PRIM’S ALGORITHM

#include<stdio.h>

#include<stdlib.h>

#define infinity 9999

#define MAX 20

**int** G[MAX][MAX],spanning[MAX][MAX],n;

**int** prims();

**int** main()

{

**int** i,j,total\_cost;

**printf**("Enter no. of vertices:");

scanf("%d",&n);

**printf**("\nEnter the adjacency matrix:\n");

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

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

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

total\_cost=prims();

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

**return** 0;

}

**int** prims()

{

**int** cost[MAX][MAX];

**int** u,v,min\_distance,distance[MAX],from[MAX];

**int** visited[MAX],no\_of\_edges,i,min\_cost,j;

*//create cost[][] matrix,spanning[][]*

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

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

{

**if**(G[i][j]==0)

cost[i][j]=infinity;

**else**

cost[i][j]=G[i][j];

spanning[i][j]=0;

}

*//initialise visited[],distance[] and from[]*

distance[0]=0;

visited[0]=1;

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

{

distance[i]=cost[0][i];

from[i]=0;

visited[i]=0;

}

min\_cost=0; *//cost of spanning tree*

no\_of\_edges=n-1; *//no. of edges to be added*

**while**(no\_of\_edges>0)

{

*//find the vertex at minimum distance from the tree*

min\_distance=infinity;

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

**if**(visited[i]==0&&distance[i]<min\_distance)

{

v=i;

min\_distance=distance[i];

}

u=from[v];

*//insert the edge in spanning tree*

spanning[u][v]=distance[v];

spanning[v][u]=distance[v];

no\_of\_edges--;

visited[v]=1;

*//updated the distance[] array*

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

**if**(visited[i]==0&&cost[i][v]<distance[i])

{

distance[i]=cost[i][v];

from[i]=v;

}

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

}

**return**(min\_cost);

}

OUTPUT

Enter no. of vertices:6

*Enter the adjacency matrix:*  
*0 3 1 6 0 0*  
*3 0 5 0 3 0*  
*1 5 0 5 6 4*  
*6 0 5 0 0 2*  
*0 3 6 0 0 6*  
*0 0 4 2 6 0*

*spanning tree matrix:*

*0 3 1 0 0 0*  
*3 0 0 0 3 0*  
*1 0 0 0 0 4*  
*0 0 0 0 0 2*  
*0 3 0 0 0 0*  
*0 0 4 2 0 0*