ADA LAB TEST-2 (KRUSHKALS)

NAME - Rituika Singh Rathore
USN - 1BM19CS131
SEC - C
SEM - 4
Date: 8/7/21

Program - Find minimum cost of spanning tree of given undirected graph using Kruskal's algorithm.

```
#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("Enter the value of n");
   scanf("%d", &n);
   printf("Enter the adjacency matrix");
   for(i=0; i<n; i++){
      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 && (mod!=(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]=1;
     c[posy][posx]=1;
     for(k=0; k<n; k++){
       ~~for(k=0~~
         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];
       mod = mod+1;
       d[posx][posy] = a[posx][posy];
       d[posy][posx] = a[posy][posx];
       }}
```

```
if (ned == n-1)
{   for (i=0; j<n; j++){
    printf("%d", d[i][j]);
}
printf("\n Spanning tree cost is %d", val1);
}
else{
    printf("Spanning tree doesn't exist");
}}
```

Modified Program - ~~If it~~ while finding MST using Kruskal's algorithm. If u come accross cycle, print the vertices of in the cycle.

```
# include< stdio.h>
   void Kruskals();
 int cost[10][10],min, n, i, j, count, k, u, v, parent[10], sum,
   t[10][2], num_edge=0;
   void union_ij(int, int);
   int find(int);

   void main(){
printf("Enter no. of vertices");
scanf("%d", &n);
printf("Enter adjacency matrix");
  for(i=0; i<n; i++){
    for(j=0; j<n; j++){
      scanf("%d", &cost[i][j]);
      if(cost[i][j]!=0 && cost[i][j]<999)num_edge++;
    }}
    Kruskals();
  }
```

```
void Kruskals(){
  count =0;
  k=0;
  sum=0;
for(i=0; i<n; i++){
  parent[i] = i; }
while(count! =num_edge)
 { min= 999;
   for(i=0; i<n; i++){
     for(j=0; j<n; j++){
 if cost[i][j] < min and cost[i][j] != 0 }
  { min = cost[i][j];
    u=i;
    v=j;
 }}}
  i = find(u);
  j = find(v);
 if(i==j){
    printf("Vertex forming cycle are : %d and %d",
          u,v);
       count++;
       }
```

```
else if (i!=j)
{ t[k][0]=u;
  t[k][1]=v;
  k++;
  count++;
  sum=sum+cost[u][v];
  union_ij(i,j);
}
  else {
       count++;
     }
  cost[u][v]=cost[v][u]=999;
  }
printf("Minimum Spanning Tree \n");
     for(i=0; i<=n-1; i++){
          if(t[i][0]!=t[i][1])
               printf("%d → %d ", t[i][0], t[i][
     printf("Total cost = %d", sum);
   }
void union_ij(int i, int j){
     if(i<j)
       parent[j]=i;
  else
    parent[i]=j; }
```

```
int find (int v) {
while ( parent[v] != v )
        v = parent[v];
    return v;
}.
```