The C program implements Dijkstra's algorithm to find the shortest paths from a single source vertex to all other vertices in a weighted graph with non-negative edge weights. The graph is represented using an adjacency matrix.

Code Structure & Functionality

Header Files

stdio.h is used for input/output operations.

limits.h is often included for the INT_MAX constant, representing infinity.

Macros

V defines the number of vertices in the graph (fixed-size implementation).

Functions

a) minDistance(int dist[], int sptSet[])

Purpose: Finds the vertex with the minimum distance value from the set of vertices not yet processed.

Logic: Iterates over all vertices, checks if the vertex is not yet included in the shortest path tree (sptSet[v] == 0) and if its distance is smaller than the current minimum. Returns the index of that vertex.

b) printSolution(int dist[])

Purpose: Prints the computed shortest distances from the source vertex to all other vertices.

Logic: Iterates over the distance array and displays the result in a readable format.

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

Purpose: Core implementation of Dijkstra's algorithm.

Logic:

Initializes a dist[] array with INT_MAX values except for the source vertex, which is set to 0.

Maintains a sptSet[] array to mark vertices included in the shortest path tree.

Repeats V-1 times:

Selects the vertex u with the minimum distance using minDistance.

Marks u as processed.

Updates the distance values of adjacent vertices if:

They are not yet processed.

An edge exists from u to v.

The total distance from the source to v through u is less than the current known distance.

Calls printSolution to display the result.

main() Function

Defines the graph as a 2D array (adjacency matrix).

Calls the dijkstra function with the graph and the chosen source vertex.