Hemanth Kumar V.C
1BM18CS037
Sec 'A'

Dijkstra's algorithm to compute shortest path through a graph

```
# include <bits/stdc++.h>
  using namespace std;
 # define V5
 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 printsolution ( 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;
```

```
            print path (parent , i );
        }
    }
void dijkstra (int graph [V][V] , int src)
{
        int dist [V];
        bool sptset [V];
        int parent [V];
        for (int i=0; i<V; i++)
        {
            parent [v] = -1;
            dist(i) = 9999;
            sptset [i]= false ;
        }
        dist [src] = 0;
```