**/\*10.Find minimum cost spanning tree of a given undirected path using a Prim’s**

**algorithm.\*/**

#include<stdio.h>

#include<stdlib.h>

#define inf 9999

int a[10][10],spanning[10][10],n;

int prims();

void main()

{

int i,j,total\_cost;

clrscr();

printf("enter number of vertices:");

scanf("%d",&n);

printf("\n enter the adjaceny matrix:\n");

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

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

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

total\_cost=prims();

printf("\n spanning tree matrix:\n");

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

{

printf("\n");

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

printf("%d\t",spanning[i][j]);

}

printf("\n\n total cost of spanning tree=%d",total\_cost);

getch();

}

int prims()

{

int cost[10][10];

int u,v,min\_dis,dis[10],from[10];

int vis[10],no\_of\_edges,i,min\_cost,j;

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

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

{

if(a[i][j]==0)

cost[i][j]=inf;

else

cost[i][j]=a[i][j];

spanning[i][j]=0;

}

dis[0]=0;

vis[0]=1;

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

{

dis[i]=cost[0][i];

from[i]=0;

vis[i]=0;

}

min\_cost=0;

no\_of\_edges=n-1;

while(no\_of\_edges>0)

{

min\_dis=inf;

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

if(vis[i]==0&&dis[i]<min\_dis)

{

v=i;

min\_dis=dis[i];

}

u=from[v];

spanning[u][v]=dis[v];

spanning[v][u]=dis[v];

no\_of\_edges--;

vis[v]=1;

for(i=1;i<n;

i++)

if(vis[i]==0&&cost[i][v]<dis[i])

{

dis[i]=cost[i][v];

from[i]=v;

}

min\_cost=min\_cost+cost[u][v];

}

return(min\_cost);

}

**/\*OUTPUT**

enter number of vertices:

4

enter the adjaceny matrix:

0 20 10 0

20 0 0 30

10 0 0 0

0 30 0 0

spanning tree matrix:

0 20 10 0

20 0 0 30

10 0 0 0

0 30 0 0

total cost of spanning tree=60

**\*/**

**/\*OUTPUT**

enter number of vertices:

4

enter the adjaceny matrix:

0 20 10 0

20 0 0 30

10 0 0 0

0 30 0 0

spanning tree matrix:

0 20 10 0

20 0 0 30

10 0 0 0

0 30 0 0

total cost of spanning tree=60

**\*/**

**/\*OUTPUT**

enter number of vertices:

4

enter the adjaceny matrix:

0 20 10 0

20 0 0 30

10 0 0 0

0 30 0 0

spanning tree matrix:

0 20 10 0

20 0 0 30

10 0 0 0

0 30 0 0

total cost of spanning tree=60

**\*/**

**/\*OUTPUT**

enter number of vertices:

4

enter the adjaceny matrix:

0 20 10 0

20 0 0 30

10 0 0 0

0 30 0 0

spanning tree matrix:

0 20 10 0

20 0 0 30

10 0 0 0

0 30 0 0

total cost of spanning tree=60

**\*/**