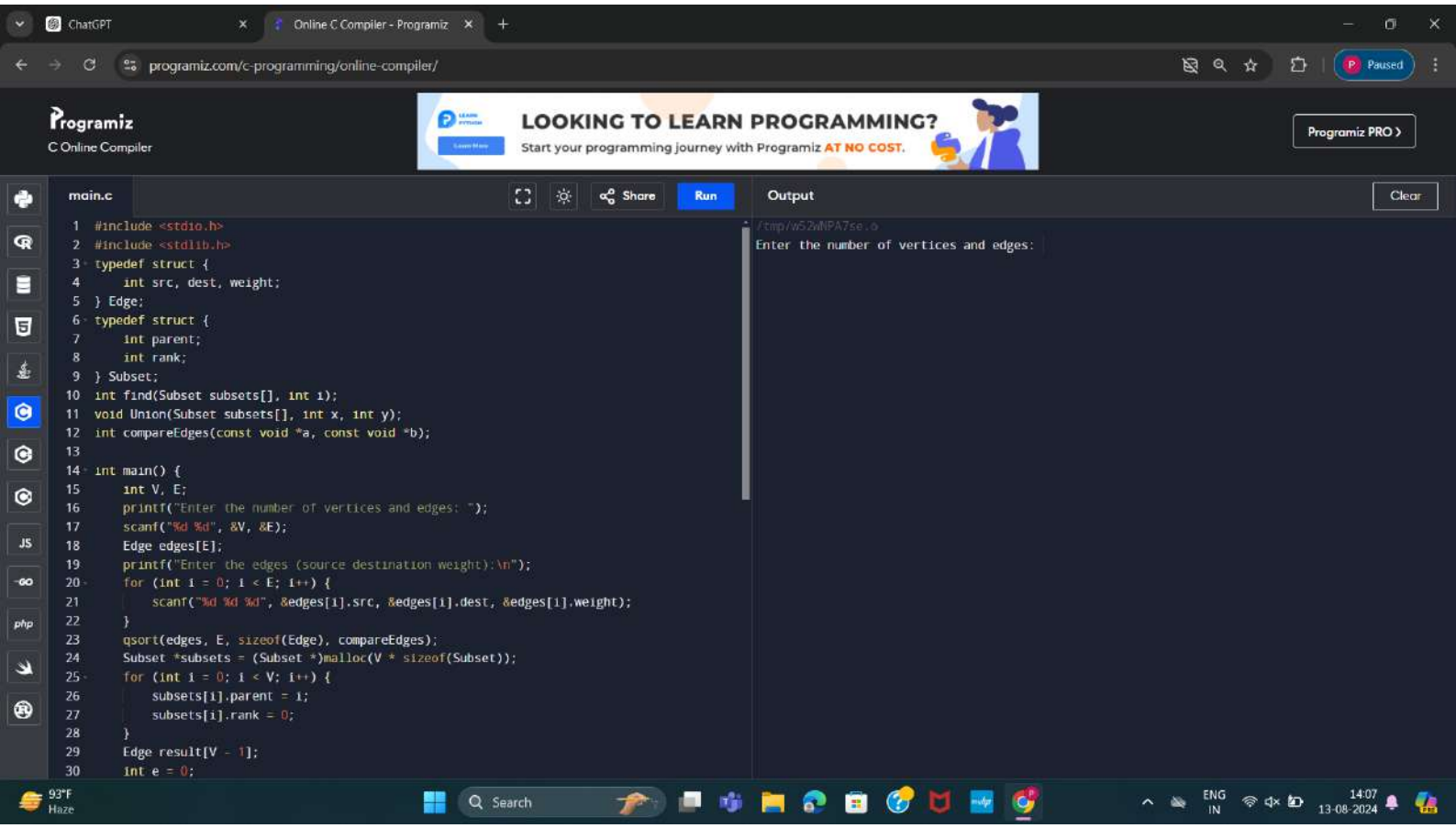

ChatGPT
Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
Paused
Programiz
C Online Compiler
LOOKING TO LEARN PROGRAMMING?
Start your programming journey with Programiz AT NO COST.
Programiz PRO
main.c
Share
Run
Output
Clear
#include <stdio.h>
#include <stdlib.h>
typedef struct {
int src, dest, weight;
} Edge;
typedef struct {
int parent;
int rank;
} Subset;
int find(Subset subsets[], int i);
void Union(Subset subsets[], int x, int y);
int compareEdges(const void *a, const void *b);
int main() {
int V, E;
printf("Enter the number of vertices and edges: ");
scanf("%d %d", &V, &E);
Edge edges[E];
printf("Enter the edges (source destination weight):\n");
for (int i = 0; i < E; i++) {
scanf("%d %d %d", &edges[i].src, &edges[i].dest, &edges[i].weight);
}
qsort(edges, E, sizeof(Edge), compareEdges);
Subset *subsets = (Subset *)malloc(V * sizeof(Subset));
for (int i = 0; i < V; i++) {
subsets[i].parent = i;
subsets[i].rank = 0;
}
Edge result[V - 1];
int e = 0;
/tmp/w52WNPA7se.o
Enter the number of vertices and edges:
93°F
Haze
Search
ENG
IN
14:07
13-08-2024

Online C Compiler - Programiz

programiz.com/c-programming/online-compiler/

Programiz
C Online Compiler

Programiz PRO >

main.c

