**PART B**

**1. Write a program for error detecting code using CRC CCIT(16 bits).**

#include<stdio.h>

#include<string.h>

#define N strlen(g)

char t[128],cs[128],g[]="10001000000100001";

int a,e,c;

void xor()

{

for(c=1;c<N;c++)

cs[c]=((cs[c] == g[c])?'0':'1');

}

void crc()

{

for(e=0;e<N;e++)

cs[e]=t[e];

do{

if(cs[0]=='1')

xor();

for(c=0;c<N-1;c++)

cs[c]=cs[c+1];

cs[c]=t[e++];

}while(e<=a+N-1);

}

int main()

{

int flag=0;

system("clear");

printf("Enter polynomial:\n");

scanf("%s",&t);

printf("Generated divisor is :%s\n",g);

a=strlen(t);

for(e=a;e<a+N-1;e++)

t[e]='0';

printf("Modified polynomial is %s\n",t);

crc();

printf("Checksum is :%s\n",cs);

for(e=a;e<a+N-1;e++)

t[e]=cs[e-a];

printf("Final codeword is %s\n",t);

printf("Test error detection?\n1(yes) 0(no):\n");

scanf("%d",&flag);

if(flag==0)

return;

if(flag==1)

{

printf("Enter position where error is to be inserted:\n ");

scanf("%d",&e);

if(e<a+N-1){ t[e]=(t[e]=='0')?'1':'0'; printf("err data : %s\n",t);

}

else

printf("Position where error should be created is above the length of

Codeword...Hence\n");

}

crc();

for(e=0;(e<N-1)&&(cs[e]!='1');e++);

if(e<N-1)

printf("Error detected\n");

else

printf("No error detected \n");

return;

}

**OUTPUT:**

[[root@god](about:blank) ~]# vi crc.c

[[root@god](about:blank) ~]# cc crc.c

[[root@god](about:blank) ~]# ./a.out

Enter polynomial:

11101

Generated divisor is :10001000000100001

Modified polynomial is 111010000000000000000

Checksum is :1100001110011100

Final codeword is 111011100001110011100

Test error detection?

1(yes) 0(no):

1

Enter position where error is to be inserted:

2

err data : 110011100001110011100

Error detected

[[root@god](about:blank) ~]# ./a.out

Enter polynomial:

11101

Generated divisor is :10001000000100001

Modified polynomial is 111010000000000000000

Checksum is :1100001110011100

Final codeword is 111011100001110011100

Test error detection?

1(yes) 0(no):

1

Enter position where error is to be inserted:

21

Position where error should be created is above the length of Codeword...Hence

No error detected

[[root@god](about:blank) ~]#

**2.Write a program for frame sorting techniques used in buffers.**

#include<stdio.h>

#include<stdlib.h>

struct frame

{

int fslno;

char finfo[20];

};

struct frame arr[20];

int n;

void sort()

{

int i,j;

struct frame temp;

for(i=0;i<n-1;i++)

{

for(j=0;j<n-i-1;j++)

if(arr[j].fslno>arr[j+1].fslno)

{

temp=arr[j];

arr[j]=arr[j+1];

arr[j+1]=temp;

}

}

}

int main()

{

int i;

system("clear");

printf("Enter the number of frames:\n");

scanf("%d",&n);

printf("Enter frame sequence number and frame elements:\n");

for(i=0;i<n;i++)

scanf("%d%s",&arr[i].fslno,&arr[i].finfo);

sort();

printf("The frames in sequence are....\n");

for(i=0;i<n;i++)

printf(" %d\t %s\n",arr[i].fslno,arr[i].finfo);

}

**OUTPUT:**

[[root@god](about:blank) ~]# vi frame.c

[[root@god](about:blank) ~]# cc frame.c

[[root@god](about:blank) ~]# ./a.out

Enter the number of frames:

4

Enter frame sequence number and frame elements:

4 Technology

1 Cambridge

3 of

2 Institute

The frames in sequence are....

