CRC - CCITT (16bit)

Shaveen Ram V.B
1BM18CS091

```
#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("\n Enter transmitted frame");
scanf("%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\n Recieved frame is correct");
}
void crc (int n)
{
int i, j;
for (i=0; i<n; i++)
temp[i] = m[i];
printf ("\n intermediate remainder \n");
for (i=0; i<n-16; i++)
{
if (r[0] == '1')
{
q[i] = '1';
xor ram();
}
else
{
q[i] = '0';
shift l();
}
r[16] = m[17+i];
r[17] = '\0';
printf ("\n remainder %d : %s", i+1, r);
for (j=0; j<=17; j++)
temp[j] = r[j];
}
```

```
    q[n-16] = '\0';
}
void calsum()
{
int i,j;
for(i=1; i<=16; i++)
r[i-1] = ((int)temp[i]-48)^((int)q[i]-48)+48;
}
void shift()
{
int i;
for(i=1; i<=16; i++)
r[i-1] = r[i];
}
void catsum(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';
}
```