1. **Implementation of Minimum Spanning Tree using Prim’s Algorithm.**

**PROGRAM:**

#include <stdio.h>

#include <limits.h>

#define MAX 100

int minKey(int key[], int mstSet[], int V) {

int min = INT\_MAX, min\_index;

for (int v = 0; v < V; v++)

if (mstSet[v] == 0 && key[v] < min)

min = key[v], min\_index = v;

return min\_index;

}

void printMST(int parent[], int graph[MAX][MAX], int V) {

printf("Edge \tWeight\n");

for (int i = 1; i < V; i++)

printf("%d - %d \t%d \n", parent[i], i, graph[i][parent[i]]);

}

void primMST(int graph[MAX][MAX], int V) {

int parent[V];

int key[V];

int mstSet[V];

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

key[i] = INT\_MAX;

mstSet[i] = 0;

}

key[0] = 0;

parent[0] = -1;

for (int count = 0; count < V - 1; count++) {

int u = minKey(key, mstSet, V);

mstSet[u] = 1;

for (int v = 0; v < V; v++)

if (graph[u][v] && mstSet[v] == 0 && graph[u][v] < key[v])

parent[v] = u, key[v] = graph[u][v];

}

printMST(parent, graph, V);

}

int main() {

int graph[MAX][MAX], V;

printf("Enter the number of vertices: ");

scanf("%d", &V);

printf("Enter the adjacency matrix:\n");

for (int i = 0; i < V; i++)

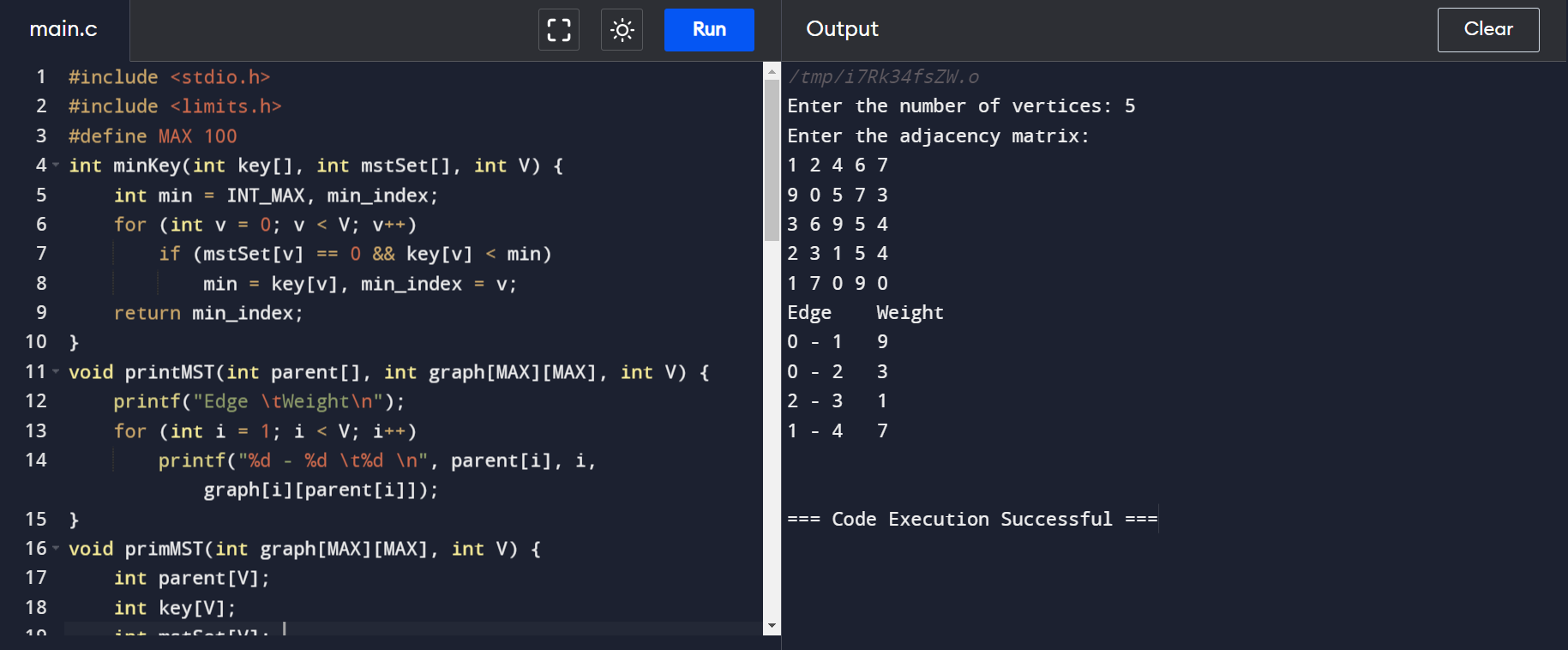
for (int j = 0; j < V; j++)

scanf("%d", &graph[i][j]);

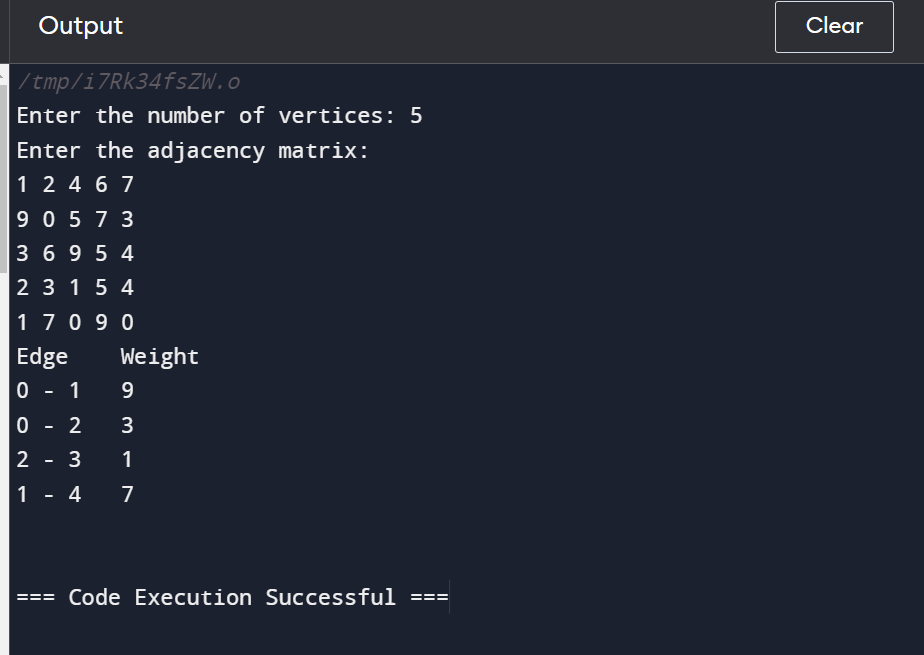
primMST(graph, V);

return 0;}

**INPUT:**

****

**OUTPUT:**

****