1 Cambridge

2 Institute

3 of

4 Technology

[[root@god](about:blank) ~]#

**3. Write a program for distance vector algorithm to find suitable path for transmission.**

#include<stdio.h>

#include<stdlib.h>

#include<ctype.h>

int min(int a,int b)

{

return (a<b)?a:b;

}

void dist(int p[10][10],int n)

{

int i,j,k;

for(k=1;k<=n;k++) for(j=1;j<=n;j++) for(i=1;i<=n;i++)

p[i][j]=min(p[i][j],p[i][k]+p[k][j]);

}

int main()

{

int a[10][10],i,j,n,flag,temp=1; printf("Enter number of nodes:\n"); scanf("%d",&n);

printf("Enter the matrix 0 for self loop and 999 for no path\n");

for(i=1;i<=n;i++) for(j=1;j<=n;j++) scanf("%d",&a[i][j]);

for(i=1;i<=n;i++)

{

printf("Initial record of %d is\n",i);

for(j=1;j<=n;j++)

printf("%d to %d is %d \n",i,j,a[i][j]);

}

dist(a,n);

for(i=1;i<=n;i++)

{

printf("Updated record of %d is \n",i);

for(j=1;j<=n;j++)

printf("%d to %d is %d\n",i,j,a[i][j]);

}

while(temp)

{

do{

printf("Enter 0 to find shortest path and 1 to exit\n");

scanf("%d",&flag);

if(flag==0)

{

printf("Enter 2 nodes\n");

scanf("%d %d",&i,&j);

printf("%d to %d is %d\n",i,j,a[i][j]);

}

else temp=0;

break;

}while(flag!=0);

}

exit(0);

}

**OUTPUT:**

[[root@god](about:blank) ~]# vi dva.c

[[root@god](about:blank) ~]# cc dva.c

[[root@god](about:blank) ~]# ./a.out

Enter number of nodes:

5

Enter the matrix 0 for self loop and 999 for no path

0 1 2 3 999

1 0 999 4 0

2 999 0 7 999

3 4 7 0 5

999 0 999 5 0

Initial record of 1 is

1 to 1 is 0

1 to 2 is 1

1 to 3 is 2

1 to 4 is 3

1 to 5 is 999

Initial record of 2 is

2 to 1 is 1

2 to 2 is 0

2 to 3 is 999

2 to 4 is 4

2 to 5 is 0

Initial record of 3 is

3 to 1 is 2

3 to 2 is 999

3 to 3 is 0

3 to 4 is 7

3 to 5 is 999

Initial record of 4 is

4 to 1 is 3

4 to 2 is 4

4 to 3 is 7

4 to 4 is 0

4 to 5 is 5

Initial record of 5 is

5 to 1 is 999

5 to 2 is 0

5 to 3 is 999

5 to 4 is 5

5 to 5 is 0

Updated record of 1 is

1 to 1 is 0

1 to 2 is 1

1 to 3 is 2

1 to 4 is 3

1 to 5 is 1

Updated record of 2 is

2 to 1 is 1

2 to 2 is 0

2 to 3 is 3

2 to 4 is 4

2 to 5 is 0

Updated record of 3 is

3 to 1 is 2

3 to 2 is 3

3 to 3 is 0

3 to 4 is 5

3 to 5 is 3

Updated record of 4 is

4 to 1 is 3

4 to 2 is 4

4 to 3 is 5

4 to 4 is 0

4 to 5 is 4

Updated record of 5 is

5 to 1 is 1

5 to 2 is 0

5 to 3 is 3

5 to 4 is 4

5 to 5 is 0

Enter 0 to find shortest path and 1 to exit

0

Enter 2 nodes

4 5

4 to 5 is 4

Enter 0 to find shortest path and 1 to exit

0

Enter 2 nodes

2 5

2 to 5 is 0

Enter 0 to find shortest path and 1 to exit

1

[[root@god](about:blank) ~]#

**4. Using TCP/IP sockets, write a client – server program to make the client send the file name and to make the server send back the contents of the requested file if present.**

**SERVER CODE:**

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<errno.h>

#include<string.h>

#include<sys/types.h>

#include<sys/stat.h>

#include<fcntl.h>

#include<netinet/in.h>

#include<sys/socket.h>

#include<arpa/inet.h>

#include<sys/wait.h>

#include<signal.h>

#define MYPORT 6490

#define BACKLOG 10 int main(void)

