```
#include <iostream>

#include <vector>

#include <limits.h>

using namespace std;

#define V 6

int minDistance(vector<int> dist, vector<bool> visited) {

  int min = INT_MAX, min_index = -1;

  for (int v = 0; v < V; v++) {

    if (!visited[v] && dist[v] <= min) {

      min = dist[v];

      min_index = v;

    }

  }

  return min_index;

}

void dijkstra(int graph[V][V], int src)

{

  vector<int> dist(V, INT_MAX);

  vector<bool> visited(V, false);

  dist[src] = 0;

  for (int count = 0; count < V - 1; count++)

{

    int u = minDistance(dist, visited);

    visited[u] = true;

    for (int v = 0; v < V; v++) {

      if (!visited[v] && graph[u][v] && dist[u] != INT_MAX &&
            dist[u] + graph[u][v] < dist[v]) {
            dist[v] = dist[u] + graph[u][v];
          }
        }
    }
    cout << "Vertex \t Distance from Source\n";
    for (int i = 0; i < V; i++) {
        cout << i << " \t " << dist[i] << endl;
    }
}
int main() {
    int graph[V][V] = { {0, 4, 0, 0, 0, 0}, {4, 0, 8, 0, 0, 0}, {0, 8, 0, 7, 0, 4},{0, 0, 7, 0, 9, 14},
    {0, 0, 0, 9, 0, 10},{0, 0, 4, 14, 10, 0} };
    int source = 0;
    dijkstra(graph, source);
    return 0;
}
```

Output:

```
Vertex  Distance from Source
0       0
1       4
2       12
3       19
4       26
5       16
```