25. Implementation of Minimum Spanning Tree using Kruskal Algorithm

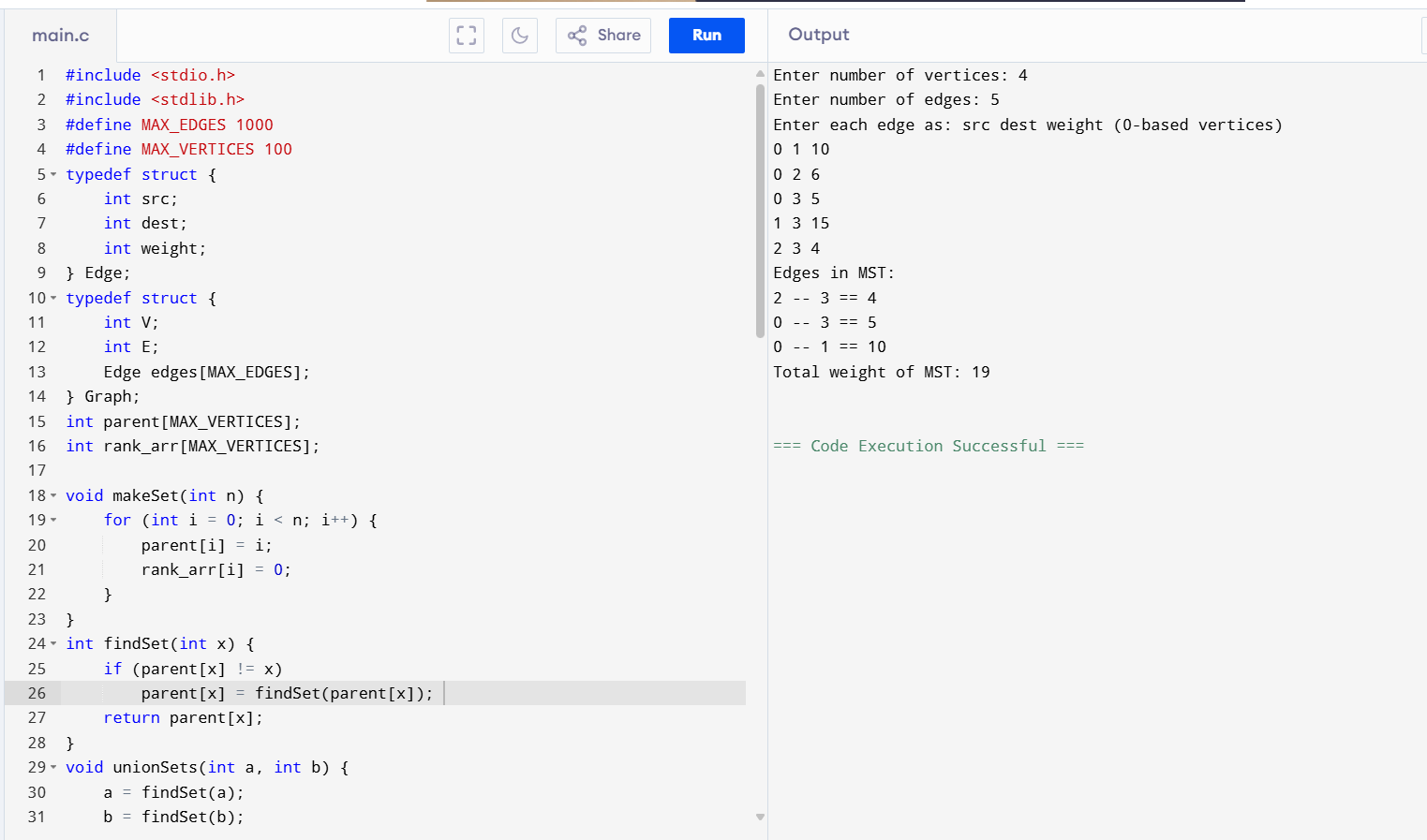
**Aim:** To write a C program that, given an undirected, connected, weighted graph, finds its Minimum Spanning Tree (MST) using **Kruskal’s Algorithm**. The program should:

