SREE DATTHA
INSTITUTE OF ENGINEERING AND SCIENCE

COMPUTER NETWORKS
Lab Manual
Department of Computer Science and Engineering
III year B. Tech Semister-1

Course Objectives:

1. To understand the working principle of various communication protocols.
2. To understand the network simulator environment and visualize a network topology and observe its performance.
3. To analyze the traffic flow and the contents of protocol frames

Course Outcomes:

1. Implement data link layer farming methods
2. Analyze error detection and error correction codes.
3. Implement and analyze routing and congestion issues in network design.
4. Implement Encoding and Decoding techniques used in presentation layer.
5. To be able to work with different network tools.

List of experiments Computer Networks

1. Implement Character Stuffing and Bit Stuffing on Given Data.
2. Implement CRC Techniques on Given Data.
3. Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-Nmechanism.
4. Implement Dijkstra's Algorithm to Compute the Shortest Path through a Graph.
5. Take an example subnet of hosts and Obtain broad cast tree for it.
6. Implement distance vector routing algorithm for obtaining routing tables at each node.
7. Take a 64-bit plain text and encrypt the same using DES algorithm.
8. Write a program for congestion control using Leaky bucket algorithm.
9. Write a program for frame sorting technique used in buffers.

Text Books

1. WEB TECHNOLOGIES: A Computer Science Perspective, Jeffrey C. Jackson, Pearson Education.

References

1. Deitel H.M. and Deitel P.J., "Internet and World Wide Web How to program", Pearson International, 2012, 4th Edition.
2. J2EE: The complete Reference By James Keogh, McGraw-Hill
3. Bai and Ekedhi, The Web Warrior Guide to Web Programming, Thomson
4. Paul Dietel and Harvey Deitel," Java How to Program", Prentice Hall of India, 8th Edition
5. Web technologies, Black Book, Dreamtech press.
6. Gopalan N.P. and Akilandeswari J., "Web Technology", Prentice Hall of India

Index page

COMPUTER NETWORKS EXPERIMENTS

Experiment – 1

Aim: To Implement Character Stuffing and Bit Stuffing on Given Data

a) Implement Character Stuffing on Given Data

Program:

```
#include<stdio.h>
#include<string.h>
main()
{
   inti,j,k,l,count=0,n;
   char  s[100],cs[50];
   clrscr();
   printf("\n ENTER THE BIT STRING:");
   gets(s);
   n=strlen(s);
   printf("\nTHE STRING IS\n");
   for(i=0;i<n;)
    {
        if(s[i]==s[i+1])
        {
            count=2;
            i++;
            while(s[i]==s[i+1])
            {
                i++;
                count++;
            }
            if(count>=5)
            {
                printf("$");
                if(count<10)
                printf("0");
                printf("%d%c",count,s[i]);
                i++;
            }
            els
            e
            {       for(j=0;j<count;j++)
                printf("%c",s[i]);
                i++;
```

```
                        }
                                                        C
                }
else
                {
                        printf("%c",s[i]);
                        i++;
                }
        }
        getch();
}
```

INPUT/OUTPUT:

ENTER THE BIT STRING:

123AAAAAAAAATYKKKPPPPP

THE STRING IS

123$10ATYKKK$05P

b) Implement Bit Stuffing on Given Data

Program:

```
#include<stdio.h>
#include<string.h>
main()
{
   char a[20],fs[50]="",t[6],r[5];
   inti,j,p=0,q=0;
   clrscr();
   printf("enter bit string : ");
   scanf("%s",a);
   strcat(fs,"01111110");
   if(strlen(a)<5)
    {
          strcat(fs,a);
    }
   els
   e
    {     for(i=0;i<strlen(a)-4;i++)
           {
```

```
    for(j=i;j<i+5;j++)
     {
            t[p++]=a[j];
     }
    t[p]='\0';
    if(strcmp(t,"11111")==0)
          {
            strcat(fs,"111110");
            i=j-1;
     }
    else
    {
            r[0]=a[i];
            r[1]='\0';
            strcat(fs,r);
     }
     p=0;
}
```

```
                for(q=i;q<strlen(a);q++)
            {
                    t[p++]=a[q];
            }
            t[p]='\0';
            strcat(fs,t);
    }
    strcat(fs,"01111110");
    printf("After stuffing : %s",fs);
    getch();
}
```

