



Exp No:  
Date:

Page No:

## 2.1) Write a Program to implement the data link layer framing method Character Stuffing.

**Aim:** Program to implement Character Stuffing.

**Program:**

```
import java.util.*;
public class CharacterStuffing
{
    public static void main(String[] args)
    {
        Scanner k =new Scanner (System.in);
        System.out.println("enter the string:\t");
        String s=k.nextLine();
        String str1;
        String str2="";
        int i,m,j;
        m=s.length();
        System.out.println("original data:"+s);
        str1="dlestd";
        for(i=0;i<=m-1;i++)
        {
            if((s.charAt(i)=='d')&&(s.charAt(i+1)=='l')&&(s.charAt(i+2)=='e'))
            {
                str1=str1+"dle";
            }
            str1=str1+s.substring(i,i+1);
        }
        str1=str1+"dleetx";
        int p=str1.length();
        System.out.println("transmitted data:"+str1);
        for(i=6;i<p-6;i++)
        {
            if((str1.charAt(i)=='d')&&(str1.charAt(i+1)=='l')&&(str1.charAt(i+2)=='e')&&(str1.charAt(i+3)=='d')&&(str1.charAt(i+4)=='l')&&(str1.charAt(i+5)=='e'))
            {
                i=i+3;
            }
            str2=str2+str1.substring(i,i+1);
        }
        System.out.println("received data is:"+str2);
    }
}
```

**Output:**

```
enter the string:  
Aditya dle c  
original data:Aditya dle c  
transmitted data:dlestdAditya dledle cdlestd  
received data is:Aditya dle c
```



## 2.2) Write a Program to implement the data link layer framing method Bit Stuffing.

**Aim:** Program to implement Bit Stuffing.

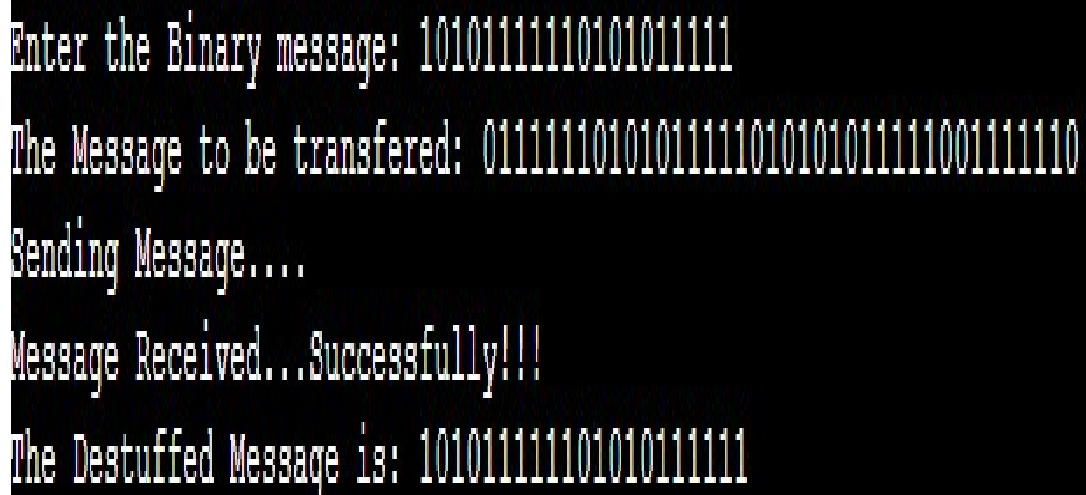
**Program:**

```
import java.util.*;
public class BitStuffing
{
    public static void main(String[] args)
    {
        System.out.print("Enter the Binary message: ");
        Scanner sn=new Scanner(System.in);
        String data = sn.nextLine();
        String res = new String();
        String out=new String();
        int counter = 0;
        for(int i=0;i<data.length();i++)
        { if (data.charAt(i)!='1' && data.charAt(i)!='0')
            {
                System.out.println("Enter only Binary values!!!");
                return;
            }
            if(data.charAt(i) == '1')
            {
                counter++;
                res = res + data.charAt(i);
            }
            else
            {
                res = res + data.charAt(i);
                counter = 0;
            }
            if(counter == 5)
            {
                res = res + '0';
                counter = 0;
            }
        }
        String inc=res;
        System.out.println("The Message to be transfered: " +inc);
        System.out.println("Sending Message ... ");
        counter=0;
        for(int i=0;i<res.length();i++)
        {

            if(res.charAt(i) == '1')
```

```
{  
  
    counter++;  
    out = out + res.charAt(i);  
  
}  
else  
{  
    out = out + res.charAt(i);  
    counter = 0;  
}  
if(counter == 5)  
{  
    if((i+2)!=res.length())  
        out = out + res.charAt(i+2);  
    else  
        out=out + '1';  
    i=i+2;  
    counter = 1;  
}  
}  
  
System.out.println("Message Received...Successfully!!!");  
System.out.println("The Destuffed Message is: "+out);  
}  
}
```

