

CN LAB MANUAL

PART A

FRAMING

1 A) Design and implement C program/ python /java in which sender module should count the number of bytes in the frame and reciever module should display the each frame recieved.

PYTHON IMPLEMENTATION

```
s=int(input("Enter the number of frames "))
se=[]
re=[]
def sender(s):
    print("SENDER")
    for i in range(0,s,1):
        f=str(input("Enter the frame "))
        se.append(f)
        ele=0
        X=0
        for j in se:
            print("Length of Frame",ele+1,"is",len(se[ele]))
            X=X+len(se[ele])
            ele=ele+1
        print("Total length of all frames", X)
    sender(s)
def reciever(se):
    print("RECIEVER")
    re=se.copy()
    print("The frames that were recieved are: ")
    el=0
    for i in re:
        print(re[el])
        el=el+1
    reciever(se)
```

OUTPUT

```
Enter the number of frames 3
SENDER
Enter the frame VARUN
Enter the frame R
Enter the frame RAO
Length of Frame 1 is 5
Length of Frame 2 is 1
Length of Frame 3 is 3
Total length of all frames 9
RECIEVER
The frames that were recieved are:
VARUN
R
RAO
```

1 B) Design a program to implement bit stuffing, encoding and decoding concept in data link layer

*Python Implementation

```
bits=[]
n=int(input("Enter number of bits"))
for j in range(0,n):
    ele=int(input("Enter the bit"))
    bits.append(ele)
print("Entered bits",bits)
stuffed=[]
count=0
for i in range(len(bits)):
    if bits[i]==1:
        count=count+1
        stuffed.append(bits[i])
    elif bits[i]!=1:
        count=0
        stuffed.append(bits[i])
    if count==5:
        stuffed.insert(i+1,0)
stuffed.append("01111110")
stuffed.insert(0,"01111110")
print("Result after bitstuffing")
for i in stuffed:
    print(i,end="")
c=0
recstu=stuffed[1:-1]
for j in range(len(recstu)-1):
    if recstu[j]==1:
        c=c+1
    elif recstu[j]!=1:
        c=0
    if c==5:
        recstu.pop(j+1)
print("\nBit recieved from Sender")
for i in recstu:
    print(i,end="")
```

OUTPUT

```
Enter number of bits4
Enter the bit1
Enter the bit0
Enter the bit1
Enter the bit0
Entered bits [1, 0, 1, 0]
Result after bitstuffing
01111110101001111110
Bit recieved from Sender
1010|
```

ERROR CONTROL

2)Design and implement CRC error detection method used in data link layer

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
int i,j,keylen,msglen,ch,f;
char data[100],key[30],temp[30],rem[30],zero[30],div[30],input[100];
void crc()
{
    for (i=0;i<keylen;i++)
        temp[i]=data[i];
    for (i=0;i<msglen;i++)
    {
        if(temp[0]!='0')
            strcpy(key,zero);
        else
            strcpy(key,div);
        for (j=0;j<keylen-1;j++)
            rem[j]=(temp[j+1]==key[j+1])?'0':'1';
        rem[j]=data[i+keylen];
        strcpy(temp,rem);
    }
}
void main()
{
    printf("Enter Data: ");
    gets(input);
    strcpy(data,input);
    printf("Enter divisor : ");
    gets(div);
    keylen=strlen(div);
    msglen=strlen(input);
    for (i=0;i<keylen-1;i++)
        data[msglen+i]='0';
    printf("\n data after appending zeros %s",data);
    for (i=0;i<keylen;i++)
        zero[i]='0';
    crc();
    strcat(input,rem);
    strcpy(data,input);
    printf("\nRemainder is %s",rem);
    printf("\ndata after appending remainder: %s",data);
    printf("\ndo you want make error(Y/N)?");
    scanf("%c",&ch);
    if(ch=='Y' || ch=='y')
        data[msglen/2]=(data[msglen/2]!='0')?'1':'0';
    crc();
}
```

