**Program No.:-10**

**AIM: Write a program to implement prim’s algorithm for calculating minimum spanning tree.**

**Source Code:**

#include<stdio.h>

#include<conio.h>

#include<limits.h>

#include<stdlib.h>

char vertex[10];

int weight[10][10],key[10],size,visited[10],parent[10],min,min\_index,i,j,u;

void build\_graph()

{

printf("enter number of vertices");

scanf("%d",&size);

printf("enter %d vertices of graph",size);

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

{

fflush(stdin);

scanf("%c",&vertex[i]);

}

printf("enter weighted matrix for the graph:\n");

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

{

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

{

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

}

}

}

int extract\_min()

{

min=INT\_MAX;

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

{

if(!visited[i]&&key[i]<min)

{

min=key[i];

min\_index=i;

}

}

return min\_index;

}

void prims()

{

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

{

key[i]=INT\_MAX;

visited[i]=0;

parent[i]=NULL;

}

key[0]=0;

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

{

u=extract\_min();

visited[u]=1;

printf("%c-%c = %d\n",vertex[parent[u]],vertex[u],weight[parent[u]][u]);

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

{

if(!visited[i] && weight[u][i]!=0)

{

if(key[i]>weight[u][i])

{

key[i]=weight[u][i];

parent[i]=u;

}

}

}

}

}

void main()

{

clrscr();

build\_graph();

printf("the nodes which form the minimum spanning tree are:\n\nnode weight of node\n");

prims();

getch();

}

**Output:**



