

Code C-15:

```
#include<iostream>

using namespace std;

class Office
{
    int n;
    int a[10][10];
    string office[10];

public:
    void input();
    void display();
    void Prims();
};

void Office::input()
{
    cout<<"\nEnter no. of offices: ";
    cin>>n;
    cout<<"\nEnter the names of offices: ";
    for(int i=0 ; i<n ; i++)
        cin >> office[i];

    cout<<"\nEnter the cost to connect the offices: ";
    for(int i=0 ; i<n ; i++)
        for(int j=i ; j<n ; j++)
        {
            if(i==j)
            {
                a[i][j] = 0;
                continue;
            }

            cout<<"\nEnter the cost to connect " << office[i] <<"
and " << office[j]<<" : ";
            cin >> a[i][j];
            a[j][i] = a[i][j];
        }
}

void Office::display()
{
    for(int i=0 ; i<n ; i++)
    {
```

```

        cout<<"\n";
        for(int j=0 ; j<n ; j++)
        {
            cout<<a[i][j] << "\t";
        }
    }
}

void Office::Prims()
{
    int visit[n], minCost=0, count=1, minIndex, cost=0;
    for(int i=0 ; i<n ; i++)
        visit[i] = 0;

    cout<<"\n\nShortest path: ";
    visit[0]=1;
    cout<<office[0] << " -> ";
    while(1)
    {
        minCost = 10000;
        for(int i=0 ; i<n ; i++)
        {
            for(int j=0 ; j<n ; j++)
            {
                if(visit[i]==1 && a[i][j]!=0 && a[i][j]< minCost &&
visit[j]==0)
                {
                    minCost = a[i][j];
                    minIndex = j;
                }
            }
        }
        visit[minIndex]=1;
        cout<<office[minIndex] << " -> ";
        cost = cost + minCost;
        count++;

        if(count==n)
            break;
    }

    cout<<"\nMinimum cost: "<<cost;

}

int main()

```

```

{
    Office o1;
    int choice;
MENU:
    cout<<"\n\nMINIMUM SPANNING TREE";
    cout<<"\n1. Input data";
    cout<<"\n2. Display data";
    cout<<"\n3. Calculate minimum cost";
    cout<<"\n4. Exit";
    cout<<"\nEnter your choice: ";
    cin >> choice;
    switch(choice)
    {
    case 1:
        o1.input();
        break;
    case 2:
        o1.display();
        break;
    case 3:
        o1.Prims();
        break;
    case 4:
        return 0;
    default:
        cout<<"\nInvalid choice.Try again!";
    }
    if(choice != 5)
        goto MENU;
    return 0;
}

```


MINIMUM SPANNING TREE

1. Input data
2. Display data
3. Calculate minimum cost
4. Exit

Enter your choice: 1

Enter no. of offices: 3

Enter the names of offices: A B C

Enter the cost to connect the offices:

Enter the cost to connect A and B : 2

Enter the cost to connect A and C : 4

Enter the cost to connect B and C : 5

MINIMUM SPANNING TREE

1. Input data
2. Display data
3. Calculate minimum cost
4. Exit

Enter your choice: 2

0 2 4

2 0 5

4 5 0

MINIMUM SPANNING TREE

1. Input data
2. Display data
3. Calculate minimum cost
4. Exit

Enter your choice: 3

Shortest path: A -> B -> C ->

Minimum cost: 6