```

printf("\ndata obtained %s",data);
printf("\nRemainder is %s\n",rem);
f=1;
for (i=0;i<keylen-1;i++)
if(rem[i]!='0'){
f=0;
break;
}
if(f==1){
printf("No Error Occurred final data is ");
for (i=0;i<msglen;i++)
printf("%c",data[i]);
}
else
printf("Error Occurred");

```

OUTPUT

```

Enter Data: 10011001
Enter divisor : 1011
data after appending zeros 10011001000
Remainder is 100
data after appending remainder: 10011001100
do you want make error(Y/N)?Y
data obtained 10010001100
Remainder is 101
Error Occurred|

```

Socket Programming

3A) Design a program to implement client server model (TCP) using socket programming.

client

```
#include<stdio.h>
#include<fcntl.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
#define PORT_ID 8000
int main()
{
    char buf[30000];
    int fd1,n;
    struct sockaddr_in s;
    system("clear");
    printf("Enter the filename to be inserted \n");
    scanf("%s",buf);

    s.sin_family=AF_INET;
    s.sin_port=htons(PORT_ID);
    s.sin_addr.s_addr=inet_addr("127.0.0.1");

    fd1=socket(AF_INET,SOCK_STREAM,0);

    if ((connect(fd1,(struct sockaddr *)&s,sizeof(struct sockaddr)))== -1)
        printf("error in socket binding!!!!\n");

    send(fd1,buf,strlen(buf),0);

    printf("*****Contents of the requested file is*****\n\n");
    while((n=recv(fd1,buf,sizeof(buf),0))>0)
    {
        buf[n]='\0';
        printf("%s",buf);
    }
    printf("\n");
    close(fd1);
    return 0;
}
```

```
}
```

server

```
#include<stdio.h>

#include<fcntl.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
#define PORT_ID 8000

int main()
{
    char buf[300];
    int fd1,fd2,n,size;
    struct sockaddr_in s;
    system("clear");
    printf("server is getting ready.....\n");

    s.sin_family=AF_INET;
    s.sin_port=htons(PORT_ID);
    s.sin_addr.s_addr=inet_addr("127.0.0.1");

    fd1=socket(AF_INET,SOCK_STREAM,0);

    if((bind(fd1,(struct sockaddr *)&s,sizeof(struct sockaddr)))==-1)
        printf("Error in socket binding!!!\n");
    if((listen(fd1,5))===-1)
        printf("Error in listening!!!\n");
    printf("Waitting for client request.....\n");
    size=sizeof(struct sockaddr);
    fd2=accept(fd1,(struct sockaddr *)&s,&size);
    size=recv(fd2,buf,sizeof(buf),0);
    buf[size]='\0';
    printf("File name RECIEVED is:%s\n",buf);

    if((fd1=open(buf,O_RDONLY))!=-1)
    {
        while((n=read(fd1,buf,sizeof(buf)))>0)
            send(fd2,buf,n,0);
    }
    else
        send(fd2,"File not found!!!!!!!!",20,0);

    close(fd1);
    close(fd2);
    printf("Sever terminated.....");
    return 0;
}
```

```
}
```

OUTPUT

Terminal	Terminal
File Edit View Search Terminal HelpServer is getting ready!.... ...waiting for client request... File name recieved is demo SERVER TERMINATEDstudent@isl01-259:~\$	File Edit View Search Terminal Help Enter the file to be sent: demoContents of file are..... hello hi namste madam sir good day

3B)Design a program to implement client server model (UDP) using socket programming

```
// server program for udp connection
#include <stdio.h>
#include <strings.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <netinet/in.h>
#define PORT 5000
#define MAXLINE 1000

int main()
{
    char buffer[100];
    char *message = "Hello Client";
    int fd, len;
    struct sockaddr_in s, c;

    bzero(&s, sizeof(s));

