```
#include <stdio.h>
#define MAX_NODES 100
#define INT_MAX 100
int findMinKey(int key[], int mstSet[], int nodes) {
    int min = INT_MAX, min_index;
    for (int v = 0; v < nodes; v++) {
        if (mstSet[v] == 0 && key[v] < min) {
            min = key[v];
            min_index = v;
        }
    }
    return min_index;
}
void printMST(int parent[], int graph[MAX_NODES][MAX_NODES], int nodes) {
    printf("Edge \tWeight\n");
    for (int i = 1; i < nodes; i++) {
        printf("%d - %d \t%d\n", parent[i], i, graph[i][parent[i]]);
    }
}
void primMST(int graph[MAX_NODES][MAX_NODES], int nodes) {
    int parent[MAX_NODES];
    int key[MAX_NODES];
    int mstSet[MAX_NODES];
    for (int i = 0; i < nodes; i++) {
        key[i] = INT_MAX;
        mstSet[i] = 0;
    }
    key[0] = 0;
    parent[0] = -1;
    for (int count = 0; count < nodes - 1; count++) {
        int u = findMinKey(key, mstSet, nodes);
        mstSet[u] = 1;
```

```
        for (int v = 0; v < nodes; v++) {
            if (graph[u][v] && mstSet[v] == 0 && graph[u][v] < key[v]) {
                parent[v] = u;
                key[v] = graph[u][v];     }
        }   }
    printMST(parent, graph, nodes); }
int main() {
    int nodes;
    printf("Enter the number of nodes in the graph: ");
    scanf("%d", &nodes);
    int graph[MAX_NODES][MAX_NODES];
    printf("Enter the adjacency matrix of the graph:\n");
    for (int i = 0; i < nodes; i++) {
        for (int j = 0; j < nodes; j++) {
            scanf("%d", &graph[i][j]);
        }   }
  primMST(graph, nodes);
    return 0;
}
```