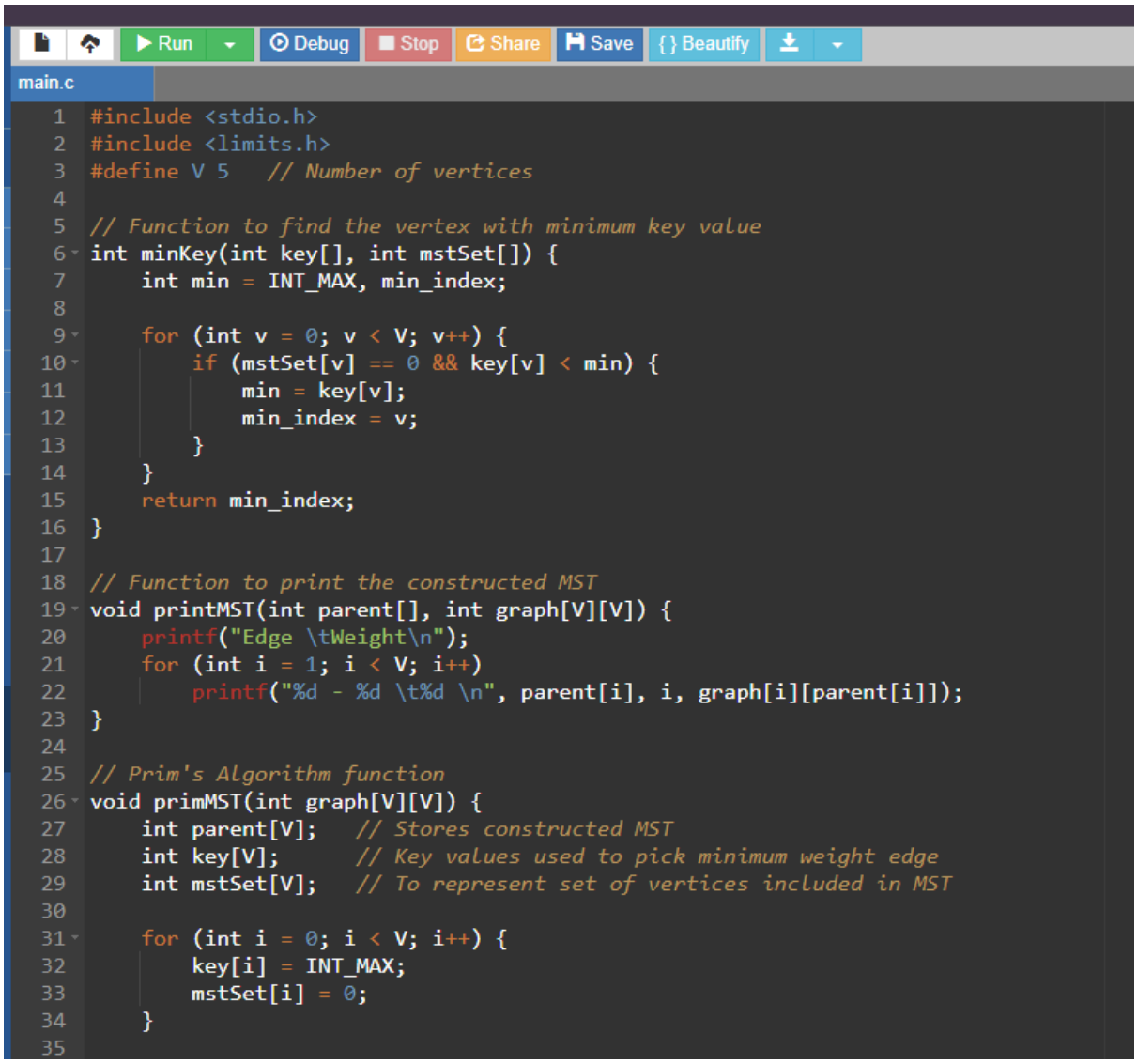
Prims algorithm:

Code:




```c
#include <stdio.h>
#include <limits.h>
#define V 5   // Number of vertices

// Function to find the vertex with minimum key value
int minKey(int key[], int mstSet[]) {
    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;
}

// Function to print the constructed MST
void printMST(int parent[], int graph[V][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]]);
}

// Prim's Algorithm function
void primMST(int graph[V][V]) {
    int parent[V];   // Stores constructed MST
    int key[V];      // Key values used to pick minimum weight edge
    int mstSet[V];   // To represent set of vertices included in MST

    for (int i = 0; i < V; i++) {
        key[i] = INT_MAX;
        mstSet[i] = 0;
    }

```

Run | Debug | Stop | Share | Save | { } Beautify

main.c

```
    }

    key[0] = 0;     // Start from first vertex
    parent[0] = -1; // First node is root of MST

    for (int count = 0; count < V - 1; count++) {
        int u = minKey(key, mstSet);
        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);
}

// Driver code
int main() {
    int graph[V][V] = {
        {0, 2, 0, 6, 0},
        {2, 0, 3, 8, 5},
        {0, 3, 0, 0, 7},
        {6, 8, 0, 0, 9},
        {0, 5, 7, 9, 0}
    };

    primMST(graph);

    return 0;
}
```

Output:

```
Edge    Weight
0 - 1   2
1 - 2   3
0 - 3   6
1 - 4   5


...Program finished with exit code 0
Press ENTER to exit console.
```