# Prims algorithm

```c
#include <stdio.h>

int graph[10][10], totalNodes, mstEdges[10][2], totalCost;

void findMST(int graph[10][10], int totalNodes);

int main() {
    int i, j;

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

    printf("Enter the cost adjacency matrix:\n");
    for (i = 0; i < totalNodes; i++) {
        for (j = 0; j < totalNodes; j++) {
            scanf("%d", &graph[i][j]);
        }
    }

    findMST(graph, totalNodes);

    printf("Edges of the minimal spanning tree:\n");
    for (i = 0; i < totalNodes - 1; i++) {
        printf("(%d, %d) ", mstEdges[i][0], mstEdges[i][1]);
    }

    printf("\nSum of minimal spanning tree: %d\n", totalCost);

    return 0;
}

void findMST(int graph[10][10], int totalNodes) {
    int visited[10] = {0}, dist[10], parent[10];
    int i, j, node, nextNode, edgeWeight, edgeCount = 0;

    for (i = 0; i < totalNodes; i++) {
        dist[i] = graph[0][i];
        parent[i] = 0;
    }

    visited[0] = 1;
```

```
    totalCost = 0;

    for (i = 1; i < totalNodes; i++) {
        edgeWeight = 999;
        nextNode = -1;

        for (j = 0; j < totalNodes; j++) {
            if (!visited[j] && dist[j] < edgeWeight) {
                edgeWeight = dist[j];
                nextNode = j;
            }
        }

        if (nextNode != -1) {
            mstEdges[edgeCount][0] = nextNode;
            mstEdges[edgeCount][1] = parent[nextNode];
            edgeCount++;
            totalCost += graph[nextNode][parent[nextNode]];
            visited[nextNode] = 1;

            for (j = 0; j < totalNodes; j++) {
                if (!visited[j] && graph[nextNode][j] < dist[j]) {
                    dist[j] = graph[nextNode][j];
                    parent[j] = nextNode;
                }
            }
        }
    }
}
```

```
Output
Enter the number of vertices: 4
Enter the cost adjacency matrix:
0 5 8 0
5 0 10 15
8 10 0 20
0 15 20 0

Edges of the minimal spanning tree:
(1, 0) (2, 0) (3, 1)
Sum of minimal spanning tree: 30
```