NAME: Satvik Dandale

GR NO: 1710797

ROLL NO: 13

DIV: TY E BATCH: B1

/\*

Finding a minimum spanning tree using Prim's Algorithm

\*/

#include <iostream>

using namespace std;

class Graph

{

int n, e, G[10][10], visited[10], total;

public:

Graph()

{

n = 10;

total = 0;

e = 10;

for (int i = 0; i < 10; i++)

{

visited[i] = 0;

for (int j = 0; j < 10; j++)

G[i][j] = 999;

}

}

void createGraph()

{

int a, b;

cout << "Enter total number of nodes (< 10): ";

cin >> n;

cout << "Enter total number of edges: ";

cin >> e;

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

{

cout << "Enter the edge's vertices: ";

cin >> a >> b;

cout << "Enter the edge's weight: ";

cin >> G[a][b];

G[b][a] = G[a][b];

}

}

void prims()

{

int min = 999, src, dest;

cout << "\n Enter the stating node: ";

cin >> src;

visited[src] = 1;

for (int edgecount = 0; edgecount < n - 1; edgecount++)

{

min = 999;

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

{

if (visited[i] == 1)

{

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

{

if (visited[j] != 1)

{

if (G[i][j] < min)

{

min = G[i][j];

src = i;

dest = j;

}

}

}

}

}

visited[dest] = 1;

cout << "\n"

<< src << "---" << dest << "\tcost: " << G[src][dest];

total += G[src][dest];

}

cout << "\nTotal cost of MST: " << total;

}

};

int main()

{

Graph o1;

o1.createGraph();

o1.prims();

}

OUTPUT:

Enter total number of nodes (< 10): 5

Enter total number of edges: 8

Enter the edge's vertices: 0

4

Enter the edge's weight: 1

Enter the edge's vertices: 2

3

Enter the edge's weight: 14

Enter the edge's vertices: 2

4

Enter the edge's weight: 7

Enter the edge's vertices: 0

3

Enter the edge's weight: 9

Enter the edge's vertices: 0

1

Enter the edge's weight: 5

Enter the edge's vertices: 1

2

Enter the edge's weight: 3

Enter the edge's vertices: 1

4

Enter the edge's weight: 2

Enter the edge's vertices: 1

3

Enter the edge's weight: 1

Enter the stating node: 0

0---4 cost: 1

4---1 cost: 2

1---3 cost: 1

1---2 cost: 3

Total cost of MST: 7