{

int sockfd,fp,new\_fd;

struct sockaddr\_in my\_addr,their\_addr;

int sin\_size,i=0;

int yes=1;

char buf1[20],buf2[20000];

if((sockfd=socket(AF\_INET,SOCK\_STREAM,0))==-1)

{

perror("socket");

exit(1);

}

if(setsockopt(sockfd,SOL\_SOCKET,SO\_REUSEADDR,&yes,sizeof(int))==-1)

{

perror("setsockopt");

exit(1);

}

my\_addr.sin\_family=AF\_INET; my\_addr.sin\_port=htons(MYPORT); my\_addr.sin\_addr.s\_addr=INADDR\_ANY; memset(&(my\_addr.sin\_zero), '\0', 8);

if(bind(sockfd,(struct sockaddr \*)&my\_addr, sizeof(struct sockaddr)) ==-1)

{

perror("Bind");

exit(1);

}

if(listen(sockfd, BACKLOG) == -1)

{

perror("listen");

exit(1);

}

printf("\n SERVER is online! \n SERVER: Waiting for the client........\n");

sin\_size=sizeof(struct sockaddr\_in);

if((new\_fd=accept(sockfd,(struct sockaddr \*)&their\_addr, &sin\_size))==-1)

{

perror("Accept");

exit(0);

}

printf("\n SERVER: Got connection from %s \n", inet\_ntoa(their\_addr.sin\_addr));

recv(new\_fd,&buf1,sizeof(buf1),0);

printf("File requested is %s\n", buf1);

if((fp=open(buf1,O\_RDONLY))<0)

{

printf("File not found\n");

strcpy(buf2,"File not found");

}

else

{

}

printf("SERVER: %s found and ready to transfer.\n",buf1);

read(fp,&buf2,20000);

close(fp);

send(new\_fd,&buf2,sizeof(buf2),0);

close(new\_fd);

close(sockfd);

printf("Transfer success \n");

printf("\n");

return 0;

}

**CLIENT CODE:**

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<errno.h>

#include<string.h>

#include<netdb.h>

#include<sys/types.h>

#include<netinet/in.h>

#include<sys/socket.h>

#include<fcntl.h>

#define PORT 6490 int main()

{

int i=0,sockfd,len;

char buf1[40],buf2[20000]; FILE\* fp;

struct sockaddr\_in their\_addr;

if((sockfd=socket(AF\_INET, SOCK\_STREAM,0))==-1)

{

perror("socket");

exit(1);

}

their\_addr.sin\_family=AF\_INET; their\_addr.sin\_port=htons(PORT); their\_addr.sin\_addr.s\_addr=inet\_addr("127.0.0.1"); memset(&(their\_addr.sin\_zero), '\0', 8);

if(connect(sockfd, (struct sockaddr \*)&their\_addr, sizeof(struct sockaddr))==-1)

{

perror("connect");

exit(1);

}

printf("CLIENT is online!\n");

printf("CLIENT:Enter the filename to be displayed: ");

scanf("%s",buf1);

send(sockfd,buf1,sizeof(buf1),0);

if(recv(sockfd,buf2,sizeof(buf2),0)==1)

{

}

else

{

}

perror("recv");

exit(1);

printf("Displyaing the contents of %s",buf1);

printf("\n%s\n",buf2);

close(sockfd);

return 0;

}

**OUTPUT:**

[[root@god](about:blank) ~]# cc sockser.c

[[root@god](about:blank) ~]# ./a.out

SERVER is online!

SERVER: Waiting for the client........

SERVER: Got connection from 127.0.0.1

File requested is /root/sample

SERVER: /root/sample found and ready to transfer. Transfer success

[[root@god](about:blank) ~]#

[[root@god](about:blank) ~]# vi sockc.c

[[root@god](about:blank) ~]# cc sockc.c

[[root@god](about:blank) ~]# ./a.out

CLIENT is online!

