```
// 24. Implementation of Minimum Spanning Tree using Prim's Algorithm
#include <limits.h>
#include <stdbool.h>
#include <stdio.h>
#define V 5
int minKey(int key[], bool mstSet[])
{
	int min = INT_MAX, min_index;
	for (int v = 0; v < V; v++)
		if (mstSet[v] == false && key[v] < min)
			min = key[v], min_index = v;
	return min_index;
}
int 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]]);
}
void primMST(int graph[V][V])
{
	int parent[V];
	int key[V];
	bool mstSet[V];
	for (int i = 0; i < V; i++)
		key[i] = INT_MAX, mstSet[i] = false;
	key[0] = 0;
	parent[0] = -1;
	for (int count = 0; count < V - 1; count++) {
		int u = minKey(key, mstSet);
		mstSet[u] = true;
		for (int v = 0; v < V; v++)
			if (graph[u][v] && mstSet[v] == false
				&& graph[u][v] < key[v])
				parent[v] = u, key[v] = graph[u][v];
	}
	printMST(parent, 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;
}
```

D:\data structures lab\prims.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug

merge sort.c | stack notations.c | merge.cpp | [*] stack nota.cpp | Minimum Spanning Tree using Prim's Algorithm.c | [*] prims.cpp

```
1  // 24. Implementation of Minimum Spanning Tree using Prim's Algorithm
2  #include <limits.h>
3  #include <stdbool.h>
4  #include <stdio.h>
5  #define V 5
6  int minKey(int key[], bool mstSet[])
7  {
8      int min = INT_MAX, min_index;
9      for (int v = 0; v < V; v++)
10         if (mstSet[v] == false && key[v] < min)
11             min = key[v], min_index = v;
12     return min_index;
13 }
14 int printMST(int parent[], int graph[V][V])
15 {
16     printf("Edge \tWeight\n");
17     for (int i = 1; i < V; i++)
18         printf("%d - %d \t%d \n", parent[i], i,
19                graph[i][parent[i]]);
20 }
21 void primMST(int graph[V][V])
22 {
23     int parent[V];
24     int key[V];
25     bool mstSet[V];
26     for (int i = 0; i < V; i++)
27         key[i] = INT_MAX, mstSet[i] = false;
28     key[0] = 0;
29     parent[0] = -1;
30     for (int count = 0; count < V - 1; count++) {
31         int u = minKey(key, mstSet);
32         mstSet[u] = true;
33         for (int v = 0; v < V; v++)
34             if (graph[u][v] && mstSet[v] == false
35                 && graph[u][v] < key[v])
```

Line: 32 Col: 28 Sel: 0 Lines: 53 Length: 1197 Insert Done parsing in 0.015 seconds

73°F Partly sunny

ENG IN 08:53 23-09-2022

D:\data structures lab\prims.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug

merge sort.c | stack notations.c | merge.cpp | [*] stack nota.cpp | Minimum Spanning Tree using Prim's Algorithm.c | [*] prims.cpp

```
20 }
21 void primMST(int graph[V][V])
22 {
23     int parent[V];
24     int key[V];
25     bool mstSet[V];
26     for (int i = 0; i < V; i++)
27         key[i] = INT_MAX, mstSet[i] = false;
28     key[0] = 0;
29     parent[0] = -1;
30     for (int count = 0; count < V - 1; count++) {
31         int u = minKey(key, mstSet);
32         mstSet[u] = true;
33         for (int v = 0; v < V; v++)
34             if (graph[u][v] && mstSet[v] == false
35                 && graph[u][v] < key[v])
36                 parent[v] = u, key[v] = graph[u][v];
37     }
38     printMST(parent, graph);
39 }
40 int main()
41 {
42     int graph[V][V] = { { 0, 2, 0, 6, 0 },
43                         { 2, 0, 3, 8, 5 },
44                         { 0, 3, 0, 0, 7 },
45                         { 6, 8, 0, 0, 9 },
46                         { 0, 5, 7, 9, 0 } };
47
48
49     primMST(graph);
50
51     return 0;
52 }
53
```

D:\data structures lab\prims.exe

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

--------------------------------
Process exited after 3.179 seconds with return value 0
Press any key to continue . . .
```

Line: 32 Col: 28 Sel: 0 Lines: 53 Length: 1197 Insert Done parsing in 0.015 seconds

73°F Partly sunny

ENG IN 08:53 23-09-2022