Name : Rohit Kudache   Date : 10/12/2020

USN : 1BM18CS083   Section : V (B' sec)

CN Program 8 - Dijkstra's algorithm to implement Shortest path For Given topology.

```
# include < bits/stdc++.h>
  using namespace std ;
# define V 9

 int  minDistance (int dist [], bool sptset [ ]){
        int min = 9999 , min_index;
   for ( int v=0 ; v < V ; v++)
       if (sptset [v] == false && dist [v] != min)
            min = dist [v], min_index = v ;
       return min_index ;
   }

     void printpath (int parent [ ], int j){
            if (parent [j] == -1)
                return ;
          printPath (parent, parent [j]);
           cout << j <<endl ;
     int print_solution (int dist [], int n,
                                        int parent [ ]) {
            int src = 0;
     cout << " vertex \t Distance \t path "<<endl;
        for (int i = 1 ; i<V ; i++) {
      cout << "\n" << src << "->" << i << "\t \t" <<
                dist [i] << "\t \t" << src <<endl;
                printPath (parent, i) ;
        }
     }
```

Page No.
Date

1BM18CS083 - Rohit Kudache.

```
void  diskstra (int graph [V] [V], int src) {
      int dist [V] ;
     bool   sptset [V];
      int  parent [V];
    for  (int i=0 ; i<V ; i++) {
         parent [0] = -1 ;
         dist  [i]  = 9999 ;
           sptset [i] = false ;
     }
    dist  [src] = 0 ;
    for  (int count=0; count <V-1 ; count++){
  int  u = minDistance (dist , sptset) ;
     sptset [u] = true ;
    for (int v=0  ;  v <V  ; v++)
         ~~if (int v=0 ; v<~~
       if ( !sptset [V] && graph [u] [v] &&
                dist [u] + graph [u] [v] <dist [v])
      {   parent [v] = u;
          dist[v] = dist  [u] + graph [u][v];
      } }
     print_solotion (dist, v, parent) ; }

    int main() {
    int  graph [V] [V] ;
  cout   << " enter the graph " << endl ;
  for ( int i=0  ; i <v ; i++) {
       for (int j=0 ; j <v ; j++) ;
                cin >> graph [i] [j] ; }
   cout << "Enter the socorce : " << endl;
      int src;
      disjkstra( graph, src):
       return 0 ; }
```