**Output:**



```
Enter the Binary message: 10101111110101011111  
The Message to be transfered: 011111110101011111010101011111001111110  
Sending Message....  
Message Received...Successfully!!!  
The Destuffed Message is: 10101111110101011111
```

**3.) Write a Program to implement data link layer framing method checksum.**

**Aim:** Program to implement checksum method.

**Program:**

```
import java.util.*;
public class CheckSumMethod
{
    public static void main(String args[])
    {
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter the string input:");
        String input = scan.next();
        int checksum = generateChecksum(input);
        System.out.println("The checksum generated is = "
            + Integer.toHexString(checksum));
        System.out.println("Enter the data to be sent:");
        input = scan.next();
        System.out.println("Enter the checksum to be sent:");
        checksum = Integer.parseInt((scan.next()), 16);
        receive(input, checksum);
        scan.close();
    }

    static int generateChecksum(String s)
    {
        String hex_value = new String();
        int x, i, checksum = 0;
        for (i = 0; i < s.length() - 2; i = i + 2)
        {
            x = (int) (s.charAt(i));
            hex_value = Integer.toHexString(x);
            x = (int) (s.charAt(i + 1));
            hex_value = hex_value + Integer.toHexString(x);
            System.out.println(s.charAt(i) + "" + s.charAt(i + 1) + " : "
                + hex_value);
            x = Integer.parseInt(hex_value, 16);
            checksum += x;
        }
        if (s.length() % 2 == 0)
        {
            x = (int) (s.charAt(i));
            hex_value = Integer.toHexString(x);
            x = (int) (s.charAt(i + 1));
            hex_value = hex_value + Integer.toHexString(x);
            System.out.println(s.charAt(i) + "" + s.charAt(i + 1) + " : "
```



```
        + hex_value);
        x = Integer.parseInt(hex_value, 16);
    }
    else
    {
        x = (int) (s.charAt(i));
        hex_value = "00" + Integer.toHexString(x);
        x = Integer.parseInt(hex_value, 16);
        System.out.println(s.charAt(i) + " : " + hex_value);
    }
    checksum += x;
    hex_value = Integer.toHexString(checksum);
    if (hex_value.length() > 4)
    {
        int carry = Integer.parseInt(("" + hex_value.charAt(0)), 16);
        hex_value = hex_value.substring(1, 5);
        checksum = Integer.parseInt(hex_value, 16);
        checksum += carry;
    }
    checksum = generateComplement(checksum);
    return checksum;
}

static void receive(String s, int checksum)
{
    int generated_checksum = generateChecksum(s);
    generated_checksum = generateComplement(generated_checksum);
    int syndrome = generated_checksum + checksum;
    syndrome = generateComplement(syndrome);
    System.out.println("Syndrome = " + Integer.toHexString(syndrome));
    if (syndrome == 0)
    {
        System.out.println("Data is received without error.");
    }
    else
    {
        System.out.println("There is an error in the received data.");
    }
}

static int generateComplement(int checksum)
{
    checksum = Integer.parseInt("FFFF", 16) - checksum;
    return checksum;
}
}
```

### Output 1:

```
Enter the string input:
Aditya
Ad : 4164
it : 6974
ya : 7961
The checksum generated is = dbc5
Enter the data to be sent:
Aditya
Enter the checksum to be sent:
dbc5
Ad : 4164
it : 6974
ya : 7961
Syndrome = 0
Data is received without error.
```

### Output 2:

```
Enter the string input:
Aditya
Ad : 4164
it : 6974
ya : 7961
The checksum generated is = dbc5
Enter the data to be sent:
Aditya
Enter the checksum to be sent:
bdc5
Ad : 4164
it : 6974
ya : 7961
Syndrome = 1e00
There is an error in the received data.
```