BIT STUFFING OUTPUT

Enter bit string : 10101111110

After stuffing : 01111110101011111101001111110

Enter bit string : 10111110111110111110

After stuffing : 01111110101111100111101111100011111110

Experiment – 2

Aim: To Implement CRC Techniques on Given Data

Program

```
#include<stdio.h>
const char * bindiv(const char *,const char *);
const char * binsub(const char *,const char *);
int f=0,ll=0;
main()
 {
    char *a,p[13]="1100000001011",g[30],g1[30],yy[30]="",td[30],*aa;
   int l=0,i;
   clrscr();
   printf("enter transfered data : ");
   scanf("%s",g);
   printf("enter received data : ");
   scanf("%s",td);
   strcpy(g1,g);
   strcat(g,"000000000000");
   printf("\n%s  %s ",p,g);
   a=bindiv(g,p);
   if(strlen(a)<12)
    {
          for(i=strlen(a);i<12;i++)
          {
                yy[l++]='0';
           }
          yy[l]='\0';
   }
   strcat(yy,a);
   strcat(g1,yy);
   printf("\ncrc is %s",yy);
   printf("\n_____");
   strcat(td,yy);
   printf("\n\n%s ) %s (",p,td);
   ll=0;
   aa=bindiv(td,p);
   strcpy(a,aa);
   printf("\n %s",a);
   printf("\n_____");
```

```
   if(f==1)



   printf("\ndatatransfered correctly");
   else
   printf("\ndatatransfered incorrectly");
   getch();
 }
const char * bindiv(const char *s,const char *d)
 {
   inti,j,k=0,x=13,h,p=0,l;
   char q[15]="",b[30],*w;
   for(i=0;i<strlen(s);i++)
   {
            if((i+x)>strlen(s))
            x=(i+x)-strlen(s)+1;
            for(j=i;j<(i+x);j++)
             {
                     b[k++]=s[j];
             }

            b[k]='\0';
            if(ll!=0)
            printf("\n %s",b);
            ll=1;
            if(strlen(b)==12)
            {
                     break;
             }
            printf("\n %s",d);
            printf("\n_____");
             w=binsub(b,d);
             k=0;i=j-1;
            for(l=0;l<strlen(w);l++)
             {
                     if(w[l]=='1')
                     break;
             }
            if(l==strlen(w))
             {
                      f=1;
                     return(w);

             }
```

```
            for(h=l;h<strlen(w);h++)
             {
                    q[p++]=w[h];
                  }

             q[p]='\0';
             x=13-strlen(q);
            strcpy(b,"");
            strcat(b,q);
            k=strlen(q); p=0;
   }
   return(b);
 }
const char * binsub(const char *x,const char *y)
 {
   inti,j=0;
    char w[15]="",e[3],f[3],n[3];
   e[0]='1';
   e[1]='\0';
   f[0]='0';
   f[1]='\0';
   for(i=0;i<strlen(x);i++)
    {
            if((x[i]=='1')&&(y[i]=='1'))
            strcat(w,f);
            else
            if((x[i]=='0')&&(y[i]=='0'))
            strcat(w,f);
            else
            strcat(w,e);
    }
   n[0]='\0';
   n[1]='\0';
   strcat(w,n);
   return(w);
}
```