CLIENT:Enter the filename to be displayed: /root/sample

Displyaing the contents of /root/sample

This is Network programming lab

[[root@god](about:blank) ~]#

**5. Implement the above program using as message queues or FIFOs as IPC channels.**

**SERVER CODE:**

#include<stdio.h>

#include<stdlib.h>

#include<errno.h>

#include<string.h>

#include<fcntl.h>

#include<sys/types.h>

#include<sys/stat.h>

#include<unistd.h>

#define FIFO1\_NAME "fifo1"

#define FIFO2\_NAME "fifo2"

int main()

{

char p[100],f[100],c[300],r,ch;

int num1,num2,f1,fd1,fd2,i=0;

mknod(FIFO1\_NAME,S\_IFIFO|0666,0);

mknod(FIFO2\_NAME,S\_IFIFO|0666,0);

printf("Server Online!!!!!!!\n");

fd1=open(FIFO1\_NAME,O\_RDONLY);

printf("Client Online!!!!\n Waiting for request..........\n\n");

while(1)

{

if((num1=read(fd1,p,100))==-1)

perror("Read Error\n");

else

{

p[num1]='\0';

if((f1=open(p,O\_RDONLY))<0)

{

}

else

{

printf("\n Server %s not found!!!!\n",p);

exit(1);

printf("Server %s found!!!\n Transferring contents............\n",p); stdin=fdopen(f1,"r"); while((ch=(char)getc(stdin))!=EOF)

{

c[i]=ch;

i++;

}

perror("Transfer error\n");

else

c[i]='\0'; fd2=open(FIFO2\_NAME,O\_WRONLY); printf("File content in server is\n %s\n",c); if(num2=write(fd2,c,strlen(c))==-1)

printf("Transfer completed......\n");

exit(1);

}

}

}

}

**CLIENT CODE:**

#include<stdio.h>

#include<stdlib.h>

#include<errno.h>

#include<string.h>

#include<fcntl.h>

#include<sys/stat.h>

#include<unistd.h>

#define FIFO1\_NAME "fifo1"

#define FIFO2\_NAME "fifo2" int main()

{

char p[100],f[100],c[300]; int num1,num2,f1,fd1,fd2; mknod(FIFO1\_NAME,S\_IFIFO|0666,0); mknod(FIFO2\_NAME,S\_IFIFO|0666,0); printf("Waiting for Server.....!!\n\n"); fd1=open(FIFO1\_NAME,O\_WRONLY); printf("File descriptor fd is %d\n\n",fd1); if(fd1<0)

{

}

else

printf("File Open Error....\n");

getchar();

exit(1);

printf("\n Server Online....\n Client: Enter the path\n\n");

scanf("%s",p);

while(!feof(stdin))

{

if((num1=write(fd1,p,strlen(p)))==-1)

perror("Write Error...\n");

else

{

printf("Waiting for reply....\n"); fd2=open(FIFO2\_NAME,O\_RDONLY); if((num2=read(fd2,c,300))==-1) perror("Transfer Error.....\n");

else

{

printf("File received!!!\n Displaying the contents.....!!!\n\n");

printf("%s",c);

exit(1);

}

}

}

}

**OUTPUT:**

(Note: Create a new file named sample and type some lines in it. Now open two terminals

simultaneously and execute server in one window and then client in another window.)

[[root@god](about:blank) ~]# cc fserver.c [[root@god](about:blank) ~]# ./a.out Server Online!!!!!!!

Client Online!!!!

Waiting for request..........

Server /root/sample found!!! Transferring contents............ File content in server is

This is Network programming lab

Transfer completed...... [[root@god](about:blank) ~]#

[[root@god](about:blank) ~]# cc fclient.c [[root@god](about:blank) ~]# ./a.out Waiting for Server.....!!

File descriptor fd is 3

Server Online.... Client: Enter the path

/root/sample Waiting for reply.... File received!!!

Displaying the contents.....!!!

