**Program No.:-9**

**AIM: Write a program to implement kruskal’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],size,set\_index[10],min,i,j,u,k,v,range=1,temp;

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]);

set\_index[i]=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]);

}

}

}

void kruskal()

{

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

{

min=INT\_MAX;

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

{

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

{

if(weight[i][j]<min && weight[i][j]>=range && set\_index[i]!=set\_index[j])

{

min=weight[i][j];

u=i;

v=j;

}

}

}

range=min;

if(set\_index[u]!=set\_index[v])

{

temp=set\_index[v];

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

{

if(set\_index[i]==temp)

{

set\_index[i]=set\_index[u];

}

}

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

}

}

}

void main()

{

clrscr();

build\_graph();

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

kruskal();

getch();

}

**Output:**



