**Assignment No: 07**

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.

**Program:**

#include<iostream> #include<climits> using namespace std; template <class T> class Graph

{ int \*\* AM,num;

T \* data; public:

Graph(int n)

{ AM=new int\*[n]; for(int i=0;i<n;i++) AM[i]=new int[n]; num=n; data=new T[n]; cout<<"Enter names of all cities : "; for(int i=0;i<n;i++) cin>>data[i]; cout<<"Enter cost if you want to connect cities else enter 0: \n"; for(int j=0;j<n;j++) cout<<data[j]<<" "; cout<<endl; for(int i=0,cost=0;i<n;i++)

{ cout<<"Nodes connected to "<<data[i]<<" :\n"; for(int j=0;j<i;j++) cout<<AM[i][j]<<"\t"; for(int j=i;j<n;j++) if(j==i) {cout<<"0\t";AM[i][j]=AM[j][i]=0;} else {cin>>cost;AM[i][j]=AM[j][i]=cost;}

}

for(int i=0;i<n;i++) for(int j=0;j<n;j++) if(AM[i][j]==0)AM[i][j]=INT\_MAX;

}

void prims()

{

cout<<"\nCities that we need to connect:\n"; int \*visited=new int[num](),\*distance=new int[num],\*from=new int[num](),cost=0; visited[0]=1; for(int i=0;i<num;i++) distance[i]=AM[0][i]; int u,v;

for(int count=num-1;count>0;count--)

{ int min=INT\_MAX; for(int j=1;j<num;j++) if(visited[j]==0&&distance[j]<min)

{v=j;min=distance[j];} u=from[v]; cout<<data[u]<<"==>"<<data[v]<<"\tcost: "<<AM[u][v]<<endl;

visited[v]=1; for(int j=1;j<num;j++) if(visited[j]==0&&AM[j][v]<distance[j]) {distance[j]=AM[j][v];from[j]=v;} cost+=AM[u][v];

}

cout<<"Total cost of connecting all cities : "<<cost<<endl;

} }; int main() { int n; cout<<"Enter number of cities: "; cin>>n;

Graph<string> gr(n); gr.prims(); return 0;

}

**Output:**

