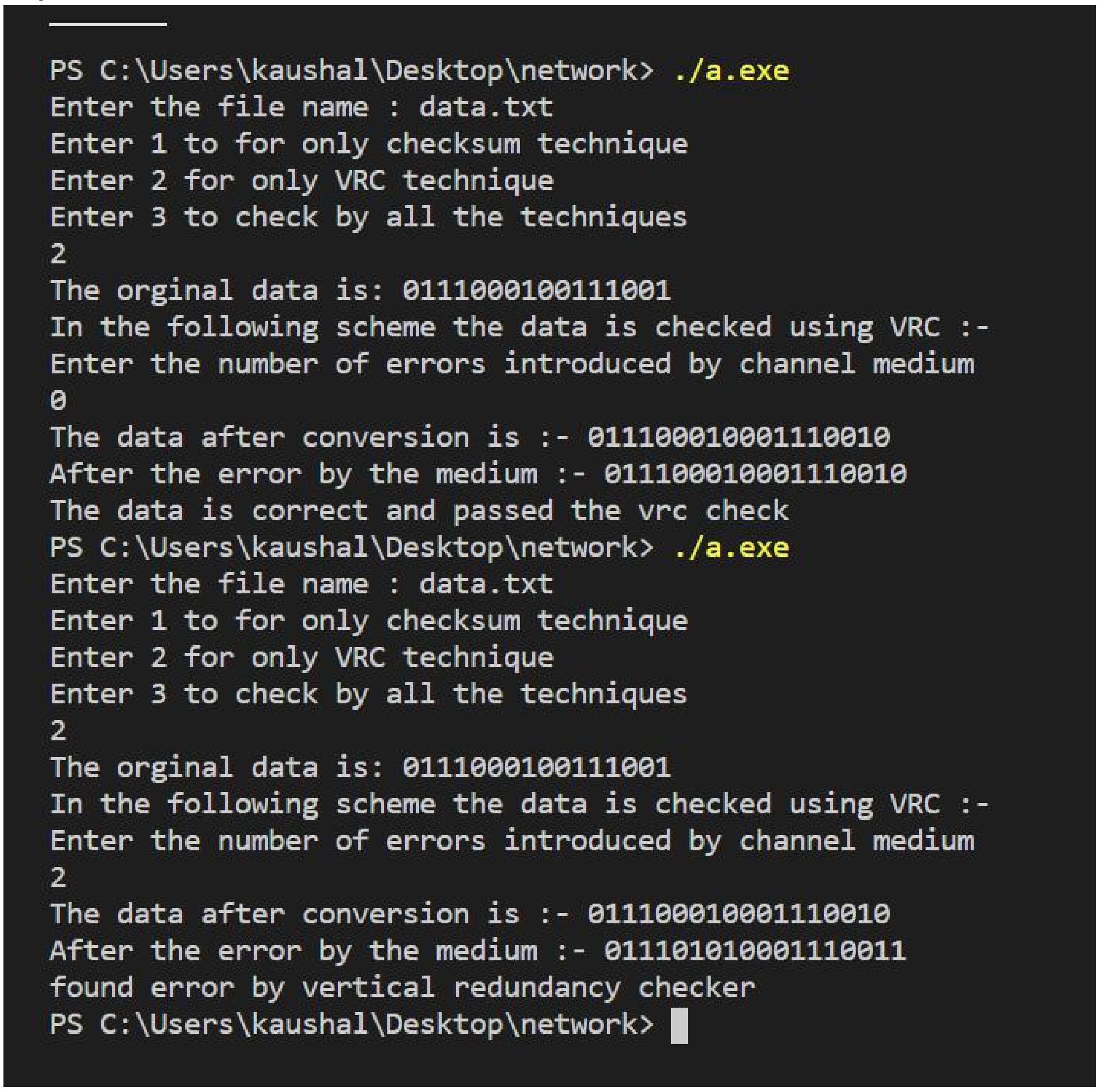
Testcase:-

By CHECKSUM:-



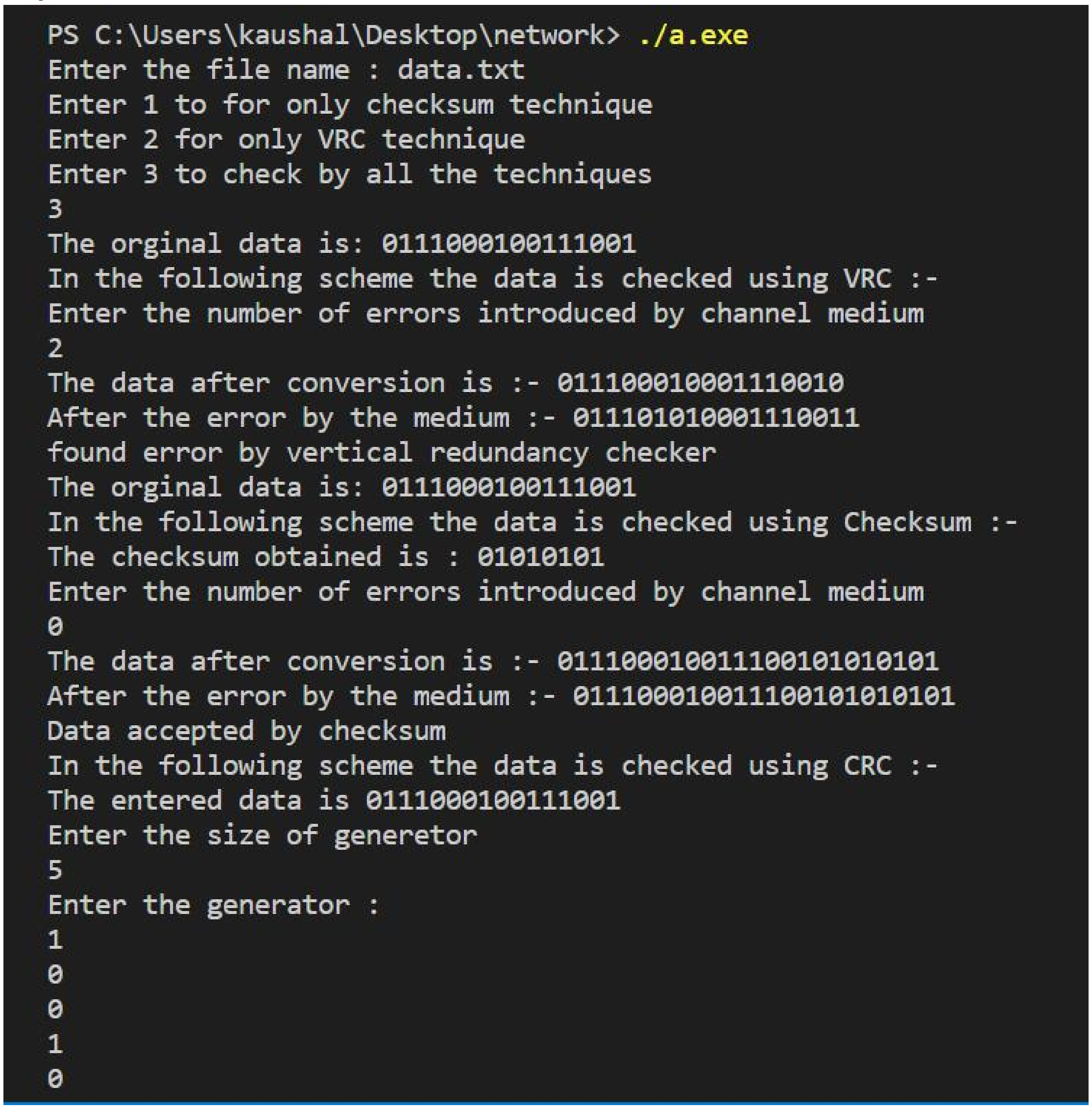
```
PS C:\Users\kaushal\Desktop\network> g++ errordetection.cpp
PS C:\Users\kaushal\Desktop\network> ./a.exe
Enter the file name : data.txt
Enter 1 to for only checksum technique
Enter 2 for only VRC technique
Enter 3 to check by all the techniques
1
The orginal data is: 0111000100111001
In the following scheme the data is checked using Checksum :-
The checksum obtained is : 01010101
Enter the number of errors introduced by channel medium
0
The data after conversion is :- 011100010011100101010101
After the error by the medium :- 011100010011100101010101
Data accepted by checksum
PS C:\Users\kaushal\Desktop\network> 
```