Share | Run | Output | Clear

```c
    }
    Edge result[V - 1];
    int e = 0;
    int i = 0;
    while (e < V - 1) {
        Edge next_edge = edges[i++];
        int x = find(subsets, next_edge.src);
        int y = find(subsets, next_edge.dest);
        if (x != y) {
            result[e++] = next_edge;
            Union(subsets, x, y);
        }
    }
    printf("Edges in the constructed MST.\n");
    for (int i = 0; i < e; i++) {
        printf("%d -- %d == %d\n", result[i].src, result[i].dest, result[i].weight);
    }
    free(subsets);
    return 0;
}
int find(Subset subsets[], int i) {
    if (subsets[i].parent != i) {
        subsets[i].parent = find(subsets, subsets[i].parent);
    }
    return subsets[i].parent;
}
void Union(Subset subsets[], int x, int y) {
    int xroot = find(subsets, x);
    int yroot = find(subsets, y);
    if (subsets[xroot].rank < subsets[yroot].rank) {
        subsets[xroot].parent = yroot;
    } else if (subsets[xroot].rank > subsets[yroot].rank) {
        subsets[yroot].parent = xroot;
    } else {
        subsets[yroot].parent = xroot;
        subsets[xroot].rank++;
    }
}
int compareEdges(const void *a, const void *b) {
    return ((Edge *)a)->weight - ((Edge *)b)->weight;
}
```

