```cpp
#include <iostream>

using namespace std;

int a[20][100], y, x, VIZ[100], TATA[100], n, m;

void creareMatrice(int n, int m, int a[20][100])
{
    int x, y;
    for (int i = 1; i <= m; i++)
    {
        cout << "Dati extremitatile muchiei " << i << ": ";
        cin >> x >> y;
        a[x][y]++;
        ///NEORIENTAT
        if (x != y)
            a[y][x]++;
    }
}

void Viziteaza (int x) {
    cout << x << " ";
}

void DF_Recursiv (int x)
{
    Viziteaza(x);
    VIZ[x] = 1;
    for (y = 1; y <= n; y++)
    {
        if (a[x][y] >= 1 && VIZ[y] == 0)
        {
            TATA[y] = x;
            DF_Recursiv(y);
        }
    }
}

int main()
{
    cout << "Nr. noduri: ";
    cin >> n;
    cout << "Nr. muchii: ";
    cin >> m;
    creareMatrice(n, m, a);

    cout << "Introduceti nodul de pornire: ";
    cin >> x;

    DF_Recursiv(x);
}
#include <iostream>
```

```
using namespace std;

int a[20][100], y, x, VIZ[100], TATA[100], n, m, S[100], URM[100];

void creareMatrice(int n, int m, int a[20][100])
{
    int x, y;
    for (int i = 1; i <= m; i++)
    {
        cout << "Dati extremitatile muchiei " << i << ": ";
        cin >> x >> y;
        a[x][y]++;
        ///NEORIENTAT
        if (x != y)
            a[y][x]++;
    }
}

void Viziteaza (int x) {
    cout << x << " ";
}

void DF (int x)
{
    Viziteaza(x);
    VIZ[x] = 1;
    TATA[x] = 0;
    int varf = 1;
    S[varf] = x;
    while (varf > 0)
    {
        int i = S[varf];
        int j = URM[i] + 1;

        while (a[i][j] == 0 && j <= n)
            j++;
        if (j > n)
            varf--;
        else
        {
            URM[i] = j;

            if (VIZ[j] == 0)
            {
                Viziteaza(j);
                VIZ[j] = 1;
                TATA[j] = i;
                varf++;
                S[varf] = j;
            }
        }
    }
}
```

```
int main()
{
    cout << "Nr. noduri: ";
    cin >> n;
    cout << "Nr. muchii: ";
    cin >> m;
    creareMatrice(n, m, a);

    cout << "Introduceti nodul de pornire: ";
    cin >> x;

    DF(x);
}
#include <iostream>

using namespace std;

int a[20][100], y, x, VIZ[100], TATA[100], n, m, URM[100], S[100];

void creareMatrice(int n, int m, int a[20][100])
{
    int x, y;
    for (int i = 1; i <= m; i++)
    {
        cout << "Dati extremitatile muchiei " << i << ": ";
        cin >> x >> y;
        a[x][y]++;
        ///NEORIENTAT
        if (x != y)
          a[y][x]++;
    }
}

void Viziteaza (int x) {
    cout << x << " ";
}

void BF (int x)
{
    Viziteaza(x);
    VIZ[x] = 1;
    TATA[x] = 0;
    int coada = 1, varf = 1;
    S[coada] = x;
    while (varf <= coada)
    {
        int i = S[varf];
        int j = URM[i] + 1;
        while (a[i][j] == 0 && j <= n)
            j++;
        if (j > n)
            varf++;
        else
```

```
        {
            URM[i] = j;
            if (VIZ[j] == 0)
            {
                Viziteaza(j);
                VIZ[j] = 1;
                TATA[j] = i;
                coada++;
                S[coada] = j;
            }
        }

    }
}

int main()
{
    cout << "Nr. noduri: ";
    cin >> n;
    cout << "Nr. muchii: ";
    cin >> m;
    creareMatrice(n, m, a);

    cout << "Introduceti nodul de pornire: ";
    cin >> x;

    BF(x);
}
#include <iostream>

using namespace std;

int a[20][100], y, x, VIZ[100], TATA[100], n, m, nrc, i, CC[100];

void creareMatrice(int n, int m, int a[20][100])
{
    int x, y;
    for (int i = 1; i <= m; i++)
    {
        cout << "Dati extremitatile muchiei " << i << ": ";
        cin >> x >> y;
        a[x][y]++;
        ///NEORIENTAT
        if (x != y)
            a[y][x]++;
    }
}

void Viziteaza (int x) {
    cout << x << " ";
    CC[x] = nrc;
}

void DF_Recursiv (int x)
```

```
{
    Viziteaza(x);
    VIZ[x] = 1;
    for (y = 1; y <= n; y++)
