```
#include <stdio.h>
#include <limits.h>
#define SIZE 20
#define INF 9999
int minKey(int key[], int mstSet[], int n) {
    int min = INF, minIndex = -1;
    for (int v = 0; v < n; v++)
        if (!mstSet[v] && key[v] < min) {
            min = key[v];
            minIndex = v;
        }
    return minIndex;
}
void printMST(int parent[], int graph[SIZE][SIZE], int n) {
    int totalWeight = 0;
    printf("Edge \tWeight\n");
    for (int i = 1; i < n; i++) {
        printf("%d - %d \t%d\n", parent[i], i, graph[i][parent[i]]);
        totalWeight += graph[i][parent[i]];
    }
    printf("Total weight of MST: %d\n", totalWeight);
}
void primMST(int graph[SIZE][SIZE], int n) {
    int parent[SIZE];
    int key[SIZE];
    int mstSet[SIZE];
     for (int i = 0; i < n; i++) {
        key[i] = INF;
        mstSet[i] = 0;
    }
    key[0] = 0;
    parent[0] = -1;
    for (int count = 0; count < n - 1; count++) {
        int u = minKey(key, mstSet, n);
        mstSet[u] = 1;
        for (int v = 0; v < n; v++) {
            if (graph[u][v] && !mstSet[v] && graph[u][v] < key[v]) {
                parent[v] = u;
                key[v] = graph[u][v];
            }
        }
    }
    printMST(parent, graph, n);
}
```

```
int main() {
    int graph[SIZE][SIZE], n;
    printf("Enter the number of vertices: ");
    scanf("%d", &n);
    printf("Enter the adjacency matrix (0 if no edge):\n");
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            scanf("%d", &graph[i][j]);
    primMST(graph, n);
    return 0;
}
```