**CODE:**

#include <stdio.h>

#define INF 9999

void dijkstra(int n, int matrix[n][n], int source)

{

int dist[n];

int visited[n];

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

{

dist[i] = INF;

visited[i] = 0;

}

dist[source] = 0;

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

{

int minDist = INF;

int u;

for (int v = 0; v < n; v++)

{

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

{

minDist = dist[v];

u = v;

}

}

visited[u] = 1;

for (int v = 0; v < n; v++)

{

if (!visited[v] && matrix[u][v] != INF && dist[u] + matrix[u][v] < dist[v])

{

dist[v] = dist[u] + matrix[u][v];

}

}

}

printf("\nShortest distances from node %d to all other nodes:\n", source + 1);

printf("NODE DISTANCE\n");

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

{

printf("%d\t\t%d\n", i + 1, dist[i]);

}

}

int main()

{

int n;

printf("Enter the number of nodes: ");

scanf("%d", &n);

int matrix[n][n];

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

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

{

printf("Enter the distance for node %d:\n", i + 1);

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

{

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

if (i != j && matrix[i][j] == 0)

{

matrix[i][j] = INF;

}

}

}

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

{

printf("The link state packets for node %d:\n", i + 1);

printf("NODE DISTANCE\n");

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

{

printf("%d\t\t%d\n", j + 1, matrix[i][j]);

}

}

int source;

printf("Enter the source node for Dijkstra's algorithm: ");

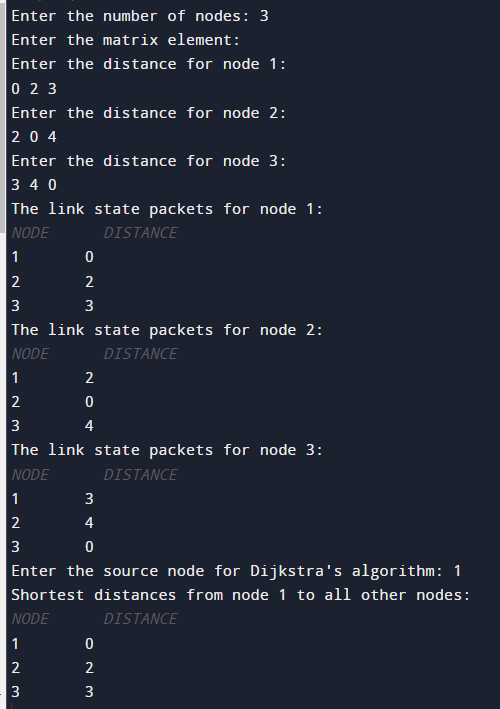
scanf("%d", &source);

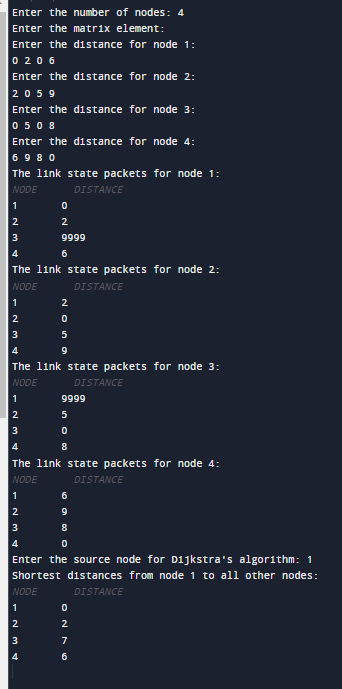
dijkstra(n, matrix, source - 1);

return 0;

}

**OUTPUT:**

****

****