5. Dijikstra algorithm

```
#include <stdio.h>
#include <limits.h>

#define MAX_VERTICES 20
#define INF 999

int main() {
    int i, j, vertices, source, u, min, currentVertex;
    int cost[MAX_VERTICES][MAX_VERTICES], visited[MAX_VERTICES], distance[MAX_VERTICES];

    printf("Enter the number of vertices: ");
    scanf("%d", &vertices);

    printf("Enter the adjacency matrix (Enter 999 for no edge):\n");
    for (i = 1; i <= vertices; i++) {
        for (j = 1; j <= vertices; j++) {
            scanf("%d", &cost[i][j]);
        }
    }

    printf("Enter the source vertex: ");
    scanf("%d", &source);

    for (i = 1; i <= vertices; i++) {
        visited[i] = 0;
        distance[i] = cost[source][i];
    }

    distance[source] = 0;
    visited[source] = 1;

    for (currentVertex = 2; currentVertex <= vertices; currentVertex++) {
        min = INF;

        for (i = 1; i <= vertices; i++) {
            if (visited[i] == 0 && distance[i] < min) {
                min = distance[i];
                u = i;
            }
        }
        visited[u] = 1;

        for (i = 1; i <= vertices; i++) {
            if (visited[i] == 0) {
                if (distance[i] > distance[u] + cost[u][i]) {
                    distance[i] = distance[u] + cost[u][i];
                }
            }
        }
    }

    printf("The shortest distances from vertex %d are:\n", source);
    for (i = 1; i <= vertices; i++) {
        printf("%d → %d = %d\n", source, i, distance[i]);
    }

    return 0;
}
```

```
Enter the number of vertices: 5
Enter the adjacency matrix (Enter 999 for no edge):
999 3 999 7 999
3 999 4 999 999
999 4 999 1 999
7 999 1 999 2
999 999 999 2 999
Enter the source vertex: 1

The shortest distances from vertex 1 are:
1 → 1 = 0
1 → 2 = 3
1 → 3 = 7
1 → 4 = 7
1 → 5 = 9
```