**Question 1:**

#include <iostream>

using namespace std;

void printMST(int\*\* matrix, int\* parent, int n)//function for printing MST

{

int cost= 0;

cout << "edges cost" << endl;

for (int i = 1; i < n; i++)

{ //vertex 1 vertex 2 weight

cout << parent[i] << "--" << i << " " << matrix[i][parent[i]] << endl;

cost += matrix[i][parent[i]];//adding total cost

}

cout << "Total Cost: " << cost;

}

void createMST(int\*\* matrix, int n)//function to create MST

{

int\* edges; int\* parent; int\* visit;

edges = new int[n];//array of edges

parent = new int[n];//array for parent

visit = new int[n];// array for checking visited vertex

int temp = 100, index;

for (int i = 0; i < n; i++)//initializing all the arrays

{

edges[i] = 100;

visit[i] = 0;

parent[i] = -1;

}

edges[0] = 0;

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

{

temp = 100;

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

{

if (visit[j] == 0 && edges[j] < temp)//chacking for minimum cost edge adjacent to parent

{

index = j;

temp = edges[j];

}

}

visit[index] = 1;//checking the vertex for visit

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

{

if (matrix[index][k] != 0 && visit[k] == 0 && matrix[index][k] < edges[k] )//condition for getting the weight between both vertices

{

edges[k] = matrix[index][k];//minmum edge corresponding to k vertex

parent[k] = index;//parent of k vertex

}

}

}

printMST(matrix, parent, n);//funtion to print MST

}

int main()

{

int\*\* arr;

int n, m, c;

cout << "Input number of vertices: ";//getting number of vertices

cin >> n;

arr = new int\* [n];//matrix of certain vertices

for (int i = 0; i < n; i++)//making matrix

{

arr[i] = new int[n];

}

for (int i = 0; i < n; i++)//initializing matrix with 0

{

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

{

arr[i][j] = 0;

}

}

cout << "Enter number of edges: ";//get number of edges

cin >> m;

int x, y;

for (int k = 0; k < m; k++)

{

cout << "Enter 1st vertice: ";//1st vertice

cin >> x;

cout << "Enter 2nd vertice: ";//2nd vertice

cin >> y;

cout << "Enter cost: ";//cost between two vertices

cin >> c;

arr[x - 1][y - 1] = c;

arr[y - 1][x - 1] = c;

}

cout << "Adjacency Matrix: " << endl << " ";

for (int i = 0; i < n; i++)//adcency matrix making

{

cout<< i + 1 << " ";

}

cout << endl;

for (int i = 0; i < n; i++)//output adjacency matrix

{

cout << i + 1 << " ";

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

{

cout << arr[i][j] << " ";

}

cout << endl;

}

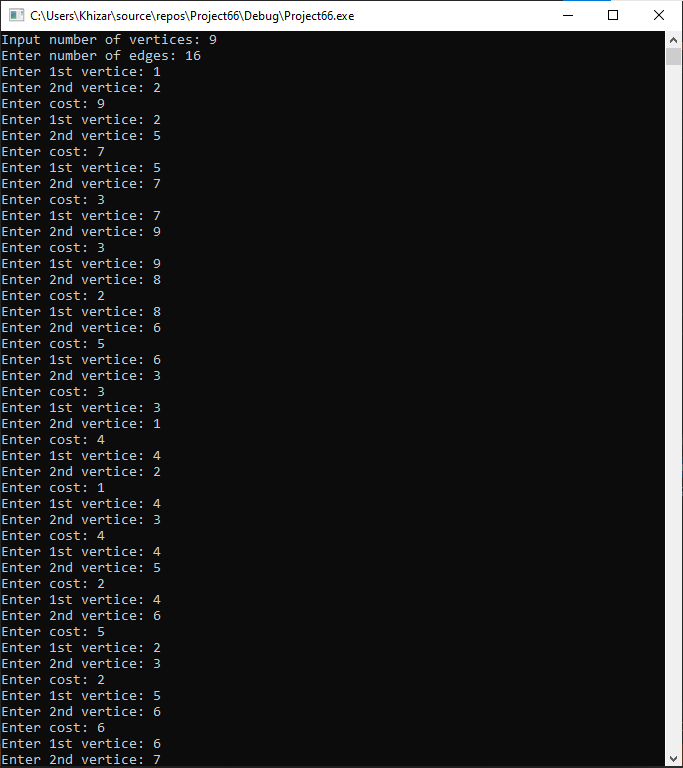
createMST(arr, n);//function to create MST

