

Practical 5 : Implement sender parity LRC,VRC & CRC programs for input1.txt and input2.txt.

Hardware Requirement : N/A

Software Requirement : Code Blocks

Knowledge Requirement : C++

Theory:

LRC:-

```
#include<iostream>

#include<fstream>

using namespace std ;

void evenparity ( int arr [16]){
int parity [4 ] ;
int count [4]={0};
for ( int p=0;p<4;p++){
{ for ( int i=p; i <16; i+=4){
i f ( arr [ i ]==1)
count [p]++;
}
}
for ( int i =0;i <4; i++)
{ i f ( count[ i]%2==0)
{ parity [ i ]=0; }
else {
parity [ i ]=1;
}
}
cout<<endl ;
for ( int i =0;i <16; i++)
{ cout<<arr [ i ] ; }
cout<<" ";
for ( int i =0;i <4; i++){
cout<<parity [ i ] ;
}
}

void oddparity ( int arr [16]){
int parity [4 ] ;
int count [4]={0};
for ( int p=0;p<4;p++){
{ for ( int i=p; i <16; i+=4){
i f ( arr [ i ]==1)
count [p]++;
}
```

Data Communication And Networking Practicals

```
    }
}
for ( int i =0;i <4; i++)
{
    if ( count[ i]%2==0)
    {
        parity [ i ]=1;
    }
    else {
        parity [ i ]=0;
    }
}
cout<<endl ;
for ( int i =0;i <16; i++)
{
    cout<<arr [ i ] ;
}
cout<<"+" ;
for ( int i =0;i <4; i++){ cout<<parity[];
}
}
void filecheck ()
{
    int parity [ 4 ] ;
    int count [4]={0};
    int x ;
    char arr [ 17 ];
    fstream f i l e ;
    f i l e . open(" data2 . txt " , ios :: in | ios :: out );
    for ( int j =0;j <16; j++)
    {
        file >>arr ;
    }

    cout<<endl<<"Choose vertical redundancy check:"<<endl ;

    cout<<"1.Even Parity"<<endl ; cout<<"2.Odd Parity"<<endl ;

    cout<<"Enter your choice :";
    cin>>x ;
    switch (x)
    {
        case 1:
for ( int p=0;p<4;p++)
{
    for ( int i=p; i <16;i+=4)
    {
        if ( arr [ i ]=='1')
        count [p]++;
    }
}
```

Data Communication And Networking Practicals

```
}
} for ( int i =0;i <4; i++)
{
    i f ( count [ i]%2==0)
    {
        parity [ i ]=0;
    }
    else {
        parity [ i ]=1;
    }
}
cout<<endl ;
for ( int i =0;i <16; i++)
{ cout<<arr [ i ] ;
}
cout<<"+"";
for ( int i =0;i <4; i++)
{
    cout<<parity [ i ] ;
}
break ;
case 2:
for ( int p=0;p<4;p++)
{
    for ( int i=p; i<16; i+=4){
        i f ( arr [ i ]=='1')
            count [p]++;
    }
} for ( int i =0;i <4; i++)
{
    i f ( count [ i]%2==0)
    {
        parity [ i ]=1;
    }
    Else
    {
        parity [ i ]=0;
    }
}
cout<<endl ;
for ( int i =0;i <16; i++)
{
    cout<<arr [ i ] ;
}
cout<<"+"";
for ( int i =0;i <4; i++)
{
    cout<<parity [ i ] ;
}
}
```

Data Communication And Networking Practicals

```
    }
}
int main()
{
    int data [16] ;
    int x ;
    char g ;
    do{
        cout<<endl<<"Choose type of longitudinal redundancy check:"<< cout<<"1.Even Parity"<<endl ;
        cout<<"2.Odd Parity"<<endl ; cout<<"3.Check from    f i l e"<<endl ; cout<<"Enter your choice :";
        cin>>x ;
        switch (x)
        {
case 1:
            cout<<"Enter Data :";
            for ( int i =0;i <16; i++){
                cin>>data [ i ] ;
            }
            evenparity ( data ); break ;
            case 2: cout<<"Enter Data :";
                for ( int i =0;i <16; i++){
                    cin>>data [ i ] ;
                }
                oddparity ( data );
            break ;
            case 3:
                filecheck ();
                break ;
            default :
                cout<<"Invalid Choice ";
            }

            cout<<endl<<endl<<"Do you want to continue ?";

            cin>>g ;

        }
    while (g=='y ' || g=='Y' ) ;
}
```

Output of LRC:

Data Communication And Networking Practicals

```
Choose type of longitudinal redundancy check:
1.Even Parity
2.Odd Parity
3.Check from file
Enter your choice:1
Enter Data:1 0 1 0 1 1 0 1 0 1 0 0 1 0 0 1

1010110101001001+1010

Do you want to continue?y

Choose type of longitudinal redundancy check:
1.Even Parity
2.Odd Parity
3.Check from file
Enter your choice:2
Enter Data:1 0 1 0 1 1 0 1 0 1 0 0 1 0 0 1

1010110101001001+0101

Do you want to continue?y

Choose type of longitudinal redundancy check:
1.Even Parity
2.Odd Parity
3.Check from file
Enter your choice:3

Choose vertical redundancy check:
1.Even Parity
2.Odd Parity
Enter your choice:1

1010111001100110+0100

Do you want to continue?y

Choose type of longitudinal redundancy check:
1.Even Parity
2.Odd Parity
3.Check from file
Enter your choice:3

Choose vertical redundancy check:
1.Even Parity
2.Odd Parity
Enter your choice:2

1010111001100110+1011
```

2. VRC:-

Code of VRC:-

```
#include<iostream>
#include<fstream>
using namespace std ;
void evenparity ( int arr [8])
{
    int count=0;
    for ( int i =0;i <8; i++)
    {
        if ( arr [ i ]==1)
            count++;
    }
    if ( count%2==0)
    {
        for ( int i =0;i <8; i++)
```

Data Communication And Networking Practicals

```
{
    cout<<arr [ i ] ;
}
cout<<"0";
}
else
{
    for ( int i =0;i <8; i++)
    {
        cin>>arr [ i ];
    }
    cout<<"1";
}
} void oddparity ( int arr [8])

{

    int count=0;

    for ( int i =0;i <8; i++){
        if ( arr [ i ]==1)
            count++;
    }
    if ( count%2==0)
    {
        for ( int i =0;i <8;i++)
        {
            cout<<arr [ i ] ;
        }
        cout<<"1";
    }
    else
    {
        for ( int i =0;i <8; i++)
        {
            cin>>arr [ i ];
        }
        cout<<"0";
    }
}
void filecheck ()
{
    char c ;
    int x ;
    int count=0;
    char arr [ 9 ] ;
    fstream f i le;
    fi l e . open(" data . txt " , ios : : in | ios : : out );
    for ( int j =0;j <8; j++)
```

Data Communication And Networking Practicals

```
{
    file >>arr ;
}
cout<<endl<<"Choose vertical redundancy check:"<<endl ;
    cout<<"1.Even Parity"<<endl ; cout<<"2.Odd Parity"<<endl ; cout<<"Enter

    your choice :";

    cin>>x ;

    switch (x)
    {
        case 1:
            for ( int i =0;i <8; i++){
                if ( arr [ i ]=='1')
                    count++;
            }
            if ( count%2==0)
            {
                for ( int i =0;i <8;i++)
                { cout<<arr [ i ] ;
                }

                cout<<"0";
            }

            else {
                for ( int i =0;i <8; i++)
                {
                    cin>>arr [ i ] ;
                }

                cout<<"1";
            }
            break ;
        case 2: for ( int i =0;i <8; i++)
        {
            if ( arr [ i ]=='1')
                count++;
        }
        if ( count%2==0)
        {
            for ( int i =0;i <8;i++)
            {
                cout<<arr [ i ] ;
            }
            cout<<"1";
        }

    }
}
```

Data Communication And Networking Practicals

```
else

{

for ( int i =0;i <8; i++)

{

cin>>arr [ i ] ;

}

    cout<<"0";
}
}

int main()
{
    int i;
    int x ;
    char c ,p;
    int data [ 8 ] ;
    do{
        cout<<endl<<"Choose vertical redundancy check:"<<endl ;
        cout<<"1.Even Parity"<<endl ; cout<<"2.Odd Parity"<<endl ;
        cout<<"3.Check from file"<<endl ;
        cout<<"Enter your choice :";
        cin>>x ;
        switch (x)
        {
            case 1:
                cout<<"Enter Data in form of 8 bits :";
                for ( i =0;i <8; i++)
                {
                    cin>>data [ i ] ;
                }
                evenparity ( data );
                break ;
            case 2:
                cout<<"Enter Data in form of 8 bits :";
                for ( i =0;i <8; i++)
                {
                    cin>>data [ i ] ;
                }
                oddparity ( data );
                break;
            case 3:
                filecheck ();
                break ;
        }
    }
}
```


Data Communication And Networking Practicals

```
        default :  
cout<<"Invalid      Choice ";  
    }  
  
    cout<<endl<<endl<<"Do you want t continue ?";  
  
    cin>>p;  
  
    }  
    while (p=='y' ||      p=='Y' );  
    }
```