CRC-12 OUTPUT:

```
Enter transferred data: 10101
Enter received data : 10101
1100000001011 ) 10101000000000000 (
 1100000001011
 ----------------------
 1101000010110
 1100000001011
 ----------------------
 1000011101000
 1100000001011
 ----------------------
crc is 100011100011
 ----------------------
1100000001011 ) 10101100011100011 (
 1100000001011
 ----------------------
 1101100001010
 1100000001011
 ----------------------
 1100000001011
 1100000001011
---------------------- 0000000000000
Data transfered correct
```

Experiment – 3

Aim: To Implement CRC - 16 on Given Data

Program

```
#include<stdio.h>
const char * bindiv(const char *,const char *);
const char * binsub(const char *,const char *);
int f=0,ll=0;
main()
 {
   char *a,p[20]="10001000000100001", g[30],g1[30],yy[30]="",td[30],*aa;
   int l=0,i;
   clrscr();
   printf("enter transfered data : ");
   scanf("%s",g);
   printf("enter received data : ");
   scanf("%s",td);
   strcpy(g1,g);
   strcat(g,"0000000000000000");
   printf("\n%s %s ",p,g);
   a=bindiv(g,p);
   if(strlen(a)<16)
    {
          for(i=strlen(a);i<16;i++)
          {
                yy[l++]='0';
           }
          yy[l]='\0';
    }
       strcat(yy,a);
   strcat(g1,yy);
   printf("\n_____");
   printf("\ncrc is %s",yy);
   strcat(td,yy);
   printf("\n\n%s ) %s (",p,td);
   ll=0;
   aa=bindiv(td,p);
   strcpy(a,aa);
   printf("\n %s",a);
   printf("\n_____");
```

```
   if(f==1)
   printf("\ndatatransfered correctly");
   else
   printf("\ndatatransfered incorrectly");
   getch();
 }
const char * bindiv(const char *s,const char *d)
 {
   inti,j,k=0,x=17,h,p=0,l;
   char q[25]="",b[30],*w;
   for(i=0;i<strlen(s);i++)
   {
          if((i+x)>strlen(s))
          x=(i+x)-strlen(s)+1;
          for(j=i;j<(i+x);j++)
           {
                 b[k++]=s[j];
           }
          b[k]='\0';
          if(ll!=0)
          printf("\n %s",b);
          ll=1;
          if(strlen(b)==16)
           {
                 break;
           }
          printf("\n %s",d);
          printf("\n_____");
          w=binsub(b,d);
          k=0;i=j-1;
          for(l=0;l<strlen(w);l++)
           {
                 if(w[l]=='1')
                 break;
           }
          if(l==strlen(w))
           {
                 f=1;
                 return(w);
           }

          for(h=l;h<strlen(w);h++)
```

```
            {
                q[p++]=w[h];
            }
            q[p]='\0';
            x=17-strlen(q);
           strcpy(b,"");
           strcat(b,q);
           k=strlen(q); p=0;
    }
   return(b);
 }
const char * binsub(const char *x,const char *y)
 {
   inti,j=0;
    char w[25]="",e[3],f[3],n[3];
   e[0]='1';
   e[1]='\0';
   f[0]='0';
   f[1]='\0';
   for(i=0;i<strlen(x);i++)
    {
           if((x[i]=='1')&&(y[i]=='1'))
           strcat(w,f);
           else
           if((x[i]=='0')&&(y[i]=='0'))
           strcat(w,f);
           else
           strcat(w,e);
    }
   n[0]='\0';
   n[1]='\0';
   strcat(w,n);
   return(w);
 }
```

CRC-16 OUTPUT:

```
Enter transferred data : 11011
Enter received data : 11011
10001000000100001 ) 11011000000000000000 (
 10001000000100001
 ---------------------------
 10100000001000010
 10001000000100001
 ---------------------------
 10100000110001100
 10001000000100001
 ---------------------------
 1010001101011010
 ---------------------------
crc is 1010001101011010
10001000000100001 ) 11011101000110101 1010 (
 10001000000100001
 ---------------------------
 10101010000101001
 10001000000100001
 ---------------------------
 10001000000100001
 10001000000100001
 ---------------------------
 00000000000000000
Data transferred correctly
```

Experiment – 4

Aim:
ToDevelopasimpledatalinklayerthatperformstheflowcontrolusingtheslidingwindowprotocol , and loss recovery using the Go-Back-Nmechanism.

