ADA Lab-test-2 Name: N. AKhilesh USN: 1BM19C5092

section: 48 semester: 4th sign: Akhilesh

```
End the Minimus cost of sharring tree of a given
 undirected graph using Kruskals algorithm:
   # include / stdio. h >
   # include z Process. h>
     void Kruskalsco:
     int cost [10][10], n, i, i, sum, min;
     int count, K, u, V, Parkert[10];
     int those for
     void union_id(int, int);
     int find Cint);
     int major )
     Print ("Enter the No: of Vertices:");
      scanf ("dod", En);
      Printf ("Enter the cost Adjacency matrix: /v").
     gfor (i=o; icn; itt)
      for Co=o; jen; j++)
      3 Scarf("old", & cost [][i]);
      Knuskaly);
     refine o?
   g void Kryskals ()
      count = 0;
      K=0;
Sum=0;
      for (100; Kn; 17+1)
      Parent[i]=i:
```

Page-No:1

```
cubileccount 1=n-1)
 for Cizo; icu; itt)
  (++6; N36; 0=1) not
  (0=![6][1] teos & rims [6][1] teos) 7:3
   ¿ [ illisteos=vimes
   3V=j.
 =tind(u);
 j=find(v);
 if (i)= i)
 tEKTEGT=U'
 t[K][i]=V;
  count ++ ',
  Sum - Sum+ cost Eul[v];
  union_isci, i);
  cost EUTEVI = cost EVI EU = 9999;
Print ("Minimum spanning tree:16");
for (i=0; icn-1; i++)
3 Print((10/d -->%d", tei)[0], tei)[1);
  Printf("Total cost = %d", sum);
void union is Cinti, into)
  ifcical
  Parent[d]=i';
  e13e
 Parent[i]=1;
  "int find (int v)
 E while (Parent [v] !=v)
    N=Parent(V).
  2 return v;
```

Name: N-Akhilesh USN: 18M19cso92 Sign: Akhilosh

Modification :-

while finding the NOT using knuskal algorithm if you come acrossa cycle Print the vertices in the cycle

```
void Kruskalsco
count = 0;
for (i=0; i60; i++)
  Parent [i]= i';
 while (court ! to-1)
   min=999;
  for (1:0; 1210; 1++)
   for (i=0; i<n; i+t)
    min=costciJ[i];
      はニッ
      V= 6',
  i=find (u);
   : (v) brit= 6
   if ( i== i)
    Printf (a vertices forming cycle are: % d and
               olog /2, 120;
   3
   if (i) = 3)
     + CKJEB= u;
     t CKJ CIJ = V;
      K++'
     count++;
```

Page-No:3

```
Sum = Sum+ cost [u][v];
    coin is Cital;
    COSECUTED = COSECUTED =999;
   Printf ("Minimal spanning tree: In");
   for ( i= 0; ick; i++)
     Printf("%d->%d", t[i][o], t[i][i];
    Printf (upototal cost = 1/2/10", sum);
3
  int main()
   Printf (" Enter the No: of vertices; ");
   Printle (" Enter the Got Adjacency Matrix: 1vi");
    for Ciso; icn; i++)
     for (i=0; ico; i++)
       Sanf ("0/64", & cost [] [13);
      Kruskals();
      return o;
```