#### 4) Write a program for Hamming Code generation for error detection and correction.

**Program:**

```
#include<stdio.h>
void main()
{
    int data[10];
    int dataatrec[10],c,c1,c2,c3,i;
    printf("Enter 4 bits of data one by one\n");
    scanf("%d",&data[0]);
    scanf("%d",&data[1]);
    scanf("%d",&data[2]);
    scanf("%d",&data[4]);
    //calculation of even parity
    data[6]=data[0]^data[2]^data[4];
    data[5]=data[0]^data[1]^data[4];
    data[3]=data[0]^data[1]^data[2];
    printf("The encoded data is \n");
    for(i=0;i<7;i++)
    printf("%d",data[i]);
    printf("\n\nEnter received data bits one by one\n");
    for(i=0;i<7;i++)
    scanf("%d",&dataatrec[i]);
    c1=dataatrec[6]^dataatrec[4]^dataatrec[2]^dataatrec[0];
    c2=dataatrec[5]^dataatrec[4]^dataatrec[1]^dataatrec[0];
    c3=dataatrec[3]^dataatrec[2]^dataatrec[1]^dataatrec[0];
    c=c3*4+c2*2+c1;
    if(c==0)
    {
        printf("\nNo error while transmission of data\n");
    }
    else
    {
        printf("\nError on position %d",c);
        printf("\nData sent: ");
        for(i=0;i<7;i++)
        printf("%d",dataatrec[i]);
        printf("\nCorrect message is \n");
    }
}
```





```
//if erroneous bit is 0 we complement it else vice versa
if(dataatrec[7-c]==0)
    dataatrec[7-c]=1;
else
    dataatrec[7-c]=0;
for(i=0;i<7;i++)
{
    printf("%d",dataatrec[i]);
}
}
```

**Output:**

```
C:\Users\admin\Desktop\hammingcode.exe
Enter 4 bits of data one by one
1
0
1
0
The encoded data is
1010010

Enter received data bits one by one
1
0
1
0
1
1
0

Error on position 3
Data sent: 1010110
Correct message is
1010010
-----
Process exited after 19.51 seconds with return value 1
Press any key to continue . . .
```



**5) Write a Program to implement on a data set of characters the three CRC polynomials - CRC 12, CRC 16 and CRC CCIP.**

**Program:**

```
#include<stdio.h>
#include<string.h>
#define N strlen(g)
char t[50],cs[50],g[50];
int a,e,c,b;
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;
do
{
printf("\n1.crc1\n2.crc16\n3.crc ccit\n4.exit\n\nEnter your opinion");
scanf("%d",&b);
switch(b)
{
case 1:strcpy(g,"1100000001111");
break;
case 2:strcpy(g,"11000000000000101");
break;
case 3:strcpy(g,"10001000000100001");
break;
case 4:return 0;
```



```
}
printf("\nEnter data :");
scanf("%s",t);
printf("\n-----\n");
printf("\n Generating polynomial :%s",g);
a=strlen(t);
for(e=a;e<a+N-1;e++)
t[e]='0';
printf("\n-----\n");
printf("Modified data is :%s",t);
printf("\n-----\n");
crc();
printf("checksum is :%s",cs);
for(e=a;e<a+N-1;e++)
t[e]=cs[e-a];
printf("\n-----\n");
printf("\n Final codeword is :%s",t);
printf("\n-----\n");
printf("\ntest error detection 0(yes) 1(no)?:" );
scanf("%d",&e);
if(e==0)
{
do
{
printf("\n\t enter the position where error is to be inserted:");
scanf("%d",&e);
}
while(e==0||e>a+N-1);
t[e-1]=(t[e-1]=='0')?'1':'0';
printf("\n-----\n");
printf("\n\terroneous data :%s\n",t);
}
crc();
for(e=0;(e<N-1)&&(cs[e]!='1');e++);
if(e<N-1)
printf("Error detected\n\n");
else
printf("No error detected \n\n");
printf("\n-----\n");
}while(flag!=1);
}
```

**Output:**

```
C:\Users\admin\Desktop\lab.exe

1.crc1
2.crc16
3.crc ccit
4.exit

Enter your opinion1
Enter data :10101010101010
-----
Generating polynomial :1100000001111
-----
Modified data is :101010101010101000000000000000
-----
checksum is :101010000001
-----
Final codeword is :1010101010101010101010000001
-----
test error detection 0(yes) 1(no)?:0
    enter the position where error is to be inserted:7
-----
    erroneous data :1010100010101010101010000001
Error detected
-----

1.crc1
2.crc16
3.crc ccit
4.exit

Enter your opinion
```