This is Network programming lab

[[root@god](about:blank) ~]

**6. Write a program for simple RSA algorithm to encrypt and decrypt the data.**

#include<stdio.h>

#include<stdlib.h>

#include<ctype.h>

#include<string.h>

long int e,d,n;

long int val[50];

char decode(long int ch)

{

int i;

long int temp=ch;

for(i=1;i<d;i++)

ch=temp \*ch%n;

return(ch);

}

int gcd(long a,long b)

{

if(b==0)

return (a);

else

return (gcd(b,a%b));

}

int prime(int a)

{

int i;

for(i=2;i<a;i++)

{

if((a%i)==0)

return 0;

}

return 1;

}

int encode(char ch)

{

int i;

long int temp=ch;

for(i=1;i<e;i++) temp=temp\*ch%n; return temp;

}

int main()

{

int i;

long int p,q,phi,c[50];

char text[50],ctext[50];

printf("Enter text to be encoded:\n");

scanf("%[^\n]",text);

printf("%s\n",text);

do

{

p=rand()%30;

}while(!prime(p));

do

{

q=rand()%30;

}while(!prime(q));

n=p\*q;

phi=(p-1)\*(q-1);

printf("\n p=%d\t q=%d\t n=%d\t phi=%d\n",p,q,n,phi);

do

{

e=rand()%phi;

}while(!gcd(e,phi));

do

{

d=rand()%phi;

}while((d\*e%phi)!=1);

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*Encoding message\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

sleep(2);

for(i=0;text[i]!='\0';i++)

val[i]=encode(text[i]);

val[i]=-999;

printf("\nEncoded message is.............\n");

for(i=0;val[i]!=-999;i++)

printf("%ld",val[i]);

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*Decoding message\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

sleep(2);

for(i=0;val[i]!=-999;i++)

ctext[i]=decode(val[i]);

ctext[i]='\0';

printf("\nDecoded message is.............\n");

printf("%s\n",ctext);

}

**OUTPUT:**

[[root@god](about:blank) ~]# vi rsa.c

[[root@god](about:blank) ~]# cc rsa.c [[root@god](about:blank) ~]# ./a.out Enter text to be encoded:

Welcome to Cambridge Institute of Technology

Welcome to Cambridge Institute of Technology

p=13 q=23 n=299 phi=264

\*\*\*\*\*\*\*\*\*\*\*\*Encoding message\*\*\*\*\*\*\*\*\*\*\*\*\*\* Encoded message is.............

165751478322726875981162279876111268123114248481947598187280184116248116

3911675982272898228758313280227147227194238

\*\*\*\*\*\*\*\*\*\*\*\*Decoding message\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Decoded message is.............

Welcome to Cambridge Institute of Technology

[[root@god](about:blank) ~]#

**7. Write a program for Hamming code generation for error detection and correction.**

#include<stdio.h>

#include<math.h>

#include<stdlib.h>

int main()

{

int i,j,k,count,err\_pos=0,flag=0;

char dw[20],cw[20],data[20];

printf("Enter data as binary bit stream (7 bits):\n");

scanf("%s",data);

for(i=1,j=0,k=0;i<12;i++)

{

if(i==(int)pow(2,j))

{

}

else

{

}

}

dw[i]=' ? ';

j++;

dw[i]=data[k];

k++;

for(i=0;i<4;i++)

{

count=0;

for(j=(int)pow(2,i);j<12;j+=(int)pow(2,i))

{

for(k=0;k<(int)pow(2,i);k++)

{

if(dw[j]=='1')

count++;

j++;

}

}

if(count%2==0)

else

}

dw[(int)pow(2,i)]='0';

dw[(int)pow(2,i)]='1';

printf("\n Code word is\n\n");

for(i=1;i<12;i++)

printf("%c",dw[i]);

printf("\n\n Enter the received hamming code\n\n");

scanf("%s",cw);

for(i=12;i>0;i--)

cw[i]=cw[i-1];

for(i=0;i<4;i++)

{

count=0;

for(j=(int)pow(2,i);j<12;j+=(int)pow(2,i))

{

for(k=0;k<(int)pow(2,i);k++)

{

if(cw[j]=='1')

count++;

j++;

}

}

if(count%2!=0)

err\_pos=err\_pos+(int)pow(2,i);

}

if(err\_pos==0)

printf("\n\n There is no error in received code word\n");

else

{

if(cw[err\_pos]==dw[err\_pos])

{

printf("\n\n There are 2 or more errors in received code....\n\n");

printf("SORRy....!Hamming code cannot correct 2 or more errors.....\n");

flag=1;

}

else

printf("\nThere is an error in bit position %d of received code word\n\n",err\_pos);

if(flag==0)

{

cw[err\_pos]=(cw[err\_pos]=='1') ? '0' : '1';

printf("\n\n Corrected code word is \n\n");

for(i=1;i<12;i++)

printf("%c",cw[i]);

}

}

printf("\n\n");

}

**OUTPUT:**

[[root@god](about:blank) ~]# vi hamming.c

[[root@god](about:blank) ~]# gcc hamming.c -lm

[[root@god](about:blank) ~]# ./a.out

Enter data as binary bit stream (7 bits):

1110110

Code word is

11101100110

Enter the received hamming code

10101100110

There is an error in bit position 2 of received code word

Corrected code word is

11101100110 [[root@god](about:blank) ~]# ./a.out

Enter data as binary bit stream (7 bits):

1110110

Code word is

11101100110

Enter the received hamming code

00101100110

There are 2 or more errors in received code.... SORRy....!Hamming code cannot correct 2 or more errors.....

[[root@god](about:blank) ~]#

**8. Write a program for congestion control using leaky bucket algorithm.**

#include<stdio.h>

#include<string.h>

struct frame

{

char msg[20];

int seq;

}fr[40];

int fno,front=0,rear=-1,count=0;

char q[10][20];

void insert(int i)

{

if(count==3)

{

}

else

{

printf("Bucket is full......\n"); printf("\nPacket lost is %s\n",fr[i].msg); sleep(3);

rear=(rear+1)%3; strcpy(q[rear],fr[i].msg); count++;

printf("\n Inserted packet into the bucket is %s\n",fr[i].msg);

sleep(5);

}

}

void transmitted()

{

if(count==0)

{

printf("\n\nBucket is empty\n");

return;

}

printf("\n\nTransmitted packet is %s\n",q[front]);

sleep(5);

front=(front+1)%3;

count--;

}

int main()

{

int i,j,k,arrival[40],clk=20,n;

char str[100];

printf("Enter the message\n");

scanf("%[^\n]",str);

for(i=0;str[i]!='\0';)

{

fr[fno].seq=fno; for(j=i,k=0;str[j]!='\0'&& k<5;j++,k++) fr[fno].msg[k]=str[j]; fr[fno].msg[k]='\0';

fno++;

i=j;

}

printf("Show the packets\n");

for(i=0;i<fno;i++)

printf("\n Packet %d is : %s\n",i,fr[i].msg);

sleep(5);

for(i=0;i<fno;i++)

{

n=random()%3;

arrival[i]=arrival[i-1]+n;

}

printf("Arrival time of packets is.....\n");

for(i=0;i<fno;i++)

printf("Packet %d %s : %d\n", i, fr[i].msg,arrival[i]);

printf("Size of bucket is 3\n\n");

sleep(5);

i=0;

while(i<fno || count!=0)

{

printf("CLK::%d\n",clk);

while(clk==arrival[i])

{

insert(i);

i++;

}

if((clk%5)==0) transmitted(); clk++;

}

}

**OUTPUT:**

[[root@god](about:blank) ~]# cc leaky.c

[[root@god](about:blank) ~]# ./a.out

Enter the message

Welcome to Network lab

Show the packets

Packet 0 is : Welco Packet 1 is : me to Packet 2 is : Netw Packet 3 is : ork l

Packet 4 is : ab

Arrival time of packets is..... Packet 0 Welco : 21

Packet 1 me to : 22

Packet 2 Netw : 22

Packet 3 ork l : 23

Packet 4 ab : 25

Size of bucket is 3

CLK::20

Bucket is empty

CLK::21

Inserted packet into the bucket is Welco

CLK::22

Inserted packet into the bucket is me to

Inserted packet into the bucket is Netw

CLK::23

Bucket is full......

Packet lost is ork l

CLK::24

CLK::25

Bucket is full......

Packet lost is ab

Transmitted packet is Welco

CLK::26

CLK::27

CLK::28

CLK::29

CLK::30

Transmitted packet is me to

CLK::31

CLK::32

CLK::33

CLK::34

CLK::35

Transmitted packet is Netw

[[root@god](about:blank) ~]#