By VRC:-

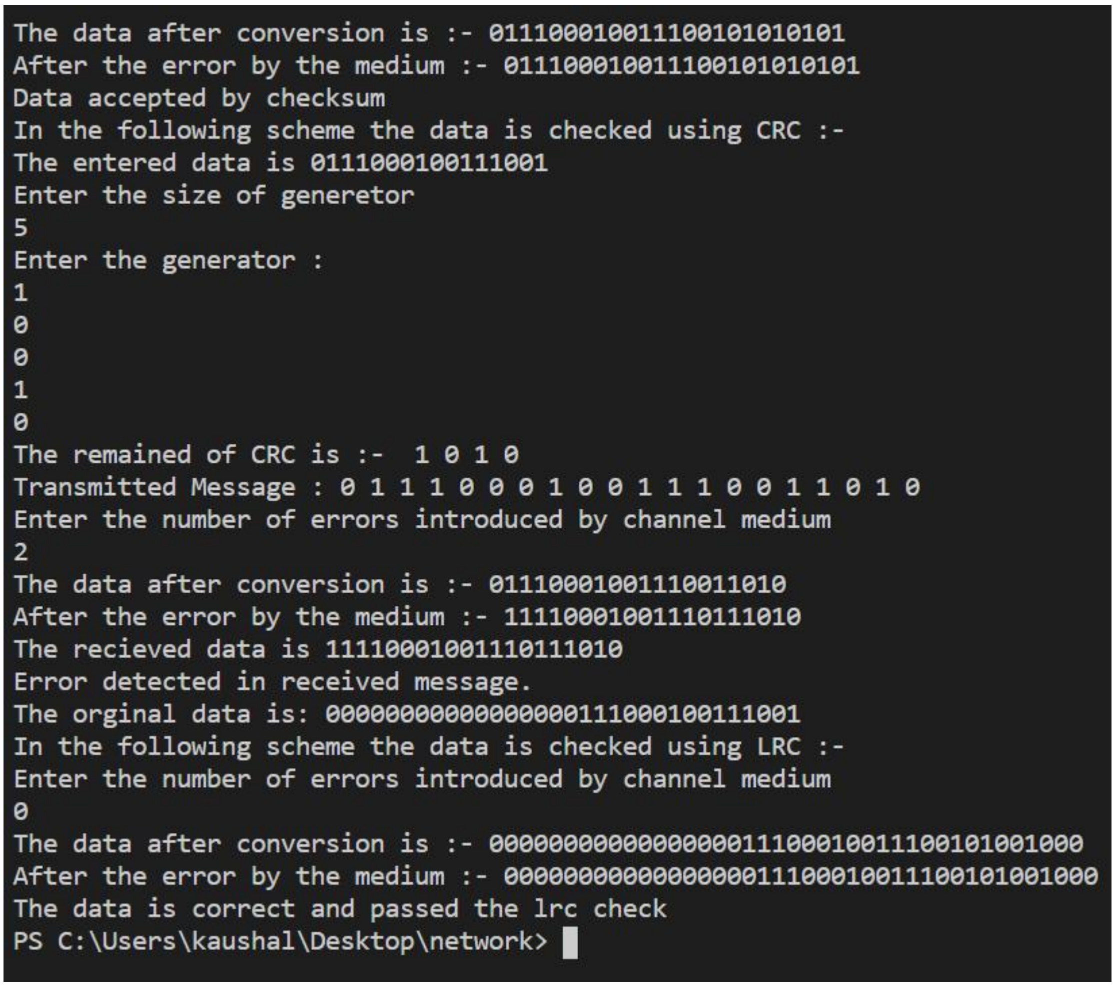


```
PS C:\Users\kaushal\Desktop\network> ./a.exe
Enter the file name : data.txt
Enter 1 to for only checksum technique
Enter 2 for only VRC technique
Enter 3 to check by all the techniques
2
The orginal data is: 0111000100111001
In the following scheme the data is checked using VRC :-
Enter the number of errors introduced by channel medium
0
The data after conversion is :- 0111000100011100010
After the error by the medium :- 0111000100011100010
The data is correct and passed the vrc check
PS C:\Users\kaushal\Desktop\network> ./a.exe
Enter the file name : data.txt
Enter 1 to for only checksum technique
Enter 2 for only VRC technique
Enter 3 to check by all the techniques
2
The orginal data is: 0111000100111001
In the following scheme the data is checked using VRC :-
Enter the number of errors introduced by channel medium
2
The data after conversion is :- 0111000100011100010
After the error by the medium :- 0111010100011100011
found error by vertical redundancy checker
PS C:\Users\kaushal\Desktop\network>
```