## 6) Write a Program to implement Sliding window protocol for Goback N.

### Program:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<unistd.h>
int n,r;
struct frame
{
    char ack;
    int data;
}frm[10];
int sender(void);
void recvack(void);
void resend_gb(void);
int main()
{
    int c;
    sender();
    recvack();
    resend_gb();
    printf("\n All Frames sent successfully\n");
}
int sender()
{
    int i;
    printf("\n Enter no.of Frames to be sent: ");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        printf("\n Enter data for Frames[%d]",i);
        scanf("%d",&frm[i].data);
        frm[i].ack='y';
    }
    return 0;
}
void recvack()
{
    int i;
```



```
rand();
r=rand()%n;
frm[r].ack='n';
for(i=1;i<=n;i++)
{
    if(frm[i].ack=='n')
        printf("\n The Frame Number %d is not received",r);
}
}
void resend_gb()
{
    int i;
    printf("\nResending Frame %d",r);
    for(i=r;i<=n;i++)
    {
        sleep(2);
        frm[i].ack='y';
        printf("\n The Received Frame is %d",frm[i].data);
    }
}
```

**Output:**

D:\C programs\goback.exe

```
Enter no.of Frames to be sent: 5
Enter data for Frames[1]10
Enter data for Frames[2]20
Enter data for Frames[3]30
Enter data for Frames[4]40
Enter data for Frames[5]50
The Frame Number 2 is not received
Resending Frame 2
The Received Frame is 20
The Received Frame is 30
The Received Frame is 40
The Received Frame is 50
All Frames sent successfully
-----
Process exited after 17.83 seconds with return value 0
Press any key to continue . . .
```



## 7) Write a Program to implement Sliding window protocol for Selective repeat.

### Program:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<unistd.h>
int n,r;
struct frame
{
    char ack;
    int data;
}frm[10];
int sender(void);
void recvack(void);
void resend_sr(void);
int main()
{
    int c;
    sender();
    recvack();
    resend_sr();
    printf("\n All Frames sent successfully\n");
}
int sender()
{
    int i;
    printf("\n Enter no.of Frames to be sent: ");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        printf("\n Enter data for Frames[%d]",i);
        scanf("%d",&frm[i].data);
        frm[i].ack='y';
    }
    return 0;
}
void recvack()
{
    int i;
    rand();
    r=rand()%n;
    frm[r].ack='n';
}
```



```
for(i=1;i<=n;i++)
{
    if(frm[i].ack=='n')
        printf("\n The Frame Number %d is not received",r);
}
}
void resend_sr()
{
    printf("\nResending Frame %d",r);
    sleep(2);
    frm[r].ack='y';
    printf("\n The Received Frame is %d",frm[r].data);
}
```

**Output:**

```
D:\C programs\selective_repeat.exe

Enter no.of Frames to be sent: 5

Enter data for Frames[1]10

Enter data for Frames[2]20

Enter data for Frames[3]30

Enter data for Frames[4]40

Enter data for Frames[5]50

The Frame Number 2 is not received
Resending Frame 2
The Received Frame is 20
All Frames sent successfully

-----
Process exited after 17.84 seconds with return value 0
Press any key to continue . . .
```





## 8) Write a Program to implement Stop and Wait Protocol.

### Program:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<unistd.h>
int main()
{
    int i,j,packet[30];
    int fsize=(rand()%16)+1;
    printf("\n\t Frame size : %d \n",fsize);
    printf("\n\t ---  Data Log----- \n");
    printf("\n FRAME\t \tDATA\tWaiting\tACKNOW\tRESEND");
    for(i=0;i<fsize;i++)
    {
        packet[i]=rand()%1000;
        printf("\n %d \t\t %d",i+1,packet[i]);
        while(j=0 || rand()%2==0)
        {
            sleep(1);
            printf("\t1");
            sleep(1);
            for(j=2;rand()%2==0 && j<4;j++)
            {
                printf("%d ",j);
                sleep(1);
            }
            if(j==4)
                printf("\t NO \tRESENDING...\n %d \t %d",i+1,packet[i]);
            else
                break
        }
        if (j==0)
        {
            sleep(1);
            printf("\t0 ");
        }
        printf("\t\t YES \t NO");
    }
    printf("\n\n----- ALL DATA PACKETS SEND ----- \n");
    return 0;
```



Exp No:

Date:

Page No:

}

## Output:

```
C:\Users\RAMA REDDY\Downloads\week8.exe

Frame size : 10

--- Data Log ---

FRAME      DATA      Waiting ACKNOW  RESEND
1           467       12             YES      NO
2           724       12 3          NO      RESENDING...
2      724    0           YES      NO
3           145       0             YES      NO
4           827       0             YES      NO
5           491       0             YES      NO
6           942       0             YES      NO
7           436       0             YES      NO
8           604       1             YES      NO
9           292       1             YES      NO
10          716       1             YES      NO

