

# Lab 10

Samrith Sanjoo.S  
1BM21CS185

**Write a program for error detecting code using  
CRC-  
CCITT (16-bits).**

```
#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';
}
```

```

Enter the frame bits:1011
Message after appending 16 zeros:10110000000000000000
generator:10001000000100001

intermediate remainder

remainder 1:01110000001000010
remainder 2:11100000010000100
remainder 3:11010000101001010
remainder 4:1011000101101011

quotient:1011
transmitted frame:10111011000101101011
Enter transmitted freme:10111011000101101011
CRC checking

intermediate remainder

remainder 1:01100110000011000
remainder 2:11001100000110001
remainder 3:10001000000100001
remainder 4:0000000000000000

last remainder:0000000000000000

Received freme is correct

```

**Write a program for congestion control using Leaky**

**bucket algorithm.**

```

#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
void main() {

```

```

int i,packets[10],content=0,newcontent,time,clk,bcktsize,oprate;
for(i=0;i<5;i+
+)
{ packets[i]=rand()
%10;
if(packets[i]==0) --i;
}
printf("\n Enter output rate of the bucket: \n");
scanf("%d",&oprate); printf("\
n Enter Bucketsize\n");
scanf("%d",&bcktsize);
for(i=0;i<5;++i)
{
if((packets[i]+content)>bcktsize)
{
if(packets[i]>bcktsize)
printf("\n Incoming packet size %d greater than the size of the bucket\
n",packets[i]);
else printf("\n bucket size
exceeded\n");
}
else
{
newcontent=packets[i]; content+=newcontent;
printf("\n Incoming Packet : %d\n",newcontent);
printf("\n Transmission left : %d\n",content);
time=rand()%10;
printf("\n Next packet will come at %d\n",time);
for(clk=0;clk<time && content>0;++clk)
{
printf("\n Left time %d", (time-clk));
if(content)
{
printf("\n Transmitted\n");
if(content<oprate) content=0; else
content=content-oprate; printf("\n Bytes
remaining : %d\n",content);
}
else printf("\n No packets to
send\n"); }
}
}

```

```
}  
}
```

Enter output rate of the bucket:

4

Enter Bucketsize

5

Incoming Packet : 1

Transmission left : 1

Next packet will come at 8

Left time 8

Transmitted

Bytes remaining : 0

Incoming packet size 7 greater than the size of the bucket

Incoming Packet : 4

Transmission left : 4

Next packet will come at 8

Left time 8

Transmitted

Activate \

Go to Setting