    {
        if (a[x][y] >= 1 && VIZ[y] == 0)
        {
            TATA[y] = x;
            DF_Recursiv(y);
        }
    }
}

void COMPONENTE_CONEXE()
{
    nrc = 0;
    for (i = 1; i <= n; i++)
        CC[i] = 0;
    for (i = 1; i <= n; i++)
    {
        if (CC[i] == 0)
        {
            nrc++;
            DF_Recursiv(i);
            cout << endl;
        }
    }
}

int main()
{
    cout << "Nr. noduri: ";
    cin >> n;
    cout << "Nr. muchii: ";
    cin >> m;
    creareMatrice(n, m, a);

    COMPONENTE_CONEXE();
}
#include <iostream>

using namespace std;

int a[20][100], y, x, VIZ[100], TATA[100], n, m, nrc, i, CC[100];

void creareMatrice(int n, int m, int a[20][100])
{
    int x, y;
    for (int i = 1; i <= m; i++)
    {
        cout << "Dati extremitatile muchiei " << i << ": ";
        cin >> x >> y;
        a[x][y]++;
    }
```

```
}

void Viziteaza (int x) {
    cout << x << " ";
    CC[x] = nrc;
}

void DF_Recursiv (int x)
{
    Viziteaza(x);
    VIZ[x] = 1;
    for (y = 1; y <= n; y++)
    {
        if ((a[x][y] >= 1 || a[y][x] >= 1) && VIZ[y] == 0)
        {
            TATA[y] = x;
            DF_Recursiv(y);
        }
    }
}

void COMPONENTE_CONEXE()
{
    nrc = 0;
    for (i = 1; i <= n; i++)
        CC[i] = 0;
    for (i = 1; i <= n; i++)
    {
        if (CC[i] == 0)
        {
            nrc++;
            DF_Recursiv(i);
            cout << endl;
        }
    }
}

int main()
{
    cout << "Nr. noduri: ";
    cin >> n;
    cout << "Nr. muchii: ";
    cin >> m;
    creareMatrice(n, m, a);

    COMPONENTE_CONEXE();
}
#include <iostream>

using namespace std;

int a[20][100], y, x, VIZ[100], TATA[100], n, m, nrc, i, CC[100];

void creareMatrice(int n, int m, int a[20][100])
```

```
{
    int x, y;
    for (int i = 1; i <= m; i++)
    {
        cout << "Dati extremitatile muchiei " << i << ": ";
        cin >> x >> y;
        a[x][y]++;
        ///NEORIENTAT
        if (x != y)
            a[y][x]++;
    }
}

void Viziteaza (int x) {
    CC[x] = nrc;
}

void DF_Recursiv (int x)
{
    Viziteaza(x);
    VIZ[x] = 1;
    for (y = 1; y <= n; y++)
    {
        if (a[x][y] >= 1 && VIZ[y] == 0)
        {
            TATA[y] = x;
            DF_Recursiv(y);
        }
    }
}

void COMPONENTE_CONEXE()
{
    nrc = 0;
    for (i = 1; i <= n; i++)
        CC[i] = 0;
    for (i = 1; i <= n; i++)
    {
        if (CC[i] == 0)
        {
            nrc++;
            DF_Recursiv(i);
        }
    }
}

int main()
{
    cout << "Nr. noduri: ";
    cin >> n;
    cout << "Nr. muchii: ";
    cin >> m;
    creareMatrice(n, m, a);
```

```
    COMPONENTE_CONEXE();
    if (nrc == 1)
        cout << "Graful este conex.";
    else
        cout << "Graful nu este conex.";
}
#include <iostream>

using namespace std;

int a[20][100], y, x, VIZ[100], TATA[100], n, m, nrc, i, CC[100];

void creareMatrice(int n, int m, int a[20][100])
{
    int x, y;
    for (int i = 1; i <= m; i++)
    {
        cout << "Dati extremitatile muchiei " << i << ": ";
        cin >> x >> y;
        a[x][y]++;
    }
}

void Viziteaza (int x) {
    CC[x] = nrc;
}

void DF_Recursiv (int x)
{
    Viziteaza(x);
    VIZ[x] = 1;
    for (y = 1; y <= n; y++)
    {
        if ((a[x][y] >= 1 || a[y][x] >= 1) && VIZ[y] == 0)
        {
            TATA[y] = x;
            DF_Recursiv(y);
        }
    }
}

void COMPONENTE_CONEXE()
{
    nrc = 0;
    for (i = 1; i <= n; i++)
        CC[i] = 0;
    for (i = 1; i <= n; i++)
    {
        if (CC[i] == 0)
        {
            nrc++;
            DF_Recursiv(i);
        }
    }
```

```
}

int main()
{
    cout << "Nr. noduri: ";
    cin >> n;
    cout << "Nr. muchii: ";
    cin >> m;
    creareMatrice(n, m, a);

    COMPONENTE_CONEXE();

    if (nrc == 1)
        cout << "Graful este conex.";
    else
        cout << "Graful nu este conex.";
}
```