```
/*
Subject: DSA Laboratory
Practical No: 07
Title: A C++ Program to check whether the given Graph is Connected Graph or Not.
      Input : A Graph (04 Cities/Nodes and 05 Edges)
      Outputs:
         a) Represent Graph using Adjacency Matrix
         b) Represent Graph using Adjacency List
         c) To check whether the given Graph is Connected Graph or Not.


********************************************************************************
*/
                                 //......Header Files
#include <iostream>
using namespace std;

int adjMtx[4][4];   //....for Adjacency Matrix
int Row = 4;
int Col = 4;

struct Node        //....for Adjancency List
{
   string data;
   struct Node *down, *next;
}*Head;


                                 //......Function to return Vertex Name
string vertex(int val)
{
   if(val == 0)
     return "Nasik";
   else if(val == 1)
     return "Pune";
   else if(val == 2)
     return "Mumbai";
   else
     return "Nagpur";
}

                                 //......Function to Create Adjacency Matrix
void create_adjMtx()
{
   int i, j;
```

```
    for(i=0; i<Row; i++)
    {
        for(j=0; j<Col; j++)
        {
            cout<<"\n Is there a Flight from "<<vertex(i)<<" to "<<vertex(j)<<" ? : ";
            cin>>adjMtx[i][j];
        }
    }
}

                        //......Function to Display Adjacency Matrix
void display_adjMtx()
{
    int i, j;

    for(i=0; i<Row; i++)
    {
        cout<<"\n";
        for(j=0; j<Col; j++)
        {
            cout<<"\t"<<adjMtx[i][j];
        }
    }
}


                        //......Function to Create Adjacency List
void create_adjList()
{
    struct Node *Newnode, *move, *p;
    int i;
    int nodes;
    int edges;

    cout<<"\n\n How many Vertices in Graph: ";
    cin>>nodes;

    for(i=0; i<nodes; i++)
    {
        Newnode = new struct Node;

        Newnode->data = vertex(i);
        Newnode->down = NULL;
        Newnode->next = NULL;

        if(Head == NULL)
        {
          Head = Newnode;
          move = Head;
        }
        else
        {
           move->down = Newnode;
           move = move->down;
        }

    }

    move = Head;
    p = Head;
    while(move != NULL)
    {
        cout<<"\n How many adjacent vertices for "<<move->data<<" : ";
        cin>>edges;

        for(i=0; i<edges; i++)
        {
           Newnode = new struct Node;

           cout<<"\n\t Enter An Adjacent Vertex: ";
           cin>>Newnode->data;
           Newnode->down = NULL;
           Newnode->next = NULL;

           p->next = Newnode;
           p = p->next;
        }

        move = move->down;
          p = move;
    }

}

                              //......Function to Display Adjacency List
void display_adjList()
{
    struct Node *move, *p;

    cout<<"\n\t -Head-";

    move = Head;
    while(move != NULL)
    {
        cout<<"\n\t | "<<move->data<<" |--> ";
        p = move->next;
        while(p != NULL)
        {
            cout<<p->data<<" --> ";
            p = p->next;
        }
        cout<<"NULL";

        move = move->down;
           p = move;
    }
    cout<<"\t  NULL ";
}

                //.......Function to check is this a Connected Graph
void check_Connect()
{
    int i,j;
    int NonZero;

    i=0;

    do
    {
        NonZero = 0;
        for(j=0; j<Col; j++)
        {
            if(adjMtx[i][j] != 0)
               NonZero++;
        }
        i++;
    }while(i < Row && NonZero >= 1);

    if(i == Row)
       cout<<"\n\n The Given Graph is Connected Graph...!!!";
    else
       cout<<"\n\n *** The Given Graph is Not Connected Graph...!!!";
}

                                    //......Main Function
int main()
{
    cout<<"\n\n A C++ Program to check whether Graph is Connected.? ";
    cout<<"\n\n 1. Creating Adjacency Matrix.........\n";
    create_adjMtx();

    cout<<"\n\n 2. Display Adjacency Matrix.........\n";
    display_adjMtx();
    cout<<"\n Non-Zero Values: Distances in Kilometers";


    cout<<"\n\n 3. Create Adjacency List.........\n";
    Head = NULL;
    create_adjList();
    cout<<"\n Adjacency List is created Successfully...!!!";

    cout<<"\n\n 4. Display Adjacency List.........";
    display_adjList();

    cout<<"\n\n 5. Check Connected Graph......???";
    check_Connect();

    return 0;
}

/*
------------OUTPUT---------------


 A C++ Program to check whether Graph is Connected.?

 1. Creating Adjacency Matrix.........

 Is there a Flight from Nasik to Nasik ? : 0

 Is there a Flight from Nasik to Pune ? : 220

 Is there a Flight from Nasik to Mumbai ? : 180

 Is there a Flight from Nasik to Nagpur ? : 0

 Is there a Flight from Pune to Nasik ? : 220

 Is there a Flight from Pune to Pune ? : 0

 Is there a Flight from Pune to Mumbai ? : 120

 Is there a Flight from Pune to Nagpur ? : 716
 Is there a Flight from Mumbai to Nasik ? : 180

 Is there a Flight from Mumbai to Pune ? : 120

 Is there a Flight from Mumbai to Mumbai ? : 0

 Is there a Flight from Mumbai to Nagpur ? : 816

 Is there a Flight from Nagpur to Nasik ? : 0

 Is there a Flight from Nagpur to Pune ? : 716

 Is there a Flight from Nagpur to Mumbai ? : 816

 Is there a Flight from Nagpur to Nagpur ? : 0
```

2. Display Adjacency Matrix.........

```
    0      220    180    0
    220     0     120    716
    180    120     0     816
    0      716    816    0
Non-Zero Values: Distances in Kilometers
```

3. Create Adjacency List.........

```
How many Vertices in Graph: 4

How many adjacent vertices for Nasik : 2

        Enter An Adjacent Vertex: Pune

        Enter An Adjacent Vertex: Mumbai

How many adjacent vertices for Pune : 3

        Enter An Adjacent Vertex: Nasik

        Enter An Adjacent Vertex: Mumbai

        Enter An Adjacent Vertex: Nagpur

How many adjacent vertices for Mumbai : 3
```

```
        Enter An Adjacent Vertex: Nasik

        Enter An Adjacent Vertex: Pune

        Enter An Adjacent Vertex: Nagpur

 How many adjacent vertices for Nagpur : 2

        Enter An Adjacent Vertex: Pune

        Enter An Adjacent Vertex: Mumbai

 Adjacency List is created Successfully...!!!

 4. Display Adjacency List.........

          -Head-
        | Nasik |--> Pune --> Mumbai --> NULL
        | Pune |--> Nasik --> Mumbai --> Nagpur --> NULL
        | Mumbai |--> Nasik --> Pune --> Nagpur --> NULL
        | Nagpur |--> Pune --> Mumbai --> NULL
           NULL

 5. Check Connected Graph......???

 *** The Given Graph is Connected Graph...!!!

...Program finished with exit code 0
Press ENTER to exit console.
*/
```