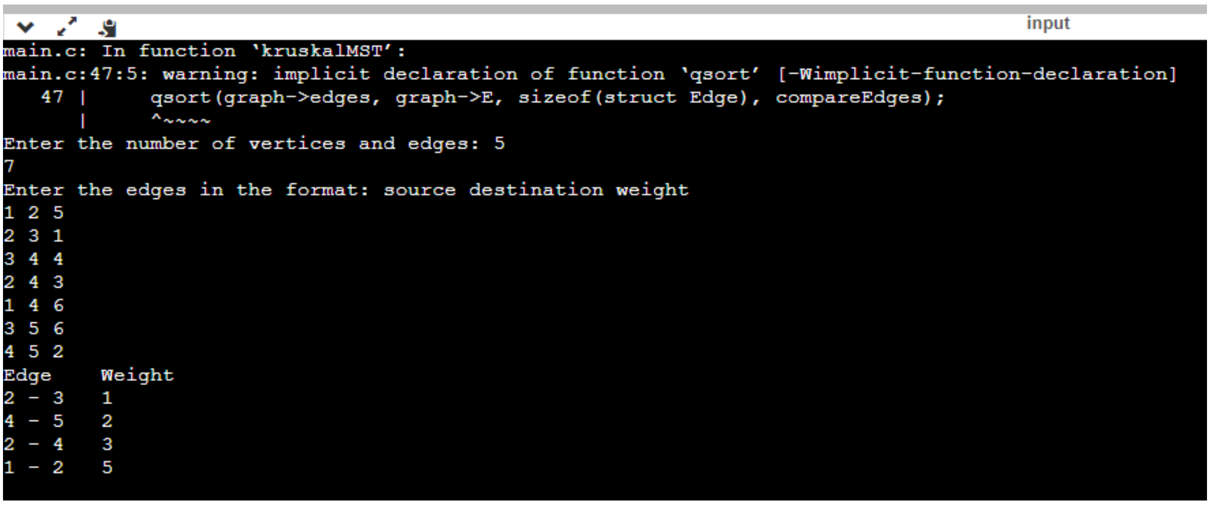
```
#include <stdio.h>

#include <stdbool.h>

#define MAX_VERTICES 100

struct Edge {

    int src, dest, weight;

};

struct Graph {

    int V, E;

    struct Edge edges[MAX_VERTICES];

};

void addEdge(struct Graph* graph, int src, int dest, int weight) {

    graph->edges[graph->E].src = src;

    graph->edges[graph->E].dest = dest;

    graph->edges[graph->E].weight = weight;

    graph->E++;

}

void unionSet(int parent[], int x, int y) {

    int xRoot = x;

    int yRoot = y;

    parent[xRoot] = yRoot;

}

int find(int parent[], int i) {

    if (parent[i] == -1)

        return i;

    return find(parent, parent[i]);

}

int compareEdges(const void* a, const void* b) {

    struct Edge* edgeA = (struct Edge*)a;
    struct Edge* edgeB = (struct Edge*)b;
    return edgeA->weight - edgeB->weight;
}
void kruskalMST(struct Graph* graph) {
    struct Edge result[MAX_VERTICES];
    int parent[MAX_VERTICES];
    int e = 0, i = 0;
    for (int v = 0; v < graph->V; v++) {
        parent[v] = -1;
    }
    qsort(graph->edges, graph->E, sizeof(struct Edge), compareEdges);
    while (e < graph->V - 1 && i < graph->E) {
        struct Edge next_edge = graph->edges[i++];
        int x = find(parent, next_edge.src);
       int y = find(parent, next_edge.dest);
        if (x != y) {
            result[e++] = next_edge;
            unionSet(parent, x, y);
        }
    }
    printf("Edge \tWeight\n");
    for (int j = 0; j < e; j++) {
        printf("%d - %d \t%d\n", result[j].src, result[j].dest, result[j].weight);
    }
}
int main() {
    struct Graph graph;
    int V, E;
    printf("Enter the number of vertices and edges: ");
    scanf("%d %d", &V, &E);
    graph.V = V;
    graph.E = 0;

    printf("Enter the edges in the format: source destination weight\n");

    for (int i = 0; i < E; i++) {

        int src, dest, weight;

        scanf("%d %d %d", &src, &dest, &weight);

        addEdge(&graph, src, dest, weight);

    }

    kruskalMST(&graph);

    return 0;

}
```



```
main.c: In function ‘kruskalMST’:
main.c:47:5: warning: implicit declaration of function ‘qsort’ [-Wimplicit-function-declaration]
   47 |     qsort(graph->edges, graph->E, sizeof(struct Edge), compareEdges);
      |     ^~~~~
Enter the number of vertices and edges: 5
7
Enter the edges in the format: source destination weight
1 2 5
2 3 1
3 4 4
2 4 3
1 4 6
3 5 6
4 5 2
Edge    Weight
2 - 3   1
4 - 5   2
2 - 4   3
1 - 2   5
```