What is Error?

Error is a condition when the output information does not match with the input information. During transmission, digital signals suffer from noise that can introduce errors in the binary bits travelling from one system to other. That means a 0 bit may change to 1 or a 1 bit may change to 0.

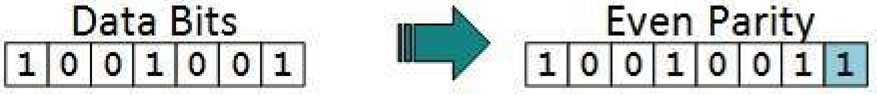
Error-Detection

Errors in the received frames are detected by means of Parity Check and Cyclic Redundancy Check (CRC).

Parity Check

One extra bit is sent along with the original bits to make number of 1s either even in case of even parity, or odd in case of odd parity.

The sender while creating a frame counts the number of 1s in it. For example, if even parity is used and number of 1s is even then one bit with value 0 is added. This way number of 1s remains even.If the number of 1s is odd, to make it even a bit with value 1 is added.



The receiver simply counts the number of 1s in a frame. If the count of 1s is even and even parity is used, the frame is considered to be not-corrupted and is accepted. If the count of 1s is odd and odd parity is used, the frame is still not corrupted.

If a single bit flips in transit, the receiver can detect it by counting the number of 1s. But when more than one bits are erro neous, then it is very hard for the receiver to detect the error.

```
#include<bits/stdc++.h>
using namespace std;

string getString(char x)
{
    string s(1, x);
    return s;
}

int  calculate_vrs(string val)
{
    int j=0;
    int res =0;
    for(int i=0; i<val.length(); i++)
    {
        int x;
        string s = getString(val[i]);
        stringstream geek(s);
        geek >> x;
        //cout<<"\n"<<x;
        res=res+x;
    }
    //cout<<"\n"<<res;
    return res;
}
int main()
{
    int n;
    //cout<<"----WELCOME-----\n";
    cout<<"Enter 0 for ODD\nENTER 1 for EVEN\n";
    cin>>n;

    int res;
    string line;
    string val[4];
    ifstream f ("datav.txt");
    int i=0;
    while(getline(f, line))
    {
        cout<<line;
        val[i]=line;
        cout<<"\n"<<val[i];
        i++;
        cout<<"\n";
    }
    for(int i=0; i<4; i++)
    {
        res = calculate_vrs(val[i]);
        cout<<"\n  "<<i<<" : "<<res;
        if(n==1)
        {
            if(res%2==0)
            {
                val[i].append("0");
                cout<<"\n"<<val[i];
            }
            else
            {
                val[i].append("1");
                cout<<"\n"<<val[i];
            }
        }
        else
        {
            if(res%2==0)
            {
                val[i].append("1");
                cout<<"\n"<<val[i];
            }
            else
            {
                val[i].append("0");
                cout<<"\n"<<val[i];
            }
        }
    }

    cout<<"\ndone";


}
```

Cyclic Redundancy Check (CRC)

CRC is a different approach to detect if the received frame contains valid data. This technique involves binary division of the data bits being sent.