ADA LAB TEST-2

VAGHDEVI PRAVEEN T

1BM19CS175

```
#include<stdio.h>
void main()
{
int a[20][20],b[20][20],c[20][20],d[20][20],nod=0,n,val1=0,i,j,k,t,m=0,posx,posy,val;
printf("\nEnter the value of n:");
scanf("%d",&n);
printf("\nEnter the adjacency matrix\n");
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{
scanf("%d",&a[i][j]);
b[i][j]=(i==j?0:a[i][j]);
m=m+(b[i][j]?1:0);
c[i][j]=0;
d[i][j]=0;
}
}
for(m=m/2;m!=0&&(nod!=(n-1));m--)
{
val=32767;
for(i=0;i<n;i++)
```

```
{
for(j=0;j<n;j++)
{
if(b[i][j]!=0&&b[i][j]<val)
{
posx=i;
posy=j;
val=b[i][j];
}
}
}
b[posx][posy]=0;
b[posy][posx]=0;
if(c[posx][posy]==0)
{
c[posx][posy]=1;
c[posy][posx]=1;
for(k=0;k<n;k++)
{
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{
c[i][j]=c[i][j]|(c[i][k]&c[k][j]);
}
```

```
}
}
val1=val1+a[posx][posy];
nod=nod+1;
d[posx][posy]=a[posx][posy];
d[posy][posx]=a[posy][posx];
}
}
if(nod==n-1)
{
for(i=0;i<n;i++)
{
printf("\n");
for(j=0;j<n;j++)
{
printf("%d ",d[i][j]);
}
printf("\nSpanning tree has a cost of %d",val1);
}
else
{
printf("\nSpanning tree does not exist!!");
}
}
```

compiler

Run Debug Stop Share Save { } Beautify

main.c

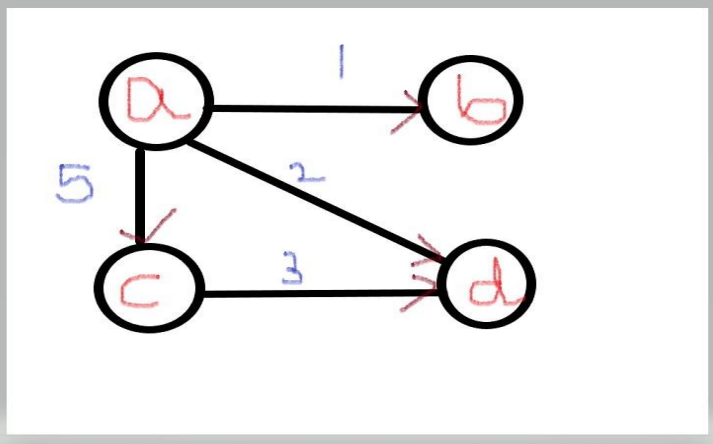
input

```
Enter the value of n:4

Enter the adjacency matrix
0 1 5 2
1 0 0 0
5 0 0 3
2 0 3 0

0 1 0 2
1 0 0 0
0 0 0 3
2 0 3 0
Spanning tree has a cost of 6

...Program finished with exit code 0
Press ENTER to exit console.
```



MODIFICATION:-

```
#include<stdio.h>
void main()
{
int a[20][20],b[20][20],c[20][20],d[20][20],nod=0,n,val1=0,i,j,k,t,m=0,posx,posy,val;
printf("\nEnter the value of n:");
scanf("%d",&n);
printf("\nEnter the adjacency matrix\n");
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{
scanf("%d",&a[i][j]);
b[i][j]=(i==j?0:a[i][j]);
m=m+(b[i][j]?1:0);
c[i][j]=0;
d[i][j]=0;
}
}
for(m=m/2;m!=0&&(nod!=(n-1));m--)
{
val=32767;
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
```

```
{
if(b[i][j]!=0&&b[i][j]<val)
{
posx=i;
posy=j;
val=b[i][j];
}

}
}
b[posx][posy]=0;
b[posy][posx]=0;
if(c[posx][posy]==0)
{
c[posx][posy]=1;
c[posy][posx]=1;
for(k=0;k<n;k++)
{
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{
c[i][j]=c[i][j]|(c[i][k]&c[k][j]);
}
}
```

```
}
val1=val1+a[posx][posy];
nod=nod+1;
d[posx][posy]=a[posx][posy];
d[posy][posx]=a[posy][posx];
}

}
if(nod==n-1)
{
for(i=0;i<n;i++)
{
printf("\n");
for(j=0;j<n;j++)
{
printf("%d ",d[i][j]);
}
}
printf("\nSpanning tree has a cost of %d",val1);
}
else
{
printf("\nSpanning tree does not exist!!");
```

```
    if(nod && val1)    printf("\n\nIt is a Connected, Cyclic Graph!");
    if(!nod && val1)   printf("\n\nIt is a Not-Connected, Acyclic Graph!");
    if(nod && !val1)   printf("\n\nGraph is a Connected, Cyclic Graph!");
    if(!nod && !val1)  printf("\n\nIt is a Not-Connected, Cyclic Graph!");
}
}
```