By ALL SCHEMES:-



```
PS C:\Users\kaushal\Desktop\network> ./a.exe
Enter the file name : data.txt
Enter 1 to for only checksum technique
Enter 2 for only VRC technique
Enter 3 to check by all the techniques
3
The orginal data is: 0111000100111001
In the following scheme the data is checked using VRC :-
Enter the number of errors introduced by channel medium
2
The data after conversion is :- 011100010001110010
After the error by the medium :- 011101010001110011
found error by vertical redundancy checker
The orginal data is: 0111000100111001
In the following scheme the data is checked using Checksum :-
The checksum obtained is : 01010101
Enter the number of errors introduced by channel medium
0
The data after conversion is :- 011100010011100101010101
After the error by the medium :- 011100010011100101010101
Data accepted by checksum
In the following scheme the data is checked using CRC :-
The entered data is 0111000100111001
Enter the size of generetor
5
Enter the generator :
1
0
0
1
0
```



```
The data after conversion is :- 011100010011100101010101
After the error by the medium :- 011100010011100101010101
Data accepted by checksum
In the following scheme the data is checked using CRC :-
The entered data is 0111000100111001
Enter the size of generetor
5
Enter the generator :
1
0
0
1
0
The remained of CRC is :-  1 0 1 0
Transmitted Message : 0 1 1 1 0 0 0 1 0 0 1 1 1 0 0 1 1 0 1 0
Enter the number of errors introduced by channel medium
2
The data after conversion is :- 01110001001110011010
After the error by the medium :- 11110001001110111010
The recieved data is 11110001001110111010
Error detected in received message.
The orginal data is: 00000000000000000111000100111001
In the following scheme the data is checked using LRC :-
Enter the number of errors introduced by channel medium
0
The data after conversion is :- 0000000000000000011100010011100101001000
After the error by the medium :- 0000000000000000011100010011100101001000
The data is correct and passed the lrc check
PS C:\Users\kaushal\Desktop\network> █
```

Results:-

The programs performs implementation of every error detection technique correctly and appropriate messages are displayed. The suitable encoded data can be seen in the respective files.Overall the task is achieved.

Analysis:-

The possible error prone areas are the strict input/output guideline. The disk should have space for “reciever.txt” and “channel.txt” and no other file with same name should be present else the data of the file would be lost. Wrong input may lead to exceptions and program to behave inappropriately. Improvements can be done to use a better flow control of data.

Comment:-

Overall the assignment was tricky to implement but is a good way to understand the error detection techniques. The theory of subject is understood if one implements it and tests for different testcases.