Program

```
#include<stdio.h>

int main()
{
   intw,i,f,frames[50];

   printf("Enter window size: ");
   scanf("%d",&w);

   printf("\nEnter number of frames to transmit: ");
   scanf("%d",&f);

   printf("\nEnter %d frames: ",f);

   for(i=1;i<=f;i++)
      scanf("%d",&frames[i]);

   printf("\nWith sliding window protocol the frames will be sent in the
following manner (assuming no corruption of frames)\n\n");
   printf("After sending %d frames at each stage sender waits for
acknowledgement sent by the receiver\n\n",w);

   for(i=1;i<=f;i++)
   {
      if(i%w==0)
      {
         printf("%d\n",frames[i]);
         printf("Acknowledgement of above frames sent is received by
sender\n\n");
      }
      else
         printf("%d ",frames[i]);
   }
```

```
        if(f%w!=0)
            printf("\nAcknowledgement of above frames sent is received by
sender\n");

        return 0;
    }
```

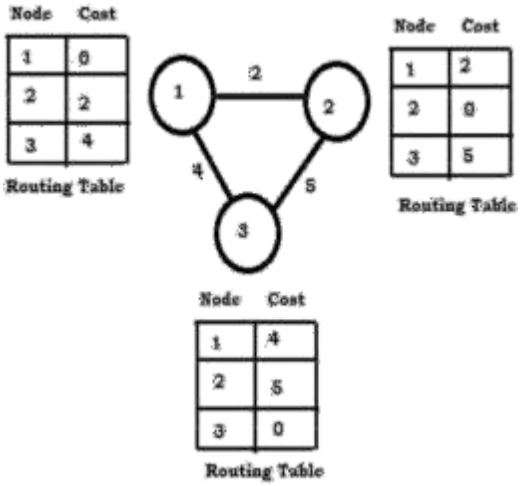
Output

Enter window size: 3
Enter number of frames to transmit: 5
Enter 5 frames: 12 5 89 4 6

Experiment – 5

Aim: To Implement Dijkstra'sAlgorithm to Compute the Shortest Path through a Graph

Program




```
#include<stdio.h>
struct node
{
   unsigneddist[20];
   unsigned from[20];
}
rt[10];
int main()
{
   intdmat[20][20];
   intn,i,j,k,count=0;
   printf("\nEnter the number of nodes : ");
   scanf("%d",&n);
   printf("Enter the cost matrix :\n");
   for(i=0;i<n;i++)
   for(j=0;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
    {
            count=0;
            for(i=0;i<n;i++)
            for(j=0;j<n;j++)
            for(k=0;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!=0);
    for(i=0;i<n;i++)
    {
            printf("\nState value for router %d is \n",i+1); for(j=0;j<n;j++)
            {
                    printf("\nnode %d via %d
            Distance%d",j+1,rt[i].from[j]+1,rt[i].dist[j]);
            }
    }
    printf("\n");
}
```

OUTPUT:

```
Enter the number of nodes : 2
Enter the cost matrix :
1 2
1 2
State value for router 1 is
node 1 via 1 Distance0
node 2 via 2 Distance2
State value for router 2 is
node 1 via 1 Distance1
node 2 via 2 Distance0
```