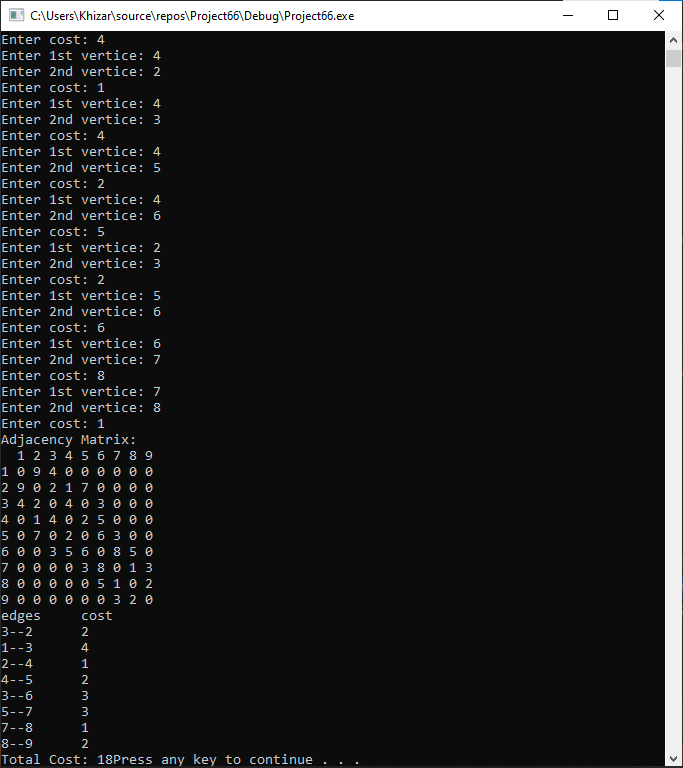
system("pause");

return 0;

}

**Output:**

****

****

**Question 2:**

#include <iostream>

using namespace std;

void printMST(int\*\* matrix, int\* parent, int n)

{

int minCost = 0;//getting minimum cost

cout << "edges cost" << endl;

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

{

if (parent[i] != -1)

{ // vertex 1 vertex 2 cost between them

cout << parent[i] + 1 << "--" << i + 1<< " " << matrix[parent[i]][i] << endl;

minCost += matrix[parent[i]][i];//adding minimum cost

}

}

cout << "minimum cost: " << minCost << endl;

}

void createMST(int\*\* matrix, int n, int\* cost, int m)//function to create MST

{

int\* parent;

int\* parent1;

parent1 = new int[n];

parent = new int[n];//array for getting vertices

bool\* ok;

ok = new bool[m];//checking if the weight is visited

int edge\_check = 0;//checking no of edges

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

{

parent[i] = -1;

}

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

{

ok[i] = false;

}

for (int i = 0; i < m; i++)//loop for checking weight

{

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

{

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

{

if (matrix[j][k] == cost[i] && ok[i] == false && edge\_check < n-1)//condition to get the weight

{

if (parent[j] != parent[k] || parent[k] == -1)

{

if (parent[k] == -1 && parent1[k] != parent[j])

{

ok[i] = true;

parent[k] = j;//j is the parent of k vertex

parent1[j] = k;

edge\_check++;//no of edge created

}

}

}

}

}

}

cout << endl;

printMST(matrix, parent, n);//function to print edges

}

int main()

{

int\*\* arr;

int n, m, c;

cout << "Input number of vertices: ";//getting number of vertices

cin >> n;

arr = new int\* [n];//matrix of certain vertices

for (int i = 0; i < n; i++)//making matrix

{

arr[i] = new int[n];

}

for (int i = 0; i < n; i++)//initializing matrix with 0

{

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

{

arr[i][j] = 0;

}

}

cout << "Enter number of edges: ";//get number of edges

cin >> m;

int\* costs;

costs = new int[m];//array for saving costs of all edges

int x, y;

for (int k = 0; k < m; k++)

{

cout << "Enter 1st vertice: ";//1st vertice

cin >> x;

cout << "Enter 2nd vertice: ";//2nd vertice

cin >> y;

cout << "Enter cost: ";//cost between two vertices

cin >> c;

arr[x - 1][y - 1] = c;

arr[y - 1][x - 1] = c;

costs[k] = c;