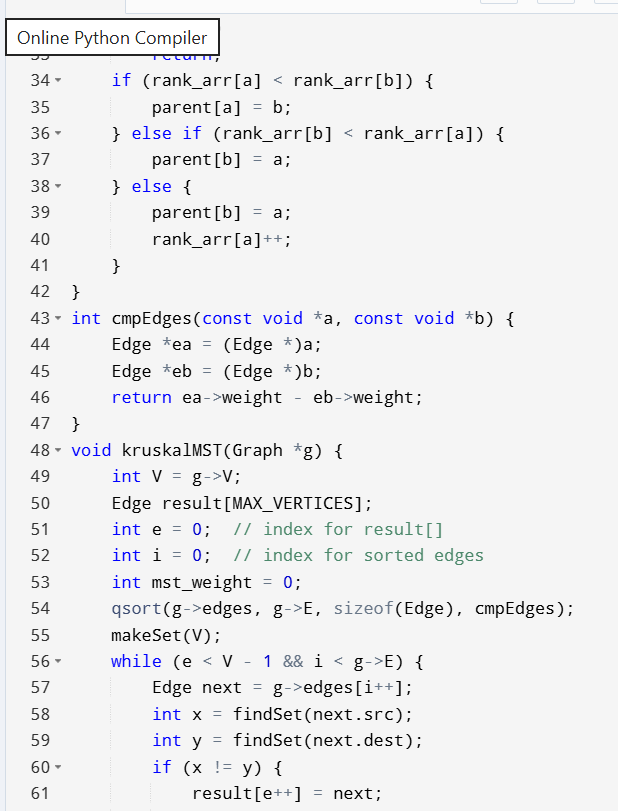
* Sort all edges by weight (non-decreasing order).
* Use a Union-Find (Disjoint Set) structure to detect cycles.
* Pick edges one by one, adding an edge if it doesn’t form a cycle.
* Stop when the MST has (V-1) edges, where V is number of vertices.

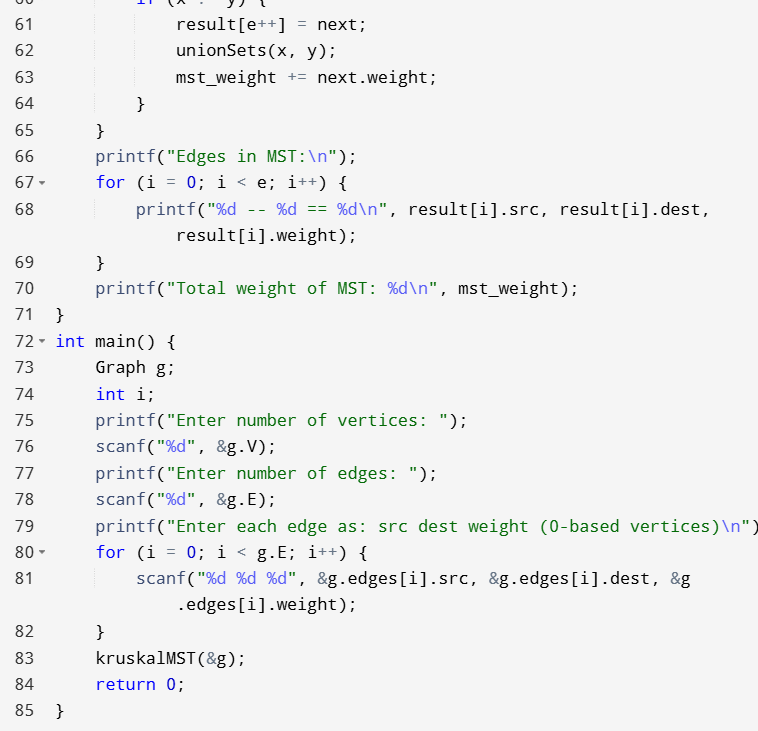
**Algorithm:**

1. Read number of vertices V and number of edges E.
2. Store all edges in an array of edges: each edge stores (src, dest, weight).
3. Sort the edges array by increasing weight.
4. Initialize Union-Find structure: each vertex is in its own set.
   * parent[i] = i for all i
   * Optionally maintain rank[i] for union by rank.
5. Initialize MST edge count count = 0 and total weight sum mst\_weight = 0.
6. For each edge in sorted list:
   * Find the root (set) of its src and dest vertices.
   * If they are in different sets, include this edge in MST:
     + Print/store the edge,
     + Union their sets,
     + Increment count, add its weight to mst\_weight.
   * If they are in the same set, skip (would form cycle).
   * If count becomes V-1, break.
7. Output the MST edges, and the total weight of the MST.

**Program:**

****

****

****

**Result:** The program has been executed successfully and the output has beed verified