CYCLE-2
LAB-1:Write a program for error detecting code using CRC-CCITT (16-bits).

Program:

```
#include<stdio.h>
char m[50],g[50],r[50],q[50],temp[50];
void caltrans(int);
void crc(int);
void calram();
void shiftl();
int main()
{
int n,i=0;
char ch,flag=0;
printf("Enter the frame bits:");
while((ch=getc(stdin))!='\n')
m[i++]=ch;
n=i;
for(i=0;i<16;i++)
m[n++]='0';
m[n]='\0';
printf("Message after appending 16 zeros:%s",m);
for(i=0;i<=16;i++)
g[i]='0';
g[0]=g[4]=g[11]=g[16]='1';g[17]='\0';
printf("\n generator:%s\n",g);
crc(n);
printf("\n\n quotient:%s",q);
caltrans(n);
printf("\n transmitted frame:%s",m);
printf("\nEnter transmitted frame:");
scanf("\n%s",m);
printf("CRC checking\n");
crc(n);
printf("\n\n last remainder:%s",r);
for(i=0;i<16;i++)
if(r[i]!='0')
flag=1;
else
continue;
if(flag==1)
printf("Error during transmission");
else
printf("\n\nReceived frame is correct");
}
void crc(int n)
{
int i,j;
for(i=0;i<n;i++)
temp[i]=m[i];
for(i=0;i<16;i++)
r[i]=m[i];
printf("\n intermediate remainder\n");
for(i=0;i<n-16;i++)
{
if(r[0]=='1')
```

```
{
q[i]='1';
calram();
}
else
{
q[i]='0';
shiftl();
}
r[16]=m[17+i];
r[17]='\0';
printf("\nremainder %d:%s",i+1,r);
for(j=0;j<=17;j++)
temp[j]=r[j];
}
q[n-16]='\0';
}
void calram()
{
int i,j;
for(i=1;i<=16;i++)
r[i-1]=((int)temp[i]-48)^((int)g[i]-48)+48;
}
void shiftl()
{
int i;
for(i=1;i<=16;i++)
r[i-1]=r[i];
}
void caltrans(int n)
{
int i,k=0;
for(i=n-16;i<n;i++)
m[i]=((int)m[i]-48)^((int)r[k++]-48)+48;
m[i]='\0';
}
```

Output:

C:\Users\prema\Desktop\crc.exe

```
remainder 1:01100100001000010
remainder 2:11001000010000100
remainder 3:10000000101001010
remainder 4:00010001011010110
remainder 5:00100010110101100
remainder 6:01000101101011000
remainder 7:1000101101011000

 quotient:1011000
 transmitted frame:10111011000101101011000
Enter transmitted frame:10111011000101101011000
CRC checking

 intermediate remainder

remainder 1:01100110000011000
remainder 2:11001100000110001
remainder 3:10001000000100001
remainder 4:00000000000000000
remainder 5:00000000000000000
remainder 6:00000000000000000
remainder 7:0000000000000000

 last remainder:0000000000000000

Received frame is correct
Process returned 0 (0x0)   execution time : 23.338 s
Press any key to continue.
```