Experiment – 6

Aim: To Take an example subnet of hosts and Obtain broad cast tree for it

Program

```
#include<stdio.h>
intp,q,u,v,n;
int min=99,mincost=0;
int t[50][2],i,j;
int parent[50],edge[50][50];
main()
{
   clrscr();
   printf("\n Enter the number of nodes");
   scanf("%d",&n);
   for(i=0;iedge[i][j])
   {
          min=edge[i][j];
          u=i;
           v=j;
   }
   p=find(u);
   q=find(v);
   if(p!=q)
   {
          t[i][0]=u;
          t[i][1]=v;
          mincost=mincost+edge[u][v];
          sunion(p,q);
    }
   els
   e
    {     t[i][0]=-1;
          t[i][1]=-1;

   }
   min=99;
   }
   printf("Minimum cost is %d\n Minimum spanning tree is\n" ,mincost);
```

```
    for(i=0;i0) l=parent[l];
    return l;
}
```

OUTPUT:

```
Enter the number of nodes3
A BC
A1 2 3 4

B1 2 3 4

C4 5 6 7

Minimum cost is 3

Minimum spanning tree is

C A 3
```

Experiment – 7

Aim: To implement distance vector routing algorithm for obtaining routing tables at eachnode.

Program

```
#include<stdio.h>
#include<math.h>
#include<conio.h>
main()
{
   inti,j,k,nv,sn,noadj,edel[20],tdel[20][20],min;
   charsv,adver[20],ch;
   clrscr();
   printf("\n ENTER THE NO.OF VERTECES:");
   scanf("%d",&nv);
   printf("\n ENTER THE SOURCE VERTEX NUM,BER AND NAME:");
   scanf("%d",&sn);
   flushall();
   sv=getchar();
   printf("\n NETER NO.OF ADJ VERTECES TO VERTEX %c",sv);
   scanf("%d",&noadj);
   for(i=0;i<noadj;i++)
   {
         printf("\n ENTER TIME DELAY and NODE NAME:");
         scanf("%d %c",&edel[i],&adver[i]);
   }
   for(i=0;i<noadj;i++)
   {
         printf("\n ENTER THE TIME DELAY FROM %c to ALL OTHER
         NODES: ",adver[i]);
         for(j=0;j<nv;j++)
         scanf("%d",&tdel[i][j]);
   }
   printf("\n DELAY VIA--VERTEX \n ");
   for(i=0;i<nv;i++)
   {
         min=1000;
         ch=0;
         for(j=0;j<noadj;j++)
```

```
            if(min>(tdel[j][i]+edel[j]))
             {
                    min=tdel[j][i]+edel[j];
                    ch=adver[j];
             }
            if(i!=sn-1)
            printf("\n%d %c",min,ch);
            else
            printf("\n0 -");
    }
    getch();
}
```

INPUT/OUTPUT:

ENTER THE NO.OF VERTECES:12

ENTER THE SOURCE VERTEX NUMBER AND NAME:10 J

ENTER NO.OF ADJ VERTECES TO VERTEX 4

ENTER TIME DELAY and NODE NAME:8 A

ENTER TIME DELAY and NODE NAME:10 I

ENTER TIME DELAY and NODE NAME:12 H

ENTER TIME DELAY and NODE NAME:6 K

ENTER THE TIME DELAY FROM A to ALL OTHER NODES:
0 12 25 40 14 23 18 17 21 9 24 29

ENTER THE TIME DELAY FROM I to ALL OTHER NODES:
24 36 18 27 7 20 31 20 0 11 22 33

ENTER THE TIME DELAY FROM H to ALL OTHER NODES:
20 31 19 8 30 19 6 0 14 7 22 9

ENTER THE TIME DELAY FROM K to ALL OTHER NODES:
21 28 36 24 22 40 31 19 22 10 0 9

DELAY VIA--VERTEX
8 a
20 a
28 i
20 h
17 i
30 i
18 h
12 h
10 i
0 -
6 k
15 K

Experiment – 8

Aim: Take a 64-bit plain text and encrypt the same using DES algorithm.