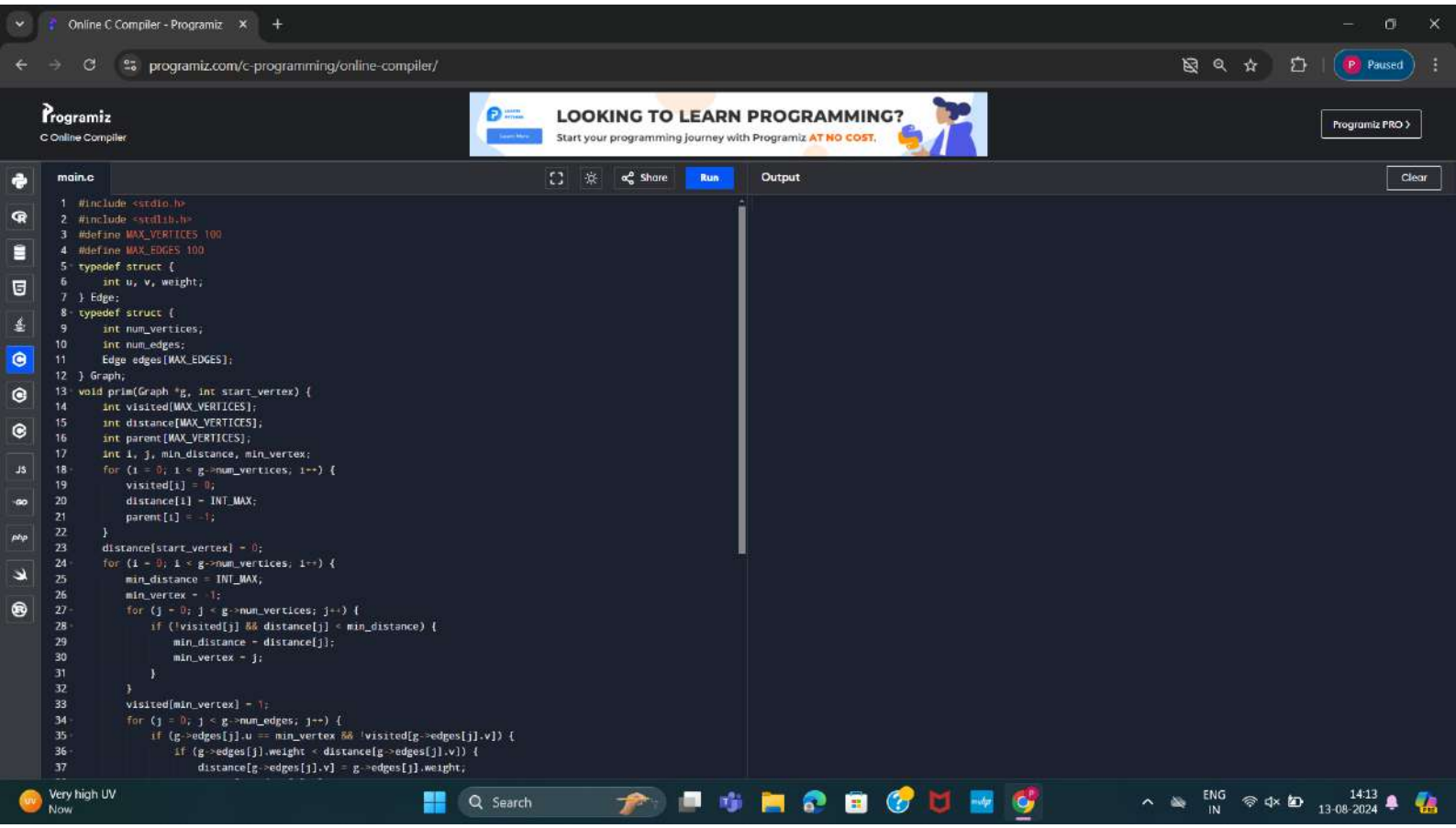
```
Enter the number of vertices and edges:
```

