# CN lab 9

```
class graph():
        def _init_(self, vertices):
              self.V = vertices
              self.graph = [[0 for column in range(v)]
                                    for row in range(v)]

       def print-solution(self, dist):
              print("Vertex \t Distance from Source")
              for node in range(self, V):
                     print(node, "\t", distance[node])

       def min_distance(self, dist, sptSet):
            min = 9999
            for v in range(Self.v):
                 if dist[v] < min and sptSet[v] == False:
                       min = dist[v]
                       min_in = v
            return min_index

    def add-edge(self, src, dest, weight):
          self. graph[src][dist] = self. graph[dest][src]
                                              = weight
```

```
def  dijstra(self, src) :
     dist=[9999] + self.V
     dist [src] = 0
     sptSet = [False] * self.V

     for cout in range(self.V):
        u=self. min_distance (dist, sptSet)
        sptSet[u] = True
        for v in range(self.V):
            if  self. graph [u][v] >0 and sptSet [v]
                   == False and dist[v] > dist[u]
                + self. graph [u][v] :
                 dist [v] = dist[u] + self. graph [u] [v]

     self. print_solution (dist)
```