CRC(16 bit)

```
#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("\ngenerator:%s\n",g);
crc(n);
printf("\n\nquotient:%s",q);
caltrans(n);
printf("\ntransmitted frame:%s",m);
printf("\nEnter transmitted freme:");
scanf("\n%s",m);
printf("CRC checking\n");
crc(n);
```

```
printf("\n\nlast 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 freme 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("\nintermediate 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

```
Enter the frame bits:100010101
Message after appending 16 zeros:10001010100000000000000000
generator:10001000000100001

intermediate remainder

remainder 1:00000101001000010
remainder 2:00001010010000100
remainder 3:00010100100001000
remainder 4:00101001000010000
remainder 5:01010010000100000
remainder 6:10100100001000000
remainder 7:01011000011000010
remainder 8:10110000110000100
remainder 9:0111000110100101
```

```
quotient:100000101
transmitted frame:10001010101110001101001O1
Enter transmitted freme:10001010101110001101001O1
CRC checking

intermediate remainder

remainder 1:00000101010100001
remainder 2:00001010101000010
remainder 3:00010101010000101
remainder 4:00101010100001010
remainder 5:01010101000010100
remainder 6:10101010000101001
remainder 7:01000100000010000
remainder 8:10001000000100001
remainder 9:0000000000000000

last remainder:0000000000000000

Received freme is correct
```