```
#include <iostream>
using namespace std;
class Graph{
private:
    int n;
    int adjMatrix[20][20];
public:
    Graph();
    void create();
    void display();
    void prims(int start_v);};
Graph::Graph()
{cout << "Enter the number of Cities: ";
    cin >> n;
for (int i = 0; i < n; i++)
    {for (int j = 0; j < n; j++)
        {adjMatrix[i][j] = 999;}}}
void Graph::create()
{cout << "Enter adjacency and cost information: " << endl;
    for (int i = 0; i < n; i++)
    {for (int j = i + 1; j < n; j++)
        {char ans;
            cout << "Is City " << i << " and  " << j << " Connected? (y/n): ";
            cin >> ans;
            if (ans == 'y' || ans == 'Y')
            {int cost;
                cout << "Enter cost for edge between " << i << " and " << j << ": ";
                cin >> cost;
                adjMatrix[i][j] = cost;
                adjMatrix[j][i] = cost;}}}
cout << "Graph created successfully.\n";
    display();}
void Graph::display()
{cout << "\nAdjacency Matrix (Cost Representation):\n";
    for (int i = 0; i < n; i++)
    {for (int j = 0; j < n; j++)
        {cout << adjMatrix[i][j] << " ";
        }cout << endl;}}
void Graph::prims(int start_v)
{int nearest[20], t[20][3];
    int mincost = 0, r = 0;
for (int i = 0; i < n; i++)
    {if (i != start_v)
            nearest[i] = start_v;
        else
            nearest[i] = -1;}
for (int i = 1; i < n; i++)
    {int min = 999, j;
for (int k = 0; k < n; k++)
        {if (nearest[k] != -1 && adjMatrix[k][nearest[k]] < min)
            {min = adjMatrix[k][nearest[k]];
                j = k;}}
t[r][0] = nearest[j];
        t[r][1] = j;
        t[r][2] = min;
        r++;
mincost += min;
        nearest[j] = -1;
cout << "Nearest before after iteration " << i << ": ";
        for (int k = 0; k < n; k++)
        {cout << nearest[k] << " ";
        }cout << endl;
for (int k = 0; k < n; k++)
        {if (nearest[k] != -1 && adjMatrix[k][nearest[k]] > adjMatrix[k][j])
            {nearest[k] = j;}}
cout << "Nearest array after iteration " << i << ": ";
```

```
        for (int k = 0; k < n; k++)
        {cout << nearest[k] << " ";
        }cout << endl;
    }cout << "\nMinimum Spanning Tree Edges:\n";
    for (int i = 0; i < r; i++)
    {cout << "Edge: (" << t[i][0] << " - " << t[i][1] << ") Cost: " << t[i][2] <<
"\n";
    }cout << "Total Minimum Cost: " << mincost << "\n";
}int main()
{Graph g;
    g.create();
    int start_v;
    cout << "Enter the starting vertex: ";
    cin >> start_v;
    g.prims(start_v);}
```