Aim: write a program in c to find MST using Prim’s algorithm

The algorithm for Prim's algorithm is as follows:

1. Create a distance array to store the minimum distance from a node to the source node and initialize it with a large value except for the source node which should be 0.
2. Create a selected array to keep track of the nodes that have been selected for the minimum spanning tree and initialize it with false.
3. For n-1 times, do the following: a. Select a node u that has not been selected and has the minimum distance from the source node. b. Mark u as selected. c. For each node v that is not selected, if the distance from u to v is smaller than the current distance stored in the distance array, update the distance array with the new distance and store the node u as the previous node in the from array.
4. After completing the above steps, print the edges and their costs in the minimum spanning tree.

Source Code:

#include <stdio.h>

#include <stdlib.h>

#include <limits.h>

#define MAX 100

*int* n, cost[MAX][MAX];

*void* prim(*int* source) {

*int* i, j, u, v, min\_distance, distance[MAX], from[MAX];

*bool* selected[MAX];

    for (i = 1; i <= n; i++) {

        distance[i] = INT\_MAX;

        selected[i] = *false*;

    }

    distance[source] = 0;

    from[source] = -1;

    for (i = 1; i < n; i++) {

        min\_distance = INT\_MAX;

        for (j = 1; j <= n; j++) {

            if (!selected[j] && distance[j] < min\_distance) {

                min\_distance = distance[j];

                u = j;

            }

        }

        selected[u] = *true*;

        for (v = 1; v <= n; v++) {

            if (!selected[v] && cost[u][v] && distance[v] > cost[u][v]) {

                distance[v] = cost[u][v];

                from[v] = u;

            }

        }

    }

    printf("Edge\tCost\n");

    for (i = 2; i <= n; i++)

        printf("%d - %d\t%d\n", from[i], i, distance[i]);

}

*int* main() {

*int* i, j, source;

    printf("Enter number of nodes: ");

    scanf("%d", &n);

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

    for (i = 1; i <= n; i++) {

        for (j = 1; j <= n; j++) {

            scanf("%d", &cost[i][j]);

        }

    }

    printf("Enter the source node: ");

    scanf("%d", &source);

    prim(source);

    return 0;

}

Output:

Enter number of nodes: 5

Enter the cost matrix:

0 9 75 0 0

9 0 95 19 42

75 95 0 51 66

0 19 51 0 31

0 42 66 31 0

Enter the source node: 1

Edge Cost

1 - 2 9

4 - 3 51

2 - 4 19

4 - 5 31