Program No: 1 - CRC-CCITT (16- bits).

```
#include<stdio.h>
#include<string.h>

char data[100],concatdata[117],src_crc[17],dest_crc[17],frame[120],divident[18];
char divisor[18];
char res[17]="0000000000000000";

void crc_cal(int node)
{
int i,j;
for(j=17;j<=strlen(concatdata);j++)
{
        if(divident[0]=='1')
        {
        for(i=1;i<=16;i++)
        if(divident[i]!=divisor[i])
        divident[i-1]='1';
        else
        divident[i-1]='0';
        }

        else
        {
        for(i=1;i<=16;i++)
        divident[i-1]=divident[i];
        }
        if(node==0)
        divident[i-1]=concatdata[j];
        else
        divident[i-1]=frame[j];
```

```
}
divident[i]='\0';
printf("\ncrc is %s\n",divident);
if(node==0)
{
strcpy(src_crc,divident);
}
else
strcpy(dest_crc,divident);
}
int main()
{
int i;
printf("enter the generator bits\n");
gets(divisor);
if(strlen(divisor)<17 || strlen(divisor)>17)
{
printf("please enter the geneartor length min of 17 bits\n");
exit(0);
}
printf("\n At src node :\n Enter the msg to be sent :");
gets(data);
strcpy(concatdata,data);
strcat(concatdata,"0000000000000000");
for(i=0;i<=16;i++)
divident[i]=concatdata[i];
divident[i]='\0';
crc_cal(0);
printf("\ndata is:\t");
puts(data);
printf("\n The frame transmitted is :\t");
printf("\n%s%s",data,src_crc);
```

```
printf("\n\t\tSOURCE NODE TRANSMITTED THE FRAME---->");
printf("\n\n\n\n\t\t\tAT DESTINATION NODE\nenter the recived frame:\t");
gets(frame);
for(i=0;i<=16;i++)
divident[i]=frame[i];
divident[i]='\0';
crc_cal(1);
if((strcmp(dest_crc,res))==0)
printf("\nRecived frame is error free .\n ");
else
printf("\nRecived frame containes one or more error ");
return 1;
}
```

```
enter the generator bits
10001000000100001

 At src node :
 Enter the msg to be sent :0101

crc is 0101000010100101

data is:        0101

 The frame transmitted is :
0101010100000010100101
                SOURCE NODE TRANSMITTED THE FRAME---->



                        AT DESTINATION NODE
enter the recived frame:        0101010100000010100101

crc is 0000000000000000

Recived frame is error free .


...Program finished with exit code 1
Press ENTER to exit console.
```

Program No: 2 -Distance vector algorithm

```
#include<stdio.h>
struct rtable
{
int dist[20],nextnode[20];
}table[20];
 int cost[10][10],n;
 void distvector()
{
int i,j,k,count=0;
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{
table[i].dist[j]=cost[i][j];
table[i].nextnode[j]=j;
}          }
do
{
count=0;
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{
for(k=0;k<n;k++)
{
if(table[i].dist[j]>cost[i][k]+table[k].dist[j])
{
table[i].dist[j]=table[i].dist[k]+table[k].dist[j];
table[i].nextnode[j]=k;
```

```
count++;
}
}
}
}
}while(count!=0);
}
int main()
{
int i,j;
printf("\nenter the no of vertices:\t");
scanf("%d",&n);
printf("\nenter the cost matrix\n");
for(i=0;i<n;i++)
for(j=0;j<n;j++)
scanf("%d",&cost[i][j]);
distvector();
for(i=0;i<n;i++)
{
printf("\nstate value for router %c \n",i+65);
printf("\ndestnode\tnextnode\tdistance\n");
for(j=0;j<n;j++)
{
if(table[i].dist[j]==99)
printf("%c\t\t-\t\t infinite\n",j+65);
else
printf("%c\t\t%c\t\t%d\n",j+65,table[i].nextnode[j]+65,table[i].dist[j]);
}
}
return 0;

 }
```

```
enter the no of vertices:       3

enter the cost matrix
0 1 5
1 0 2
5 2 0

state value for router A

destnode        nextnode        distance
A               A               0
B               B               1
C               B               3

state value for router B

destnode        nextnode        distance
A               A               1
B               B               0
C               C               2

state value for router C

destnode        nextnode        distance
A               B               3
B               B               2
C               C               0


...Program finished with exit code 0
Press ENTER to exit console.
```

