Start here | KRUSKAL'S ALGORITHM.c

```c
#include<stdio.h>
#include<conio.h>
void kruskals();
int c[10][10],n;
void main(){
    int i,j;
    printf("\nEnter the no. of vertices:\n");
    scanf("%d",&n);
    printf("\nEnter the cost matrix:\n");
    for(i=1;i<=n;i++){
        for(j=1;j<=n;j++){
            scanf("%d",&c[i][j]);
        }
    }
    kruskals();
    getch();
}
void kruskals(){
    int i,j,u,v,a,b,min;
    int ne=0,mincost=0;
    int parent[10];
    for(i=1;i<=n;i++){
        parent[i]=0;
    }
    while(ne!=n-1){
        min=9999;
        for(i=1;i<=n;i++){
            for(j=1;j<=n;j++){
                if(c[i][j]<min){
                    min=c[i][j];
                    u=a=i;
                    v=b=j;
                }
            }
        }
        while(parent[u]!=0){
            u=parent[u];
        }
        while(parent[v]!=0){
            v=parent[v];
        }
```

Start here | KRUSKAL'S ALGORITHM.c

```
12              scanf("%d",&c[i][j]);
13          }
14      }
15      kruskals();
16      getch();
17  }
18  void kruskals(){
19      int i,j,u,v,a,b,min;
20      int ne=0,mincost=0;
21      int parent[10];
22      for(i=1;i<=n;i++){
23          parent[i]=0;
24      }
25      while(ne!=n-1){
26          min=9999;
27          for(i=1;i<=n;i++){
28              for(j=1;j<=n;j++){
29                  if(c[i][j]<min){
30                      min=c[i][j];
31                      u=a=i;
32                      v=b=j;
33                  }
34              }
35          }
36          while(parent[u]!=0){
37              u=parent[u];
38          }
39          while(parent[v]!=0){
40              v=parent[v];
41          }
42          if(u!=v){
43              printf("\n%d----->%d=%d\n",a,b,min);
44              parent[v]=u;
45              ne=ne+1;
46              mincost=mincost+min;
47          }
48          c[a][b]=c[b][a]=999;
49      }
50      printf("\nMinimun cost=%d",mincost);
51  }
52
```

"C:\web developement(html.css.js)\KRUSKAL'S ALGORITHM.exe"

```
Enter the no. of vertices:
6

Enter the cost matrix:
999 3 999 999 6 3
3 999 1 999 999 4
999 1 999 6 999 4
999 6 6 999 8 5
6 999 999 8 999 2
5 4 4 5 2 999

2------>3=1

5------>6=2

1------>2=3

1------>6=3

4------>6=5

Minimun cost=14
```