S.R.PUNEETH
1BM18CS087

CN LAB - 9

WAP to implement Dijkstra's algorithm to find shorter path.

```
int minDistance (int dist[], bool sptSet[])
 {
  int min = INT_MAX, 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 printSolution (int dist[])
 {
  printf("vertex \t \t Distance from Source \n");
  for (int i = 0; i < V; i++)
   printf("%d \t \t %d \n", i, dist[i]);
 }
void dijkstra (int graph[V][V], int src)
 {
  int dist[V];
  bool sptSet[V];
  for (int i = 0; i < V; i++)
   dist[i] = INT_MAX, sptSet[i] = false;
   dist[src] = 0;
   for (int count = 0; count < V-1; count++)
    {
    int u = minDistance (dist, sptSet);
    spt set [u] = true;
```


S.R.Puneeth

S.R.PUNEETH
1BM18CS087

```
for (int v=0; v<V; v++)
 if (!sptSet[v] && graph[u][v] && dist[u]!= INT_MAX
     && dist[u] + graph[u][v] < dist[v])
     dist[v] = dist[u] + graph[u][v];
 }

 printSolution(dist);
}
```


S.R.Puneeth