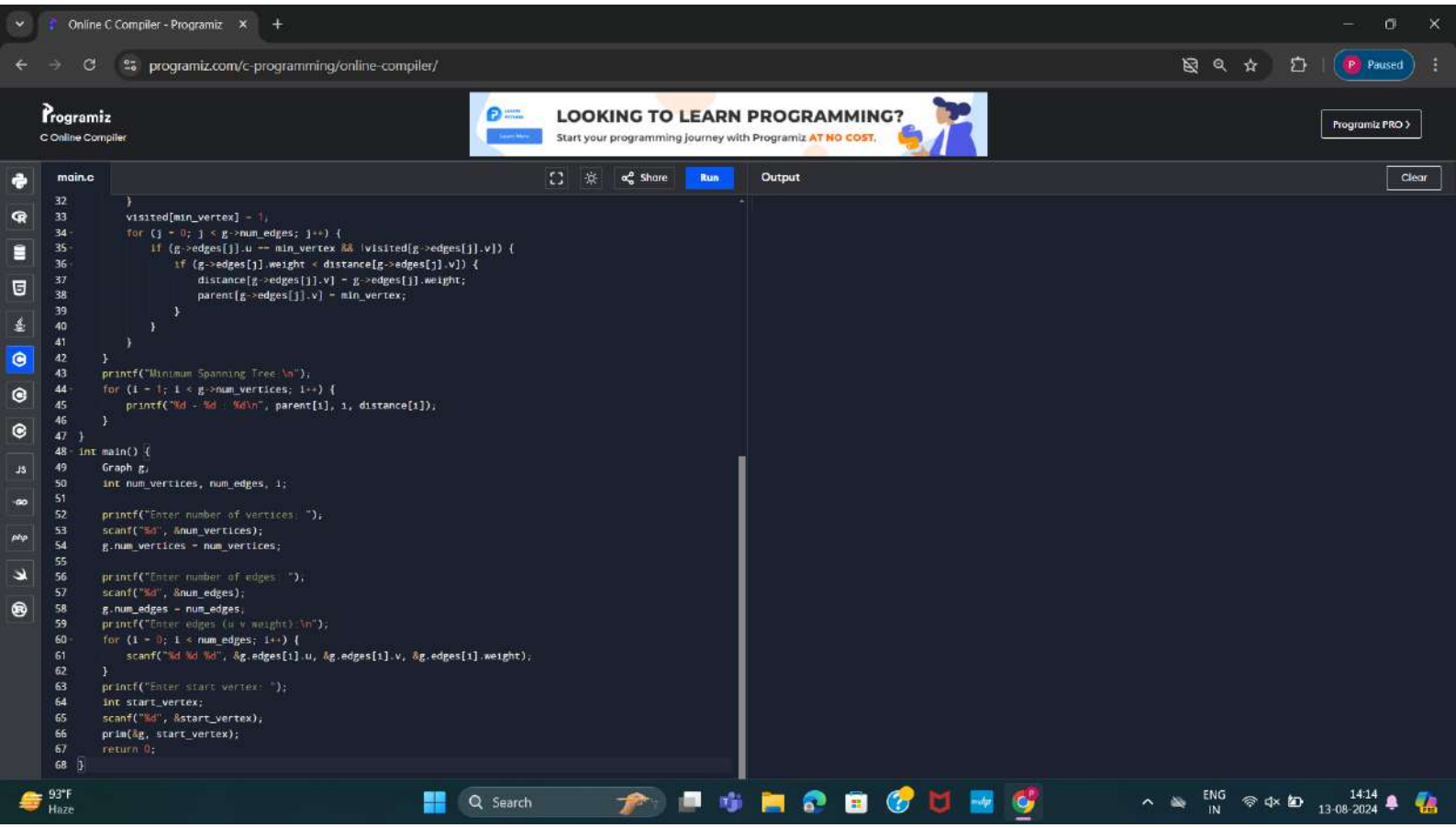
Hot weather
Now

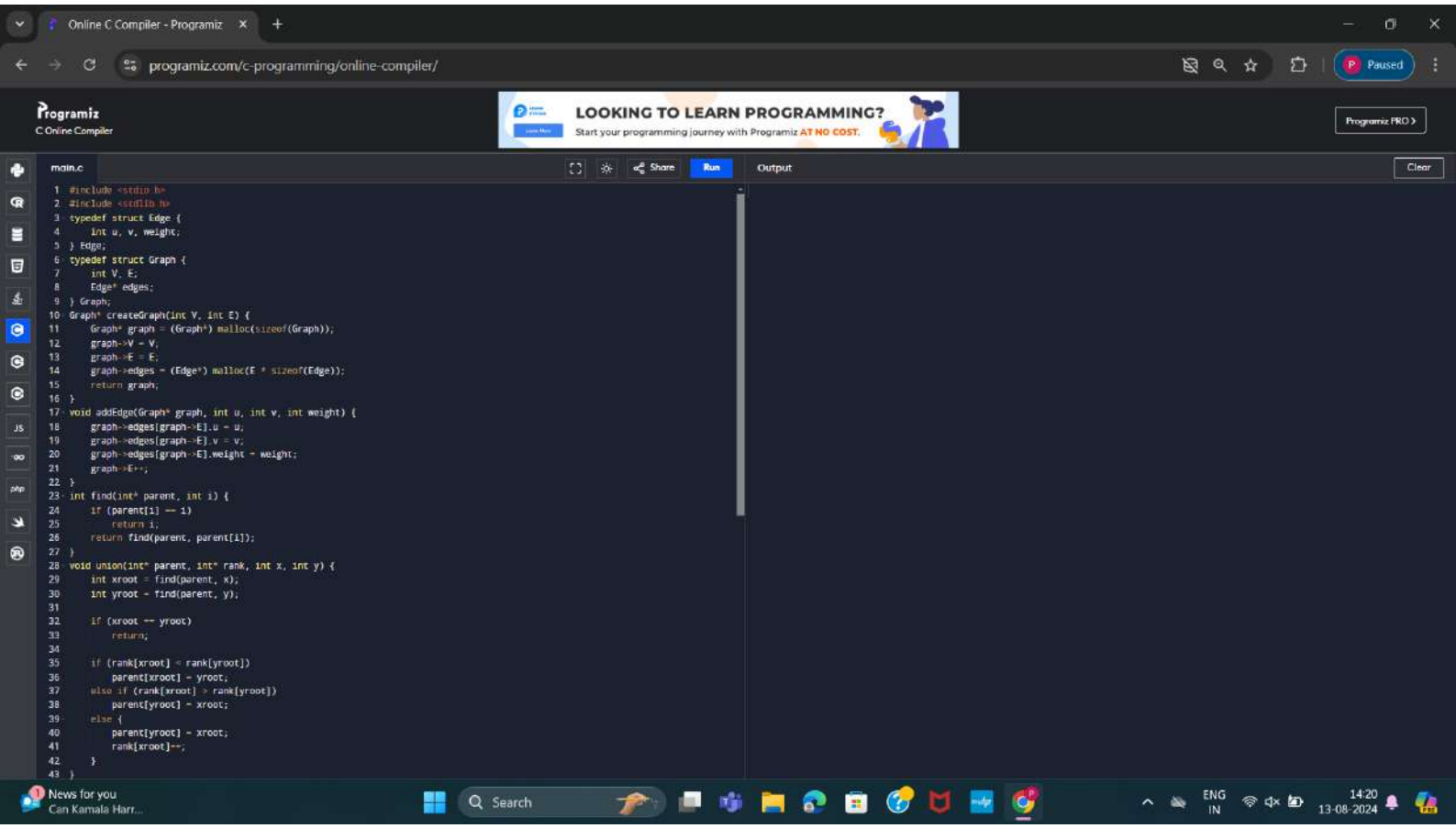
Search

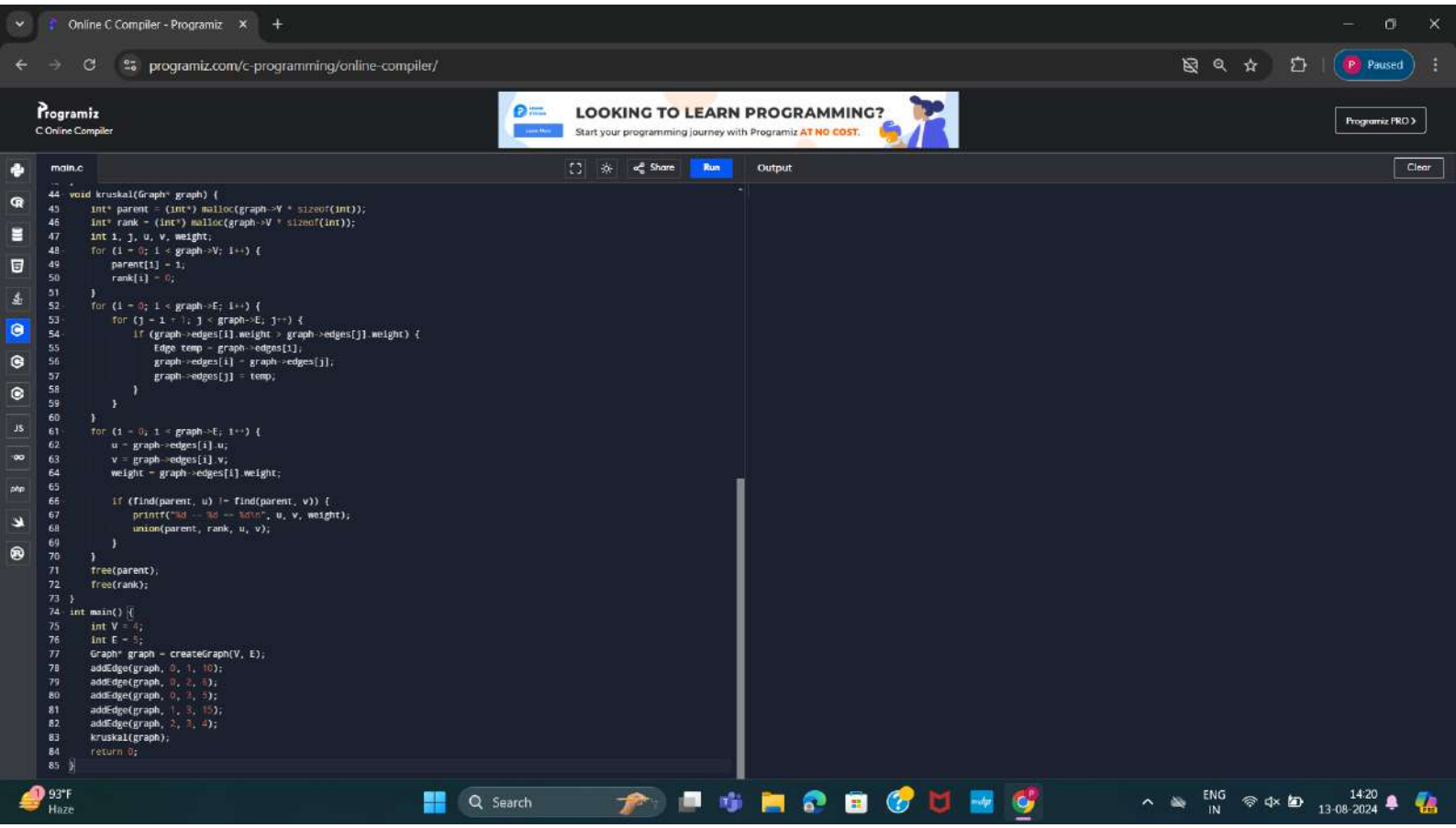
ENG
IN

14:09
13-08-2024


Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
Paused
Programiz
C Online Compiler
LOOKING TO LEARN PROGRAMMING?
Start your programming journey with Programiz AT NO COST.
Programiz PRO
main.c
Share
Run
Output
Clear
#include <stdio.h>
#include <stdlib.h>
#define MAX_VERTICES 100
#define MAX_EDGES 100
typedef struct {
int u, v, weight;
} Edge;
typedef struct {
int num_vertices;
int num_edges;
Edge edges[MAX_EDGES];
} Graph;
void prim(Graph *g, int start_vertex) {
int visited[MAX_VERTICES];
int distance[MAX_VERTICES];
int parent[MAX_VERTICES];
int i, j, min_distance, min_vertex;
for (i = 0; i < g->num_vertices; i++) {
visited[i] = 0;
distance[i] = INT_MAX;
parent[i] = -1;
}
distance[start_vertex] = 0;
for (i = 0; i < g->num_vertices; i++) {
min_distance = INT_MAX,
min_vertex = -1;
for (j = 0; j < g->num_vertices; j++) {
if (!visited[j] && distance[j] < min_distance) {
min_distance = distance[j];
min_vertex = j;
}
}
visited[min_vertex] = 1;
for (j = 0; j < g->num_edges; j++) {
if (g->edges[j].u == min_vertex && !visited[g->edges[j].v]) {
if (g->edges[j].weight < distance[g->edges[j].v]) {
distance[g->edges[j].v] = g->edges[j].weight;
Very high UV
Now
Search
ENG
IN
14:13
13-08-2024


Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
Paused
Programiz
C Online Compiler
LOOKING TO LEARN PROGRAMMING?
Start your programming journey with Programiz AT NO COST.
Programiz PRO
main.c
Share
Run
Output
Clear
}
visited[min_vertex] = 1;
for (j = 0; j < g->num_edges; j++) {
if (g->edges[j].u == min_vertex && !visited[g->edges[j].v]) {
if (g->edges[j].weight < distance[g->edges[j].v]) {
distance[g->edges[j].v] = g->edges[j].weight;
parent[g->edges[j].v] = min_vertex;
}
}
}
}
printf("Minimum Spanning Tree:\n");
for (i = 1; i < g->num_vertices; i++) {
printf("%d - %d : %d\n", parent[i], i, distance[i]);
}
}
int main() {
Graph g;
int num_vertices, num_edges, i;
printf("Enter number of vertices: ");
scanf("%d", &num_vertices);
g.num_vertices = num_vertices;
printf("Enter number of edges: ");
scanf("%d", &num_edges);
g.num_edges = num_edges;
printf("Enter edges (u v weight):\n");
for (i = 0; i < num_edges; i++) {
scanf("%d %d %d", &g.edges[i].u, &g.edges[i].v, &g.edges[i].weight);
}
printf("Enter start vertex: ");
int start_vertex;
scanf("%d", &start_vertex);
prim(&g, start_vertex);
return 0;
}
93°F
Haze
Search
ENG
IN
14:14
13-08-2024


Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
Paused
Programiz
C Online Compiler
LOOKING TO LEARN PROGRAMMING?
Start your programming journey with Programiz AT NO COST.
Programiz PRO
main.c
Share
Run
Output
Clear
#include <stdio.h>
#include <stdlib.h>
typedef struct Edge {
int u, v, weight;
} Edge;
typedef struct Graph {
int V, E;
Edge* edges;
} Graph;
Graph* createGraph(int V, int E) {
Graph* graph = (Graph*) malloc(sizeof(Graph));
graph->V = V;
graph->E = E;
graph->edges = (Edge*) malloc(E * sizeof(Edge));
return graph;
}
void addEdge(Graph* graph, int u, int v, int weight) {
graph->edges[graph->E].u = u;
graph->edges[graph->E].v = v;
graph->edges[graph->E].weight = weight;
graph->E++;
}
int find(int* parent, int i) {
if (parent[i] == i)
return i;
return find(parent, parent[i]);
}
void union(int* parent, int* rank, int x, int y) {
int xroot = find(parent, x);
int yroot = find(parent, y);
if (xroot == yroot)
return;
if (rank[xroot] < rank[yroot])
parent[xroot] = yroot;
else if (rank[xroot] > rank[yroot])
parent[yroot] = xroot;
else {
parent[yroot] = xroot;
rank[xroot]++;
}
}
News for you
Can Kamala Harr...
Search
ENG
IN
14:20
13-08-2024


Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
Paused
Programiz
C Online Compiler
LOOKING TO LEARN PROGRAMMING?
Start your programming journey with Programiz AT NO COST.
Programiz PRO
main.c
Share
Run
Output
Clear
void kruskal(Graph* graph) {
int* parent = (int*) malloc(graph->V * sizeof(int));
int* rank = (int*) malloc(graph->V * sizeof(int));
int i, j, u, v, weight;
for (i = 0; i < graph->V; i++) {
parent[i] = i;
rank[i] = 0;
}
for (i = 0; i < graph->E; i++) {
for (j = i + 1; j < graph->E; j++) {
if (graph->edges[i].weight > graph->edges[j].weight) {
Edge temp = graph->edges[i];
graph->edges[i] = graph->edges[j];
graph->edges[j] = temp;
}
}
}
for (i = 0; i < graph->E; i++) {
u = graph->edges[i].u;
v = graph->edges[i].v;
weight = graph->edges[i].weight;
if (find(parent, u) != find(parent, v)) {
printf("%d -- %d == %d\n", u, v, weight);
union(parent, rank, u, v);
}
}
free(parent);
free(rank);
}
int main() {
int V = 4;
int E = 5;
Graph* graph = createGraph(V, E);
addEdge(graph, 0, 1, 10);
addEdge(graph, 0, 2, 6);
addEdge(graph, 0, 3, 5);
addEdge(graph, 1, 3, 15);
addEdge(graph, 2, 3, 4);
kruskal(graph);
return 0;
}
93°F
Haze
Search
ENG
IN
14:20
13-08-2024