    // Create a UDP Socket
    fd = socket(AF_INET, SOCK_DGRAM, 0);
    s.sin_family = AF_INET;
    s.sin_port = htons(PORT);
    s.sin_addr.s_addr = htonl(INADDR_ANY);

    // bind server address to socket descriptor

    bind(fd, (struct sockaddr*)&s, sizeof(s));
    printf("Waiting for client request....\n");

    //receive the datagram
    len = sizeof(c);
    int n = recvfrom(fd, buffer, sizeof(buffer),
                    0, (struct sockaddr*)&c, &len);
    buffer[n] = '\0';
    puts(buffer);

    // send the response
```

```

        sendto(fd, message, MAXLINE, 0,
               (struct sockaddr*)&c, sizeof(c));
    }

// udp client
#include <stdio.h>
#include <strings.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <unistd.h>
#include <stdlib.h>

#define PORT 5000
#define MAXLINE 1000

int main()
{
    char buffer[100];
    char *message = "Hello Server";
    int fd1, n;
    struct sockaddr_in s;

    // clear servaddr

    bzero(&s, sizeof(s));
    s.sin_family = AF_INET;
    s.sin_port = htons(PORT);
    s.sin_addr.s_addr = inet_addr("127.0.0.1");

    // create datagram socket
    fd1 = socket(AF_INET, SOCK_DGRAM, 0);

    // connect to server
    if(connect(fd1, (struct sockaddr *)&s, sizeof(s)) < 0)
    {
        printf("\n Error : Connect Failed \n");
        exit(0);
    }
    // request to send datagram
    // no need to specify server address in sendto
    // connect stores the peers IP and port
    sendto(fd1, message, MAXLINE, 0, (struct sockaddr*)NULL,
sizeof(s));
    // waiting for response
    recvfrom(fd1, buffer, sizeof(buffer), 0, (struct sockaddr*)NULL,
NULL);

    puts(buffer);
    printf("\nMessage recieved from client\n");
}

```



```

        // close the descriptor
        close(fd1);
    }

```

OUTPUT

Terminal	Terminal
File Edit View Search Terminal Help	File Edit View Search Terminal Help
.....Server is getting ready!....	Enter the file to be sent:
...waiting for client request...	demo
File name recieved is demoContents of file are.....
SERVER TERMINATEDstudent@isl01-259:~\$	hello
	hi
	namste
	madam
	sir
	good day

Routing Algorithm

4) Design and implement a program to route the packet in a network using distance vector algorithm.

```

#include<stdio.h>
struct node
{
    int dist[20];
    int from[20];
    }rt[10];
int main()
{
    int dmat [20] [20];
    int n, i, j, k, count=1;
    printf ("\nEnter the number of nodes :\n");
    scanf ("%d", &n);
    printf ("\nEnter the cost matrix :\n");
    for (i=1; i<=n; i++)
    for (j=1; j<=n; j++)
    {
        scanf ("%d", &dmat[i][j]);
    }
}

```

```

dmat [i][i] = 0;
rt[i].dist[j] = dmat[i][j];
rt[i].from[j] = j;
}
do
{
for (i=1; i<=n; i++)
for (j=1; j<=n; j++)
for (k=1; k<=n; k++)
if (rt[i].dist[j] > dmat[i][k] + rt[k].dist[j])
{
rt[i].dist[j] = rt[i].dist[k] + rt[k].dist[j];
rt[i].from[j] = k;
}

count++;
}while (count < n);
for (i=1; i<=n; i++)
{
printf ("\nDistance Table for router %c is \n", i+64);
for (j=1; j<=n; j++)
printf ("\tNode %d Via %d, Distance : %d\n", j, rt[i].from[j], rt[i].dist[j]);

}

return 0;
}

```

OUTPUT

```

Enter the number of nodes :
4
Enter the cost matrix :
0 3 2 99
4 0 1 2
2 1 0 5
99 2 5 0

Distance Table for router A is
Node 1 Via 1, Distance : 0
Node 2 Via 2, Distance : 3
Node 3 Via 3, Distance : 2
Node 4 Via 2, Distance : 5

