Minimum spanning tree using prim’s algorithm

**#include <stdio.h>**

**#include <limits.h>**

**#define V 5**

**Int minKey(int key[], int mstSet[]) {**

**Int min = INT\_MAX, min\_index;**

**Int v;**

**For (v = 0; v < V; v++)**

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

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

**Return min\_index;**

**}**

**Int printMST(int parent[], int n, int graph[V][V]) {**

**Int i;**

**Printf(“Edge Weight\n”);**

**For (i = 1; i < V; i++)**

**Printf(“%d - %d %d \n”, parent[i], i, graph[i][parent[i]]);**

**}**

**Void primMST(int graph[V][V]) {**

**Int parent[V];**

**Int key[V], i, v,count;**

**Int mstSet[V];**

**For (i = 0; i < V; i++)**

**Key[i] = INT\_MAX, mstSet[i] = 0;**

**Key[0] = 0;**

**Parent[0] = -1;**

**For (count = 0; count < V – 1; count++) {**

**Int u = minKey(key, mstSet);**

**mstSet[u] = 1;**

**for (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, V, graph);**

**}**

**Int main() {**

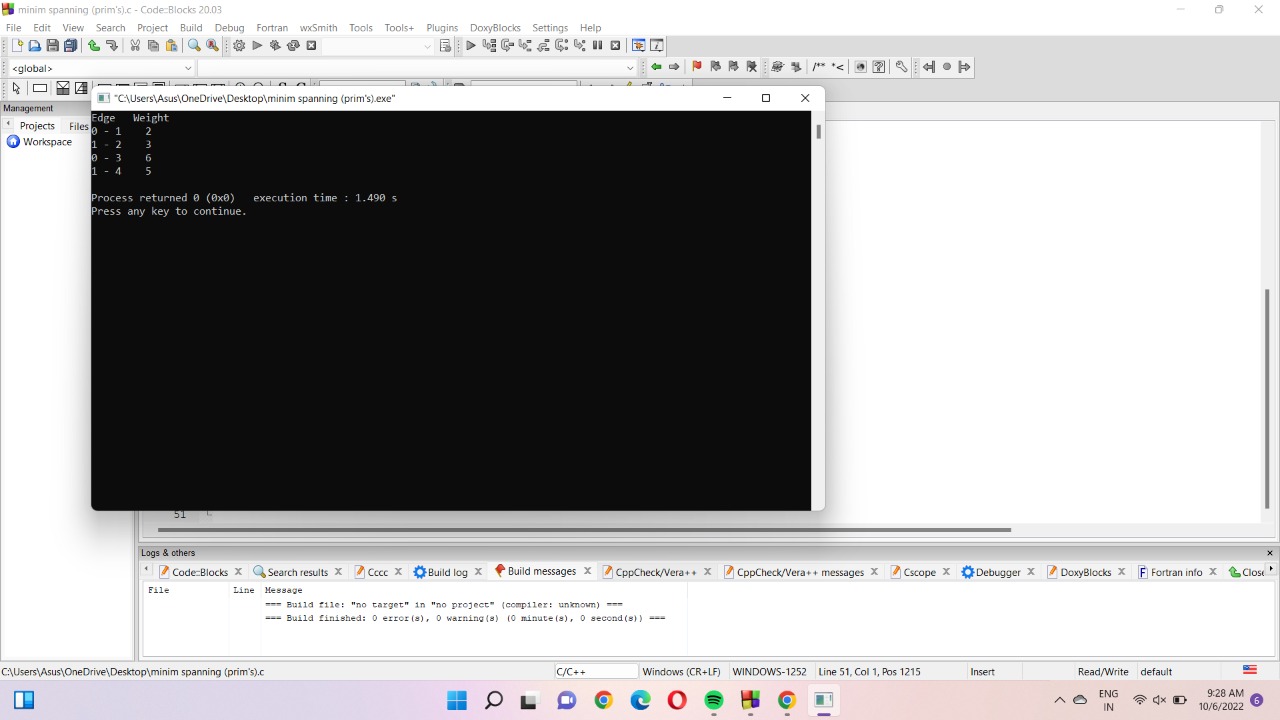
**Int graph[V][V] = { { 0, 2, 0, 6, 0 }, { 2, 0, 3, 8, 5 },**

**{ 0, 3, 0, 0, 7 }, { 6, 8, 0, 0, 9 }, { 0, 5, 7, 9, 0 }, };**

**primMST(graph);**

**return 0;**

**}**

****