**Link State Routing Protocol**

#include <stdio.h>

#define INF 9999

#define MAX 10

void dijkstra(int graph[MAX][MAX], int n, int src) {

int dist[MAX], visited[MAX], i, j, min, u;

for (i = 0; i < n; i++) {

dist[i] = INF;

visited[i] = 0;

}

dist[src] = 0;

for (i = 0; i < n - 1; i++) {

min = INF;

for (j = 0; j < n; j++) {

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

min = dist[j];

u = j;

}

}

visited[u] = 1;

for (j = 0; j < n; j++) {

if (!visited[j] && graph[u][j] && dist[u] + graph[u][j] < dist[j]) {

dist[j] = dist[u] + graph[u][j];

}

}

}

printf("\nShortest paths from Router %d:\n", src);

for (i = 0; i < n; i++)

printf("Router %d -> Distance: %d\n", i, dist[i]);

}

int main() {

int n, graph[MAX][MAX], src, i, j;

printf("Enter number of routers: ");

scanf("%d", &n);

printf("Enter the adjacency matrix:\n");

for (i = 0; i < n; i++)

for (j = 0; j < n; j++)

scanf("%d", &graph[i][j]);

printf("Enter source router: ");

scanf("%d", &src);

dijkstra(graph, n, src);

return 0;

}