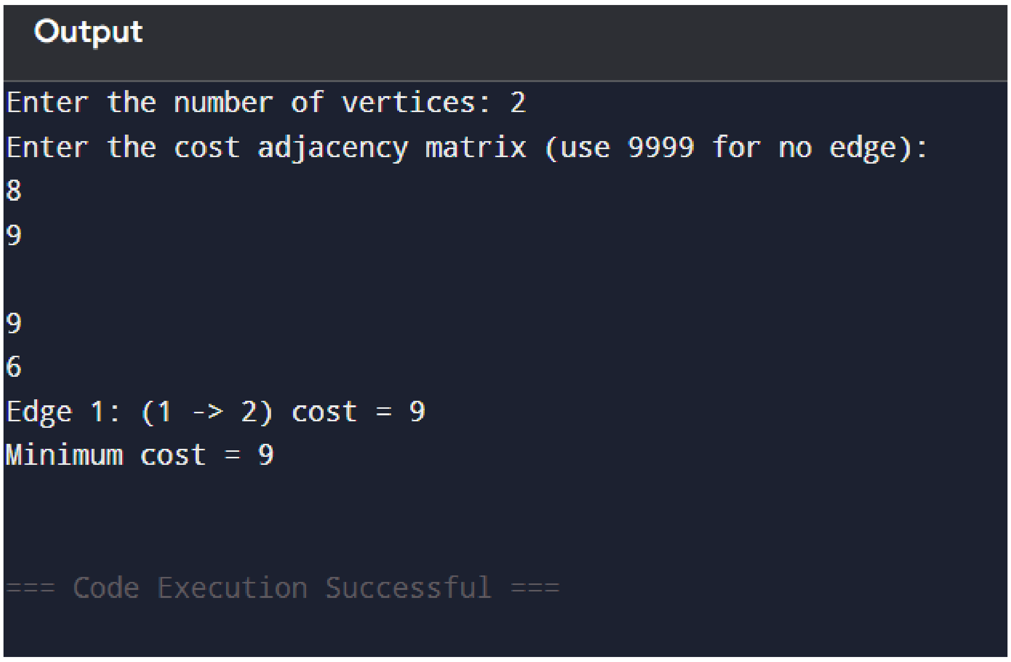
Find Minimum Cost Spanning Tree of a given undirected graph using

Kruskal’s algorithm.

```
#include <stdio.h>
#define MAX 100
#define INF 9999

int parent[MAX];

int find(int i) {
   while (parent[i])
      i = parent[i];
   return i;
}

int union_set(int i, int j) {
   if (i != j) {
      parent[j] = i;
      return 1;
   }
   return 0;
}

int main() {
   int cost[MAX][MAX], n, i, j, u, v, min, a, b;
   int ne = 1, total_cost = 0;

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

```
printf("Enter the cost adjacency matrix (use %d for no edge):\n", INF);
for (i = 1; i <= n; i++)
  for (j = 1; j <= n; j++)
    scanf("%d", &cost[i][j]);

while (ne < n) {
  min = INF;
  for (i = 1; i <= n; i++) {
    for (j = 1; j <= n; j++) {
      if (cost[i][j] < min) {
        min = cost[i][j];
        a = u = i;
        b = v = j;
      }
    }
  }

  u = find(u);
  v = find(v);

  if (union_set(u, v)) {
    printf("Edge %d: (%d -> %d) cost = %d\n", ne++, a, b, min);
    total_cost += min;
  }

  cost[a][b] = cost[b][a] = INF;
}

printf("Minimum cost = %d\n", total_cost);

return 0;
```

```
}
```



```
Enter the number of vertices: 2
Enter the cost adjacency matrix (use 9999 for no edge):
8
9

9
6
Edge 1: (1 -> 2) cost = 9
Minimum cost = 9



=== Code Execution Successful ===
```