Program

```
importjava.util.*;
importjava.io.BufferedReader;
importjava.io.InputStreamReader;
importjava.security.spec.KeySpec;
importjavax.crypto.Cipher;
importjavax.crypto.SecretKey;
importjavax.crypto.SecretKeyFactory;
importjavax.crypto.spec.DESedeKeySpec;
import sun.misc.BASE64Decoder;
import sun.misc.BASE64Encoder;
public class DES {
private static final String UNICODE_FORMAT = "UTF8";
public static final String DESEDE_ENCRYPTION_SCHEME = "DESede";
privateKeySpecmyKeySpec;
privateSecretKeyFactorymySecretKeyFactory;
private Cipher cipher;
byte[] keyAsBytes;
private String myEncryptionKey;
private String myEncryptionScheme;
SecretKey key;
```

```
staticBufferedReaderbr = new BufferedReader(new InputStreamReader(System.in));
public DES() throws Exception {
//        TODO code application logic here myEncryptionKey =
"ThisIsSecretEncryptionKey"; myEncryptionScheme =
DESEDE_ENCRYPTION_SCHEME; keyAsBytes =
myEncryptionKey.getBytes(UNICODE_FORMAT); myKeySpec = new
DESedeKeySpec(keyAsBytes);

mySecretKeyFactory = SecretKeyFactory.getInstance(myEncryptionScheme); cipher =
Cipher.getInstance(myEncryptionScheme);




key = mySecretKeyFactory.generateSecret(myKeySpec);

}

public String encrypt(String unencryptedString) { String encryptedString = null;
try {

cipher.init(Cipher.ENCRYPT_MODE, key);

byte[] plainText = unencryptedString.getBytes(UNICODE_FORMAT); byte[]
encryptedText = cipher.doFinal(plainText);

BASE64Encoder base64encoder = new BASE64Encoder();

encryptedString = base64encoder.encode(encryptedText); }

catch (Exception e) {

e.printStackTrace(); }

returnencryptedString; }

public String decrypt(String encryptedString) { String decryptedText=null;

try {

cipher.init(Cipher.DECRYPT_MODE, key);
```

```
BASE64Decoder base64decoder = new BASE64Decoder(); byte[] encryptedText =
base64decoder.decodeBuffer(encryptedString); byte[] plainText =
cipher.doFinal(encryptedText); decryptedText= bytes2String(plainText); }

catch (Exception e) {

e.printStackTrace(); }

returndecryptedText; }

private static String bytes2String(byte[] bytes) { StringBufferstringBuffer = new
StringBuffer(); for (inti = 0; i<bytes.length; i++) { stringBuffer.append((char) bytes[i]); }
returnstringBuffer.toString(); }

public static void main(String args []) throws Exception { System.out.print("Enter the
string: ");
DES myEncryptor= new DES();

String stringToEncrypt = br.readLine();

String encrypted = myEncryptor.encrypt(stringToEncrypt);




String decrypted = myEncryptor.decrypt(encrypted); System.out.println("\nString To
Encrypt: " +stringToEncrypt); System.out.println("\nEncrypted Value : " +encrypted);
System.out.println("\nDecrypted Value : " +decrypted); System.out.println("");

}

}
```

OUTPUT:

```
Enter the string: Welcome

String To Encrypt: Welcome

Encrypted Value : BPQMwc0wKvg=

Decrypted Value : Welcome
```

Experiment – 9

Aim: To Write a program for congestion control using Leaky bucketalgorithm.

Program

```
importjava.io.*;
importjava.util.*;

classLeakybucket {
   publicstaticvoidmain (String[] args) {
      intno_of_queries,storage,output_pkt_size;
      intinput_pkt_size,bucket_size,size_left;

       //initial packets in the bucket
      storage=0;

      //total no. of times bucket content is checked
      no_of_queries=4;

      //total no. of packets that can
      // be accomodated in the bucket
      bucket_size=10;

      //no. of packets that enters the bucket at a time
      input_pkt_size=4;
        //no. of packets that exits the bucket at a time
      output_pkt_size=1;
      for(inti=0;i<no_of_queries;i++)
      {
         size_left=bucket_size-storage; //space left
          if(input_pkt_size<=(size_left))
         {
            storage+=input_pkt_size;
            System.out.println("Buffer size= "+storage+
               " out of bucket size= "+bucket_size);
         }
         else
         {
            System.out.println("Packet loss = "
                  +(input_pkt_size-(size_left)));

               //full size
            storage=bucket_size;

            System.out.println("Buffer size= "+storage+
```

```
                    " out of bucket size= "+bucket_size);

            }
            storage-=output_pkt_size;
        }
    }
}
```