Output:

```
Choose vertical redundancy check:  
1.Even Parity  
2.Odd Parity  
3.Check from file  
Enter your choice:1  
Enter Data in form of 8 bits:1 0 1 0 1 1 0 0  
10101100+0  
  
Do you want to continue?y  
  
Choose vertical redundancy check:  
1.Even Parity  
2.Odd Parity  
3.Check from file  
Enter your choice:2  
Enter Data in form of 8 bits:1 0 1 0 1 1 0 0  
10101100+1+0  
  
Do you want to continue?y  
  
Choose vertical redundancy check:  
1.Even Parity  
2.Odd Parity  
3.Check from file  
Enter your choice:3  
  
Choose vertical redundancy check:  
1.Even Parity  
2.Odd Parity  
Enter your choice:1  
10101001+0  
  
Do you want to continue?y  
  
Choose vertical redundancy check:  
1.Even Parity  
2.Odd Parity  
3.Check from file  
Enter your choice:3  
  
Choose vertical redundancy check:  
1.Even Parity  
2.Odd Parity  
Enter your choice:2  
10101001+1
```

3. CRC:-

Code of LRC:-

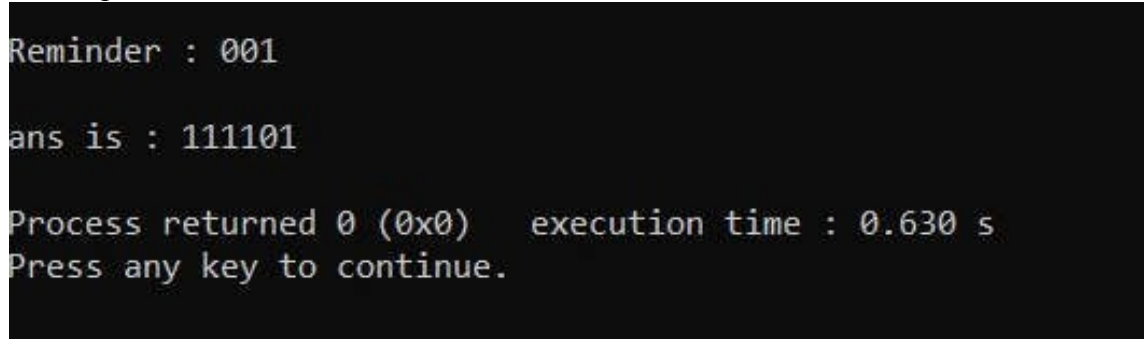
```
#include<iostream>
#include<cstring>
#include<fstream>
#include<windows.h>
using namespace std ;
string ex_or ( string , string );
int main()
{
    string frame , divisor , Reminder , multiply="0",ans ;
    // get frame and divisor      from crcbinary
    f i l e ifstream fin ( " crcbinary . txt " );
    i f ( ! fin )
    {
        cout<<"f i l e not open ";
        exit ( 0 );
    }
    fin>>frame ;
    fin>>divisor ;
    // get      size      of frame and divisor int
    f l =frame . length () ;
    int l=divisor . length () ;
    for ( int i =0 ; i < l ; i ++ ) Reminder . push back ( frame . at ( i ) );
    // for      sender side
    for ( int i =0 ; i < l -1 ; i ++ )
        frame . push back ( ' 0 ' );
    // for      receiver      side
    frame . append ( "" );
    // set      multiply
    for ( int i =1 ; i < l ; i ++ )
        multiply . push back ( ' 0 ' );
    // temporary storage of
    multiply string t=multiply ;
    ans . push back ( Reminder . at ( 0 ) );
    for ( int i =0 ; i < f l ; i ++ )
    {
        i f ( Reminder . at ( 0 ) == ' 1 ' )
            multiply=divisor ;
        else
            multiply=t ;
    }
    //      call function      ex or
```

Data Communication And Networking Practicals

```
Reminder=ex or (Reminder , multiply );
// erase 1st element of reminder
Reminder . erase (0 ,1);
// set answer and reminder i f ( i<fl -1){ ans .
push back (Reminder . at (0));

Reminder . push back (frame . at ( l+i ));
    }
}
cout<<"\nReminder : "<<Reminder<<endl ;
cout<<"\nans is : "<<ans<<endl ;
return 0;
}
string ex or ( string t , string m)
{
string temp ;
for ( int i =0;i<t . length (); i++)
{
i f ( t . at ( i)==m. at ( i ))
temp . push back ( '0 ');
else temp . push back ( '1 ');
}
return temp ;
}
```

Output of CRC:



```
Reminder : 001

ans is : 111101

Process returned 0 (0x0)   execution time : 0.630 s
Press any key to continue.
```

Conclusion:-

Thus, by performing this practical we implemented the LRC and VRC came to know about the data transfer and how it takes place from sender to receiver.