}

cout << "Adjacency Matrix: " << endl;

cout << " ";

for (int i = 0; i < n; i++)//adjacency matrix making

{

cout<< i + 1 << " ";

}

cout << endl;

for (int i = 0; i < n; i++)//output adjacency matrix

{

cout << i + 1 << " ";

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

{

cout << arr[i][j] << " ";

}

cout << endl;

}

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

{

for (int j = 0; j < m; j++)//sorting the weights in increasing order

{

if (costs[j] > costs[j + 1] && j+1 < m)

{

int t = costs[j];

costs[j] = costs[j + 1];

costs[j + 1] = t;

}

}

}

cout << endl;

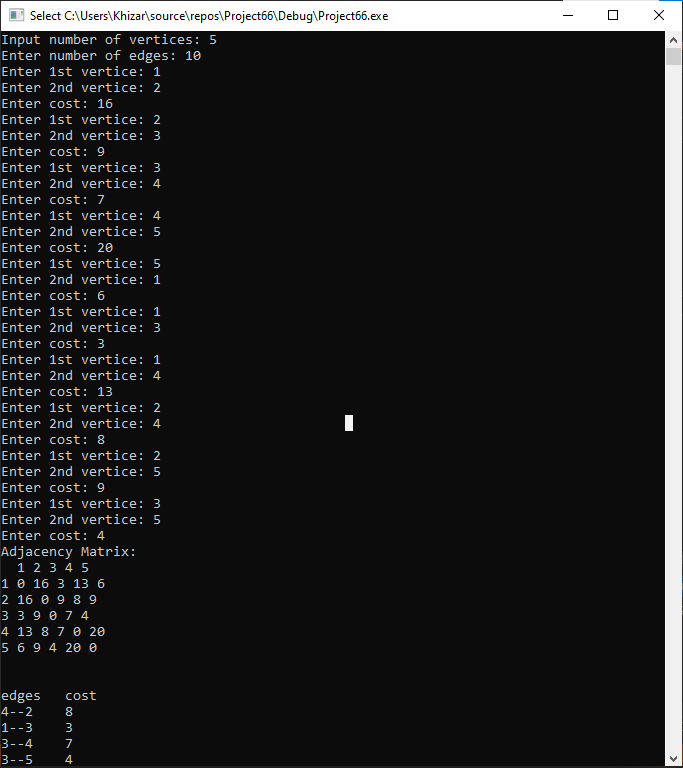
createMST(arr, n, costs, m);//function to create MST

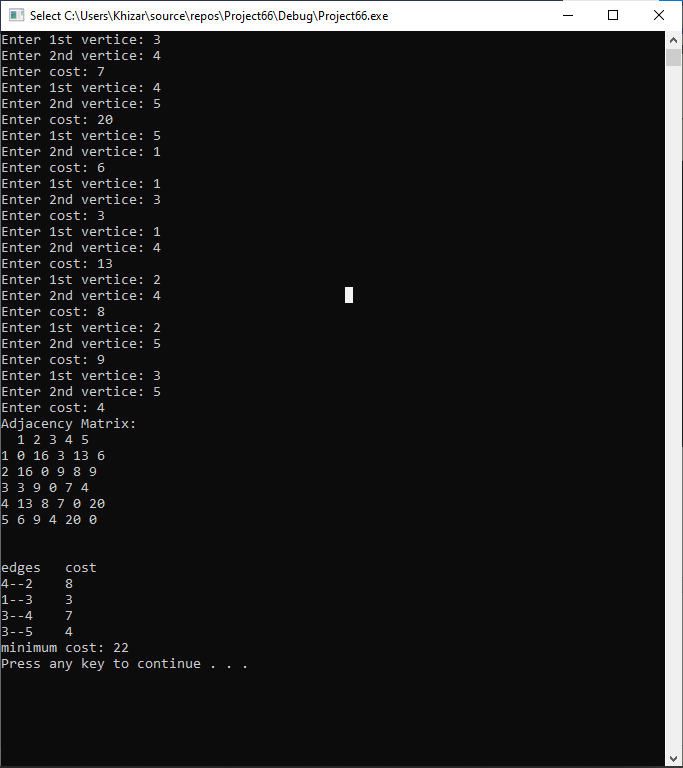
system("pause");

return 0;

}

**Output:**

****

****