Output

```
Buffer size= 4 out of bucket size= 10
Buffer size= 7 out of bucket size= 10
Buffer size= 10 out of bucket size= 10
Packet loss = 3
Buffer size= 10 out of bucket size= 10
```

Experiment 10

Aim: To Write a program for frame sorting technique used in buffers.

Program

```
#include<stdio.h>
#include<string.h>
#define FRAM_TXT_SIZ 3
#define MAX_NOF_FRAM 127
char str[FRAM_TXT_SIZ*MAX_NOF_FRAM];
struct frame // structure maintained to hold frames
{ char text[FRAM_TXT_SIZ];
int seq_no;
}fr[MAX_NOF_FRAM], shuf_ary[MAX_NOF_FRAM];
int assign_seq_no() //function which splits message
{ int k=0,i,j; //into frames and assigns sequence no
for(i=0; i < strlen(str); k++)
{ fr[k].seq_no = k;
for(j=0; j < FRAM_TXT_SIZ && str[i]!='\0'; j++)
fr[k].text[j] = str[i++];
}
printf("\nAfter assigning sequence numbers:\n");
for(i=0; i < k; i++)
printf("%d:%s ",i,fr[i].text);
return k; //k gives no of frames
}
void generate(int *random_ary, const int limit) //generate array of random nos
{ int r, i=0, j;
while(i < limit)
{ r = random() % limit;
for(j=0; j < i; j++)
if( random_ary[j] == r )
break;
if( i==j ) random_ary[i++] = r;
} }
void shuffle( const int no_frames ) // function shuffles the frames
{
int i, k=0, random_ary[no_frames];
generate(random_ary, no_frames);
for(i=0; i < no_frames; i++)
shuf_ary[i] = fr[random_ary[i]];
printf("\n\nAFTER SHUFFLING:\n");
for(i=0; i < no_frames; i++)
printf("%d:%s ",shuf_ary[i].seq_no,shuf_ary[i].text);
}
```

```
void sort(const int no_frames) // sorts the frames
{
int i,j,flag=1;
struct frame hold;
for(i=0; i < no_frames-1 && flag==1; i++) // search for frames in sequence
{
flag=0;
for(j=0; j < no_frames-1-i; j++) //(based on seq no.) and display
if(shuf_ary[j].seq_no > shuf_ary[j+1].seq_no)
{
hold = shuf_ary[j];
shuf_ary[j] = shuf_ary[j+1];
shuf_ary[j+1] = hold;
flag=1;
}
}
}
int main()
{
int no_frames,i;
printf("Enter the message: ");
gets(str);
no_frames = assign_seq_no();
shuffle(no_frames);
sort(no_frames);
printf("\n\nAFTER SORTING\n");
for(i=0;i<no_frames;i++)
printf("%s",shuf_ary[i].text);
printf("\n\n");
}
```

OUTPUT

```
[root@localhostnwcn]# ./a.out
Enter the message: Welcome To Acharya Institute of Technology
After assigning sequence numbers:
0:Wel 1:com 2:e T 3:o A 4:cha 5:rya 6: In 7:sti 8:tut 9:e o 10:f T 11:ech 12:nol 13:ogy
AFTER SHUFFLING:
1:com 4:cha 9:e o 5:rya 3:o A 10:f T 2:e T 6: In 11:ech 13:ogy 0:Wel 8:tut 12:nol 7:sti
AFTER SORTING
Welcome To Acharya Institute of Technology
```