Code for TCP socket Server

Program No: 3(a) Server

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
```

```
#include<netinet/in.h>
#include<sys/fcntl.h>
#include<stdlib.h>
int main(int argc,char *argv[])
{
int fd,sockfd,newsockfd,clilen,portno,n;
struct sockaddr_in seradd,cliadd;
char buffer[4096];
if(argc<2)
{
fprintf(stderr,"\n\n No port\n");
exit(1);
}
portno=atoi(argv[1]);
sockfd=socket(AF_INET,SOCK_STREAM,0);
if(sockfd<0)
error("\n error opening socket.\n");
bzero((char *)&seradd,sizeof(seradd));
seradd.sin_family=AF_INET;
seradd.sin_addr.s_addr=(htonl)INADDR_ANY;
seradd.sin_port=htons(portno);
if(bind(sockfd,(struct sockaddr *)&seradd,sizeof(seradd))<0)
perror("\n IP addr cannt bind");
listen(sockfd,5);
clilen=sizeof(cliadd);
printf("\n Server waiting for clint request\n");
while(1)
{
newsockfd=accept(sockfd,(struct sockaddr *)&cliadd,&clilen);
if(newsockfd<0)
perror("\n Server cannot accept connection request ");
bzero(buffer,4096);
```

```
read(newsockfd,buffer,4096);
fd=open(buffer,O_RDONLY);
if(fd<0)
perror("\n File  doesnot exist");
while(1)
{
n=read(fd,buffer,4096);
if(n<=0)
exit(0);
write(newsockfd,buffer,n);
printf("\n File transfer completet\n");
}
close(fd);
close(newsockfd);
}
return 0;}
```

Program No: 3(b) Client

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<sys/fcntl.h>
#include<stdlib.h>
#include<string.h>
#include<arpa/inet.h>

int main(int argc,char *argv[])
{
int sockfd,portno,n;
struct sockaddr_in seradd;
char buffer[4096],*serip;
```

```
if(argc<4)
{
fprintf(stderr,"usage %s serverip filename port",argv[0]);
exit(0);
}
serip=argv[1];
portno=atoi(argv[3]);
sockfd=socket(AF_INET,SOCK_STREAM,0);
if(sockfd<0)
perror("\n Error in creating socket.\n");
perror("\n Client on line.");
bzero((char *)&seradd,sizeof(seradd));
seradd.sin_family=AF_INET;
seradd.sin_addr.s_addr=inet_addr(serip);
seradd.sin_port=htons(portno);
if(connect(sockfd,(struct sockaddr *)&seradd,sizeof(seradd))<0)
perror("\n Error in connection setup \n");
write(sockfd,argv[2],strlen(argv[2])+1);
bzero(buffer,4096);
n=read(sockfd,buffer,4096);
if(n<=0)
{
perror("\n File not found");
exit(0);
}
write (1,buffer,n);
}
```

Program No: 4(a) Server

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<fcntl.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<unistd.h>
#define FIFO1 "fifo1"
#define FIFO2 "fifo2"

int main()
{
char p[100],c[5000,ch;
int num,fd,fd2,f1;
mknod(FIFO1,S_IFIFO|0666,0);
mknod(FIFO2,S_IFIFO|0666,0);
printf("\n Server online...\n");
fd=open(FIFO1,O_RDONLY);
fd2=open(FIFO2,O_WRONLY);
printf("Server online\n waiting for client \n\n");
if((num=read(fd,p,100))==-1)
perror("\n Read Error ");
else
{
p[num]='\0';
printf("\n File is %s .\n",p);
if((f1=open(p,O_RDONLY))<0)
{
write(fd2,"File not found",15);
return 1;
}
else
```

```
{
stdin=fdopen(f1,"r");
num=0;
while((ch=fgetc(stdin))!=EOF)
c[num++]=ch;
c[num]=0;
printf(" Server: Transfering the contents of :%s ",p);
if(num=write(fd2,c,strlen(c))==-1)
printf("\n Error in writting to FIFO\n");
else
printf("\n File transfer completed \n");
}}}
```

Program No: 4(b) Client

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<fcntl.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<unistd.h>
#define FIFO1 "fifo1"
#define FIFO2 "fifo2"

int main()
{
char p[100],c[5000];
int num,fd,fd2,f1;
mknod(FIFO1,S_IFIFO|0666,0);
mknod(FIFO2,S_IFIFO|0666,0);
printf("\n Client online...\n");
fd=open(FIFO1,O_WRONLY);
fd2=open(FIFO2,O_RDONLY);
```

```
printf("Client : Enter the filename . \n\n");
scanf("%s",p);
num=write(fd,p,strlen(p));
if(num==-1)
{
perror("\nWrite Error.\n");
return 1;
}
else
{
printf("\n Waiting for reply\n");
if((num=read(fd2,c,5000))==-1)
perror("\nError while reading from fifo \n");
else
{
c[num]=0;
printf("%s",c);
}}
return 1;
}
```