From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.

```
#include <stdio.h>
int dist[10], cost[100][100], n, vis[10], src;
void dijkstra()
{     int count, min, u;     for
(int i = 1; i <= n; i++)
   {        dist[i] =
cost[src][i];        vis[src]
= 1;
   }     count = 1;
while (count < n)
   {        min = 9999;        for
(int i = 1; i <= n; i++)
      {
         if (dist[i] < min && vis[i] == 0)
         {               min
= dist[i];               u =
i;
         }        }        vis[u] =
1;        for (int i = 1; i <= n;
i++)
      {
         if (dist[u] + cost[u][i] < dist[i] && vis[i] == 0)
         {               dist[i] = dist[u] +
cost[u][i];           }

      }
```

```
        count++;
    }
}

void main() {     int m, u, v, w;     printf("\n
Dijkstra's Algorithm\n");     printf("    -------
----------");     printf("\nEnter the
number of vertices: ");     scanf("%d", &n);
for (int i = 1; i <= n; i++)
    {         for (int j = 1; j <= n;
j++)
        {             if (i ==
j)             {
cost[i][j] = 0;
            }             else
{                 cost[i][j] =
9999;
            }
        }
    }
    printf("Enter the number of edges: ");
scanf("%d", &m);     printf("Enter the edge
with its weight\n");     for (int i = 1; i <= m;
i++)

    {
        scanf("%d%d%d", &u, &v, &w);
cost[v][u] = cost[u][v] = w;
```

```
    }
    printf("Enter the source\n");
scanf("%d", &src);     dijkstra();

    printf("\n");     for (int i = 2; i <= n; i++)         printf("The
distance from %d --> %d is %d\n", src, i, dist[i]);
} TRA
```

OUTPUT: