**5. Take an example subnet of hosts and obtain a broadcast tree for the subnet.**

#include<stdio.h>

int a[10][10],n;

void main()

{

int i,j,root;

printf("Enter no.of nodes:");

scanf("%d",&n);

printf("Enter adjacent matrix\n");

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

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

{

printf("Enter connecting of %d>%d::",i,j);

scanf("%d",&a[i][j]);

}

printf("Enter root node:");

scanf("%d",&root);

adj(root);

}

adj(int k)

{

int i,j;

printf("Adjacent node of root node::\n");

printf("%d\n",k);

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

{

if(a[k][j]==1 || a[j][k]==1)

printf("%d\t",j);

}

printf("\n");

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

{

if((a[k][j]==0) && (a[i][k]==0) && (i!=k))

printf("%d",i);

}

}

**output:**

Enter adjacent matrix

Enter connecting of 1>1::

1

Enter connecting of 1>2::1

Enter connecting of 1>3::11

Enter connecting of 2>1::2

Enter connecting of 2>2::3

Enter connecting of 2>3::0

Enter connecting of 3>1::1

Enter connecting of 3>2::4

Enter connecting of 3>3::2

Enter root node:3

Adjacent node of root node::

3

1

2

**6. Implement distance vector routing algorithm for obtaining routing tables at each node.**

#include<stdio.h>

struct node

{

unsigned dist[20];

unsigned from[20];

}rt[10];

int main()

{

int dmat[20][20];

int n,i,j,k,count=0;

printf("\nEnter the number of nodes : ");

scanf("%d",&n);

printf("\nEnter 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("\n\nState value for router %d is \n",i+1);

for(j=0;j<n;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 :

1 2 3

4 5 6

7 8 9

State value for router 1 is

node 1 via 1 Distance0

node 2 via 2 Distance2

node 3 via 3 Distance3

State value for router 2 is

node 1 via 1 Distance4

node 2 via 2 Distance0

node 3 via 3 Distance6

State value for router 3 is

node 1 via 1 Distance7

node 2 via 2 Distance8

node 3 via 3 Distance0

**7. Implement data encryption and data decryption**

#include <stdio.h>

int main()

{

int i, x;

char str[100];

printf("\nPlease enter a string:\t");

gets(str);

printf("\nPlease choose following options:\n");

printf("1 = Encrypt the string.\n");

printf("2 = Decrypt the string.\n");

scanf("%d", &x);

//using switch case statements

switch(x)

{

case 1:

for(i = 0; (i < 100 && str[i] != '\0'); i++)

str[i] = str[i] + 3; //the key for encryption is 3 that is added to ASCII value

printf("\nEncrypted string: %s\n", str);

break;

case 2:

for(i = 0; (i < 100 && str[i] != '\0'); i++)

str[i] = str[i] - 3; //the key for encryption is 3 that is subtracted to ASCII value

printf("\nDecrypted string: %s\n", str);

break;

default:

printf("\nError\n");

}

return 0;

}

**output:**

**for encryption:**

Please enter a string: hi

Please choose following options:

1 = Encrypt the string.

2 = Decrypt the string.

1

Encrypted string: kl

**for decryption:**

Please enter a string: kl

Please choose following options:

1 = Encrypt the string.

2 = Decrypt the string.

2

Decrypted string: hi

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

#include<stdio.h>

int main(){

int incoming, outgoing, buck\_size, n, store = 0;

printf("Enter bucket size, outgoing rate and no of inputs: ");

scanf("%d %d %d", &buck\_size, &outgoing, &n);

while (n != 0) {

printf("Enter the incoming packet size : ");

scanf("%d", &incoming);

printf("Incoming packet size %d\n", incoming);

if (incoming <= (buck\_size - store)){

store += incoming;

printf("Bucket buffer size %d out of %d\n", store, buck\_size);

} else {

printf("Dropped %d no of packets\n", incoming - (buck\_size - store));

printf("Bucket buffer size %d out of %d\n", store, buck\_size);

store = buck\_size;

}

store = store - outgoing;

printf("After outgoing %d packets left out of %d in buffer\n", store,

buck\_size);

n--;

}

}

**output:**

Enter bucket size, outgoing rate and no of inputs: 10

10

3

Enter the incoming packet size : 12

Incoming packet size 12

Dropped 2 no of packets

Bucket buffer size 0 out of 10

After outgoing 0 packets left out of 10 in buffer

Enter the incoming packet size : 8

Incoming packet size 8

Bucket buffer size 8 out of 10

After outgoing -

2 packets left out of 10 in buffer

Enter the incoming packet size : 12

Incoming packet size 12

Bucket buffer size 10 out of 10

After outgoing 0 packets left out of 10 in buffer

**9. Write a program for frame sorting technique used in buffers.**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct frame

{

int sno;

char msg[15];

int flag;

};

int main()

{

int i,j,n,r,k;

printf("enter no of frames\n");

scanf("%d",&n);

struct frame fr[n];

int s[n];

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

{

s[i]=-1;

fr[i].sno=-1;

}

printf("enter the message \n");

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

{

scanf("%s",fr[i].msg);

fr[i].sno=i;

}

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

{

r=rand()%n;

if(s[r]==-1)

{

fr[j].flag=r;

s[r]=1;

}

else if(s[r]==1)

{

for(k=0;k<n;k++){

r=k;

if(s[r]==-1)

{

fr[j].flag=r;

s[r]=1;

break;

}

}

}

}

printf("arrived frame are:\n");

printf("\n sno \t msg \n");

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

{

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

if(fr[j].flag==i)

{

printf("%d\t%s",fr[j].sno,fr[j].msg);

printf("\n");

}

}

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

{

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

{

if(fr[j].sno>fr[j+1].sno)

{

struct frame temp;

temp=fr[j];

fr[j]=fr[j+1];

fr[j+1]=temp;

}

}

}

printf("after sorting arrived frames are\n");

printf("\n sno \t msg \n");

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

{

printf("%d\t%s",fr[i].sno,fr[i].msg);

printf("\n");

}

return 0;

}

**output:**

enter no of frames

5

enter the message

today

you

have

to

go

arrived frame are:

sno msg

3 to

1 you

2 have

0 today

4 go

after sorting arrived frames are

sno msg

0 today

1 you

2 have

3 to

4 go