

ASSIGNMENT NO -7

You have a business with several offices; you want to lease phone lines to connect them up with each other; and the phone company charges different amounts of money to connect different pairs of cities. You want a set of lines that connects all your offices with a minimum total cost. Solve the problem by suggesting appropriate data structures

```
//*****PRIMS ALGORITHM*****  
#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<<"Enter number of offices:";  
    cin>>n;  
    cout<<"Enter names of offices:";  
    for(int i=0;i<n;i++)  
    {  
        cin>>office[i];  
    }  
    cout<<"Enter the cost to connect the offices:"<<endl;  
    for(int i=0;i<n;i++)  
    {  
        for(int j=i;j<n;j++)  
        {  
            if(i==j)  
            {  
                a[i][j]=0;  
                continue;  
            }  
            cout<<"Enter 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<<endl;
        for(int j=0;j<n;j++)
        {
            cout<<a[i][j]<<"\t";
        }
    }
}

void Office::Prims()
{
    int minCost = 0, minIndex, cost = 0, count = 1;
    int visit[n] = {0};
    visit[0] = 1;
    while (count != n) {
        minCost = 100000;

        for (int i = 0; i < n; i++) {
            if (visit[i] == 1) {
                for (int j = 0; j < n; j++) {
                    if (visit[j] == 0 && a[i][j] != 0 && a[i][j] < minCost) {
                        minCost = a[i][j];
                        minIndex = j;
                    }
                }
            }
        }

        visit[minIndex] = 1;
        cost += minCost;
        count++;
    }
    cout << "Minimum Cost is: " << cost << endl;
}

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.Prim();
    break;
case 4:
    cout<<"Thank you for using this Program!";
    return 0;
default:
    cout<<"\nInvalid choice.Try again!";
}
if(choice != 5)
    goto MENU;
return 0;
}

```

C:\Users\user\OneDrive\Desktop\DSAL Programs Final\prim.exe

```

MINIMUM SPANNING TREE
1. Input data
2. Display data
3. Calculate minimum cost
4. Exit
Enter your choice: 1
Enter number of offices:4
Enter names of offices:A
B
C
D
Enter the cost to connect the offices:
Enter cost to connect A and B : 2
Enter cost to connect A and C : 3
Enter cost to connect A and D : 4
Enter cost to connect B and C : 5
Enter cost to connect B and D : 6
Enter cost to connect C and D : 7

```

```

MINIMUM SPANNING TREE
1. Input data
2. Display data
3. Calculate minimum cost
4. Exit
Enter your choice: 2

```

```

0  2  3  4
2  0  5  6
3  5  0  7
4  6  7  0

```

```

MINIMUM SPANNING TREE
1. Input data
2. Display data
3. Calculate minimum cost
4. Exit
Enter your choice: 3
Minimum Cost is: 9

```

```

MINIMUM SPANNING TREE
1. Input data
2. Display data
3. Calculate minimum cost
4. Exit
Enter your choice: 4
Thank you for using this Program!

```