CHIRANJEEVI
1BM19CS403

# CN-LAB

Batch 3

## Dijkshra's Algorithm to compute shortest path

```
#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] == fake && 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<<" ";
     }
int   printSolution (int dist[], int n, int parent[])
      {
        int  src=0
        cout<<"Vertex \t Distane \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);.
        }
     }
```

chy

IBM19CS403
CHIRANJEEVI

```
void dijkistra (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 = minDistane( dist, Sptset),
        sptset [u]= true;
        for ( int v=0; v<V; v++)
        if(!sptset[v] && graph[u][v] && dist[u]+ graph[u][v]
                < dist [v])
            {
                parent [v] =u;
                dist[v] =dist[u] + graph[v][v];
            }
    }
    printsolution (dist, v, parent);
}
```