```cpp
/// Dijkstra
#include<bits/stdc++.h>
using namespace std;
#define Inf 9999
vector<int>adj[100];
int node, edge, d[100], vis[100], cost[100][100];

struct st{
    int u,w;
    st(int a, int b){ u=a;w=b; }
    bool operator <(const st& p)const{
        return p.w < w;
    }
};

void Dijkstra(int src){
    memset(d,Inf,sizeof(d));
    memset(vis,0,sizeof(vis));
    priority_queue<st>pq;

    d[src]=0;
    pq.push(st(src,d[src]));

    while(!pq.empty()){
        st top = pq.top(); pq.pop();
        int u = top.u;
        if(vis[u]==1) continue;

        for(int i=0; i<adj[u].size(); i++){
            int v=adj[u][i];
            if(d[u]+cost[u][v] < d[v]){
                d[v]=d[u]+cost[u][v];
                pq.push(st(v,d[v]));
            }
        }
        vis[u]=1;
    }
}
int main(){
    int u,v,w,src;
    printf("Enter the Number of Node: ");
    cin >> node;
    printf("Enter the Number of Edge: ");
    cin >> edge;

    printf("Enter all the Edges\n");
    for(int i=0; i<edge; i++){
        cin >> u >> v >> w;
        adj[u].push_back(v);
        adj[v].push_back(u);
        cost[u][v]=w;
        cost[v][u]=w;
    }
    printf("Enter the Source: ");
    cin >> src;

    Dijkstra(src);

    for(int i=1; i<=node; i++){
        if(d[i]==Inf) printf("Not possible to go %d to %d.\n",src,i);
        else printf("%d to %d distance %d.\n",src,i,d[i]);
    }

    return 0;
}
```