EXERCISE-23

AIM: To write a C program to find the shortest path from a given source vertex to all other vertices in a graph using Dijkstra’s

Algorithm:

1. Start.
2. Input the number of vertices and the cost (weight) adjacency matrix of the graph.
3. Input the source vertex.
4. Initialize:
   - distance[] array to store shortest distance (set to infinity for all, 0 for source).
   - visited[] array to mark visited vertices.
5. Repeat for (n - 1) vertices:
   - Select the unvisited vertex with the smallest distance.
   - Mark it as visited.
   - Update the distance of all its adjacent unvisited vertices.
6. After all vertices are visited, display the shortest distances.
7. End.

Program Code:

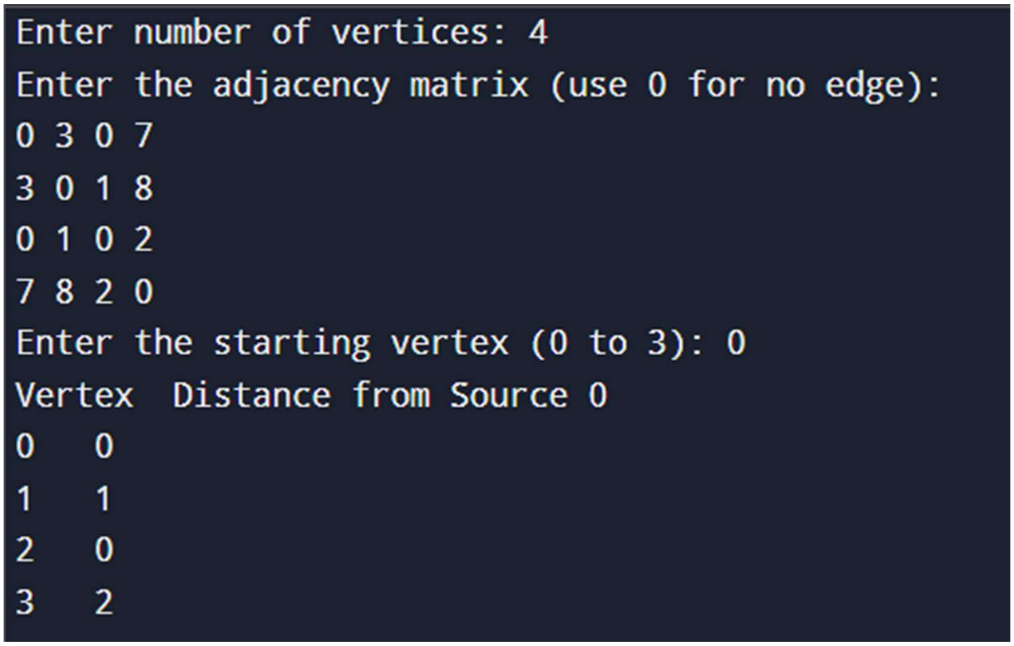
```
#include <stdio.h>

#define INFINITY 9999

#define MAX 20
```

```
void dijkstra(int graph[MAX][MAX], int n, int start) {
    int distance[MAX], visited[MAX] = {0};
    int i, j, count, minDistance, nextNode;
    for (i = 0; i < n; i++) {
        distance[i] = graph[start][i];
    }
    visited[start] = 1;
    distance[start] = 0;
    for (count = 1; count < n; count++) {
        minDistance = INFINITY;
        for (i = 0; i < n; i++) {
            if (!visited[i] && distance[i] < minDistance) {
                minDistance = distance[i];
                nextNode = i;
            }
        }
        visited[nextNode] = 1;
        for (i = 0; i < n; i++) {
            if (!visited[i] && graph[nextNode][i] != 0 &&
                distance[nextNode] + graph[nextNode][i] < distance[i]) {
                distance[i] = distance[nextNode] + graph[nextNode][i];
            }
        }
    }
    printf("Vertex\tDistance from Source %d\n", start);
    for (i = 0; i < n; i++) {
        printf("%d\t%d\n", i, distance[i]);
    }
}
int main() {
    int graph[MAX][MAX], n, start;
    printf("Enter number of vertices: ");
    scanf("%d", &n);
    printf("Enter the adjacency matrix (use 0 for no edge):\n");
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            scanf("%d", &graph[i][j]);
    printf("Enter the starting vertex (0 to %d): ", n - 1);
    scanf("%d", &start);
    dijkstra(graph, n, start);
    return 0;
}
```

Input and Output:



```
Enter number of vertices: 4
Enter the adjacency matrix (use 0 for no edge):
0 3 0 7
3 0 1 8
0 1 0 2
7 8 2 0
Enter the starting vertex (0 to 3): 0
Vertex  Distance from Source 0
0   0
1   1
2   0
3   2
```

Result:

The program correctly implements Dijkstra’s Algorithm and computes the shortest path from the source vertex to all other vertices in the graph.