Distance Table for router B is
Node 1 Via 3, Distance : 3
Node 2 Via 2, Distance : 0
Node 3 Via 3, Distance : 1
Node 4 Via 4, Distance : 2

Distance Table for router C is
Node 1 Via 1, Distance : 2
Node 2 Via 2, Distance : 1
Node 3 Via 3, Distance : 0
Node 4 Via 2, Distance : 3

Distance Table for router D is
Node 1 Via 2, Distance : 5
Node 2 Via 2, Distance : 2
Node 3 Via 2, Distance : 3
Node 4 Via 4, Distance : 0

```

CONGESTION CONTROL

5)Design a program for congestion control using leakybucket algorithm.

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
int i,j,qs,ns,t,count,size,a,choice,p[10],cap,rate,delay,flag=1,t1,t2;
printf("enter the queue size:");
scanf("%d",&size);
count=size;
printf("enter leaky bucket capacity:");
scanf("%d",&cap);
qs=cap;
printf("enter the size of the packets in the queue:");
for(i=0;i<size;i++)
{
scanf("%d",&a);
if(a>cap)
{
i--;
printf("packets cannot be entered");
}
else
p[i]=a;
}
printf("enter the output rate:");
scanf("%d",&rate);
delay=cap/rate;
printf("delay=%d\n",delay);
while(flag)
```

```

{
qs=cap;
while(qs>=p[0]&&count>0)
{
printf("\npacket of size %d is put into the bucket\n",p[0]);
qs=qs-p[0];
printf("\navailable space %d\n",qs);
count--;
for(i=0;i<count;i++)
p[i]=p[i+1];
}
t=delay;
long int t1=(long)time(NULL);
long int t2=(long)time(NULL);
while((t2-t1)<delay)
{
t2=(long)time(NULL);

if((delay-t)==(t2-t1))
{
printf("\ntransmitting packets in the leaky bucket:%d seconds\n",t);
t--;
}
}
printf("\npackets in the queue:\n");
for(i=0;i<count;i++)
printf("%d\t",p[i]);
printf("\ndo u want to enter more packets in the queue?(1 or 0)\n");
scanf("%d",&choice);
while(choice&&count<size)
{
printf("enter the no of packets (<=%d)\n",size-count);
scanf("%d",&ns);
if(ns>(size-count))
printf("\nexceeding queue size\n");
else
{
printf("\nenter the size of the packets to put in the queue:\n");

```

```
for(i=0;i<ns;i++)
{
scanf("%d",&a);
if(a>cap)
printf("packets cannot be entered");
else
p[count++]=a;
}
}
printf("\ndo u want to enter more? (0 or 1)\n");
scanf("%d",&choice);
if(choice!=0)
if(count>=size)
printf("queue is full");
}
if(count==0)
flag=0;
}
}
```

OUTPUT

```

enter the queue size:5
enter leaky bucket capacity:6
enter the size of the packets in the queue:3 4 1 2 3
enter the output rate:2
delay=3

packet of size 3 is put into the bucket

available space 3

transmitting packets in the leaky bucket:-579321184 seconds
,t
transmitting packets in the leaky bucket:-610418401 seconds
,t
transmitting packets in the leaky bucket:-610418401 seconds
,t
transmitting packets in the leaky bucket:-610418401 seconds
,t
packets in the queue:
4      1      2      3
do u want to enter more packets in the queue?(1 or 0)
0

packet of size 4 is put into the bucket

available space 2

packet of size 1 is put into the bucket

available space 1

transmitting packets in the leaky bucket:-579321184 seconds
,t
transmitting packets in the leaky bucket:-610418401 seconds
,t
transmitting packets in the leaky bucket:-610418401 seconds
,t
transmitting packets in the leaky bucket:-610418401 seconds
,t
packets in the queue:
2      3
do u want to enter more packets in the queue?(1 or 0)

```