----- ALL DATA PACKETS SEND -----

-----
Process exited after 20.77 seconds with return value 0
Press any key to continue . . .
```





## 9) Write a Program for congestion control using Leaky bucket Algorithm

### Program:

```
#include<stdio.h>
#include<stdlib.h>
struct packet
{
int time;
int size;
}p[50];
int main()
{
int i,n,m,k=0;
int bsize,bfilled,outrate;
printf("Enter the number of packets:");
scanf("%d",&n);
printf("Enter packets in the order of their arrival time\n");
for(i=0;i<n;i++)
{
printf("Enter the time and size:");
scanf("%d%d",&p[i].time,&p[i].size);
}
printf("Enter the bucket size:");
scanf("%d",&bsize);
printf("Enter the output rate:");
scanf("%d",&outrate);
m=p[n-1].time;
i=1;
k=0;
bfilled=0;
while(i<=m||bfilled!=0)
{
printf("\n\nAt time %d",i);
if(p[k].time==i)
{
if(bsize>=bfilled+p[k].size)
{
bfilled=bfilled+p[k].size;
printf("\n%d bytes packet is inserted",p[k].size);
k=k+1;
}
}
```



```
    else
    {
        printf("\n%d bytes packet is discarded",p[k].size);
        k=k+1;
    }
}
if(bfilled==0)
{
    printf("\nNo packets to transmit");
}
else if(bfilled>=outrate)
{
    bfilled=bfilled-outrate;
    printf("\n%d bytes transfered",outrate);
}
else
{
    printf("\n%d bytes transfered",bfilled);
    bfilled=0;
}
printf("\nPackets in the bucket %d bytes",bfilled);
i++;
}
return 0;
}
```





## Output:

```
Select C:\Users\RAMA REDDY\Downloads\week9\cnlab (1).exe
Enter the number of packets:5
Enter packets in the order of their arrival time
Enter the time and size:1
200
Enter the time and size:2
300
Enter the time and size:3
100
Enter the time and size:4
500
Enter the time and size:5
600
Enter the bucket size:400
Enter the output rate:100

At time 1
200 bytes packet is inserted
100 bytes transfered
Packets in the bucket 100 bytes

At time 2
300 bytes packet is inserted
100 bytes transfered
Packets in the bucket 300 bytes

At time 3
100 bytes packet is inserted
100 bytes transfered
Packets in the bucket 300 bytes

At time 4
500 bytes packet is discarded
100 bytes transfered
Packets in the bucket 200 bytes

At time 5
600 bytes packet is discarded
100 bytes transfered
Packets in the bucket 100 bytes

At time 6
100 bytes transfered
Packets in the bucket 0 bytes
-----
Process exited after 30.71 seconds with return value 0
Press any key to continue . . .
```



Exp No:  
Date:

Page No:

## Week – 10

**Write a Program to implement Dijkstra's algorithm to compute the Shortest path through a graph.**

### Program:

```
#include<stdio.h>
#define LARGE 10000
#define MAX 30
struct state
{
int len;
int pre;
int label;
};
struct state states[MAX];
int a[MAX][MAX];
int main()
{
int i,j,s,d,n,tem,min,mini;
printf("Enter no.of vertices:");
scanf("%d",&n);
printf("\nEnter adjacency matrix\n");
for(i=0;i<n;i++)
for(j=0;j<n;j++)
{
printf("a[%d][%d]=",i,j);
scanf("%d",&a[i][j]);
}
for(i=0;i<n;i++)
{
states[i].len=LARGE;
states[i].label=0;
states[i].pre=-1;
}
printf("\nEnter source vertex:");
scanf("%d",&s);
printf("\nEnter destination vertex");
scanf("%d",&d);
states[d].len=0;
```



Exp No:

Page No:

Date:

```
states[d].label=1;
tem=d;
while(tem!=s)
{
for(i=0;i<n;i++)
{
if(a[tem][i]!=0&&states[tem].len+a[tem][i]<states[i].len&&states[i].label=
=0)
{
states[i].len=states[tem].len+a[tem][i];
states[i].pre=tem;
}
}
min=LARGE;
mini=0;
for(i=0;i<n;i++)
{
if(states[i].len<LARGE&&states[i].label==0)
{
min=states[i].len;
mini=i;
}
}
states[mini].label=1;
tem=mini;
}
tem=s;
printf("\npath length:%d",states[s].len);
printf("\npath\n");
printf("%d",tem);
do
{
tem=states[tem].pre;
printf("%d",tem);
}
while(tem!=d);
return(0);
}
```

**Output:**



Exp No:  
Date:

Page No:

```
Enter no.of vertices:3
```

```
Enter adjacency matrix
```

```
a[0][0]=1
```

```
a[0][1]=0
```

```
a[0][2]=4
```

```
a[1][0]=0
```

```
a[1][1]=2
```

```
a[1][2]=0
```

```
a[2][0]=4
```

```
a[2][1]=0
```

```
a[2][2]=0
```

```
enter source vertex:0
```

```
Enter destination vertex2
```

```
path length:4
```

```
path
```

```
02
```

```
-----  
Process exited after 79.77 seconds with return value 0
```

```
Press any key to continue . . .
```







## Week – 11

**Write a Program to implement Distance vector routing algorithm by obtaining routing table at each node (Take an example subnet graph with weights indicating delay between nodes).**

### Program:

```
#include<stdio.h>
struct node
{
    unsigned dist[20];
    unsigned from[20];
}rt[10];
int main()
{
    int costmat[20][20];
    int nodes,i,j,k,count=0;
    printf("\nEnter the number of nodes : ");
    scanf("%d",&nodes);
    printf("\nEnter the cost matrix :\n");
    for(i=0;i<nodes;i++)
    {
        for(j=0;j<nodes;j++)
        {
            scanf("%d",&costmat[i][j]);
            costmat[i][i]=0;
            rt[i].dist[j]=costmat[i][j];
            rt[i].from[j]=j;
        }
    }
    do
    {
        count=0;
        for(i=0;i<nodes;i++)
        for(j=0;j<nodes;j++)
        for(k=0;k<nodes;k++)
            if(rt[i].dist[j]>costmat[i][k]+rt[k].dist[j])
            {
```



Exp No:  
Date:

Page No:

```
        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<nodes;i++)
{
    printf("\n\n For router %d\n",i+1);
    for(j=0;j<nodes;j++)
    {
        printf("\t\nnode %d via %d Distance %d",j+1,rt[i].from[j]+1,rt[i].dist[j]);
    }
}
printf("\n\n");
}
```

### Output:

```
Enter the number of nodes : 3

Enter the cost matrix :
0      2      7
2      0      1
7      1      0

For router 1
node 1 via 1 Distance 0
node 2 via 2 Distance 2
node 3 via 2 Distance 3

For router 2
node 1 via 1 Distance 2
node 2 via 2 Distance 0
node 3 via 3 Distance 1

For router 3
node 1 via 2 Distance 3
node 2 via 2 Distance 1
node 3 via 3 Distance 0

-----
Process exited after 32.26 seconds with return value 0
Press any key to continue . . .
```



Exp No:  
Date:

Page No:

## **Week – 12**

**Write a Program to implement Broadcast tree by taking subnet of hosts**

### **Program:**

```
#include<stdio.h>
int p,q,u,v,n;
int min=99,mincost=0;
int t[50][2],i,j;
int parent[50],edge[50][50];
int find(int r);
void sunion(int a,int b);
int main()
{
printf("\n Enter the number of nodes:");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("%c\t",65+i);
parent[i]=-1;
}
printf("\n");
for(i=0;i<n;i++)
{
printf("%c",65+i);
for(j=0;j<n;j++)
{
scanf("%d",&edge[i][j]);
}
}
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{
if(edge[i][j]!=99)
{
if(min>edge[i][j])
{
```



Exp No:  
Date:

Page No:

```
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);
}
else
{
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;i<n;i++)
{
if(t[i][0]!=-1 && t[i][1]!=-1)
{
printf("%c%c%d",65+t[i][0],65+t[i][1],edge[t[i][0]][t[i][1]]);
printf("\n");
}
}
return 0;
}
int find(int I)
{
if(parent[I]>0)
{
I=parent[I];
}
}
```



Exp No:

Page No:

Date:

```
return I;
}
void sunion(int I,int m)
{
parent[I]=m;
}
```

### Output:

Enter the number of nodes:4

A	B	C	D
A2	3	0	1
B3	6	9	1
C2	4	1	0
D2	5	2	1

Minimum Cost is 1

Minimum Spanning tree is

AC0

BD1

CD0

-----  
Process exited after 57.59 seconds with return value 0

Press any key to continue . . .