MỘT SỐ NỘI DUNG

LÝ THUYẾT ĐỒ THỊ

BÀI TẬP 1 – ĐẾM SỐ BẬC CỦA ĐỈNH

Cho **đơn** đồ thị **vô hướng** có **n** đỉnh được đánh số từ 1 đến **n**.

Yêu cầu: Hãy cho biết bậc của từng đỉnh trong đồ thị

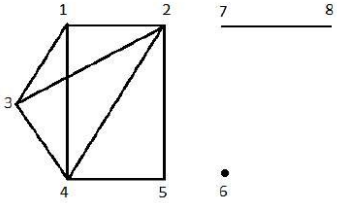
Dữ liệu vào: Cho từ file văn bản GRAPH1.INP có dạng:

- Dòng đầu ghi số nguyên dương **n** (1≤**n**≤1000).
- Dòng thứ **i** trong số **n** dòng tiếp theo ghi **n** số, số thứ **j** bằng 1 nếu đồ thị có cạnh nối từ đỉnh **i** đến đỉnh **j**, bằng 0 nếu không có cạnh nối từ **i** đến **j**.

Kết quả: Ghi ra file văn bản GRAPH1.OUT gồm một dòng ghi **n** số, số thứ **i** cho biết bậc của đỉnh **i** trong đồ thi.

Ví dụ:

GRAPH1.INP	GRAPH1.OUT
8 0 1 1 1 0 0 0 0 1 0 1 1 1 0 0 0 1 1 0 1 0 0 0 0 1 1 1 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0	3 4 3 4 2 0 1 1



Code giải mẫu

```
#include <bits/stdc++.h>
using namespace std;
int a[1001][1001];
int n, d, i, j;
int main()
{

    ifstream fin ("GRAPH1.INP");
    fin >> n;
    for (i=1; i<=n; i++)
        for (j=1; j<=n; j++) fin >> a[i][j];
    fin.close();
    ofstream fout ("GRAPH1.OUT");
    for (i=1; i<=n; i++)
    {
        d=0;
        for (j=1; j<=n; j++)
            if (a[i][j]==1) d++;
        fout << d << ' ';
    }
    fout.close();
    return 0;
}
```

BÀI TẬP 2 – ĐẾM SỐ BẬC CỦA ĐỈNH

Cho **đơn** đồ thị **vô hướng** có **n** đỉnh, **m** cạnh, các đỉnh được đánh số từ 1 đến **n**.

Yêu cầu: Hãy cho biết bậc của từng đỉnh trong đồ thị

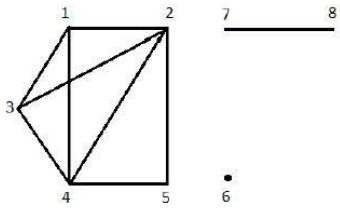
Dữ liệu vào: Cho từ file văn bản GRAPH2.INP có dạng:

- Dòng đầu ghi hai số nguyên dương **n**, **m** ($1 \le n \le 10^5$, $1 \le m \le 10^6$).
- **m** dòng tiếp theo, mỗi dòng ghi hai số u, v cho biết có cạnh nối từ đỉnh u đến đỉnh v.

Kết quả: Ghi ra file văn bản GRAPH2.OUT gồm một dòng ghi **n** số, số thứ **i** cho biết bậc của đỉnh **i** trong đồ thi.

Ví dụ:

GRAPH2.INP	GRAPH2.OUT
8 9 1 2 1 3 1 4 2 3 2 4 2 5 3 4 4 5 7 8	3 4 3 4 2 0 1 1



Code giải mẫu

```
#include <bits/stdc++.h>
using namespace std;
int U[1000001], V[1000001], Bac[1000001];
int n, m, i;
int main()
{

    ifstream fin ("GRAPH2.INP");
    fin >> n >> m;
    for (i=1; i<=m; i++) fin >> U[i] >> V[i];
    fin.close();
    ofstream fout ("GRAPH2.OUT");
    for (i=1; i<=m; i++)
    {
        Bac[U[i]]++;
        Bac[V[i]]++;
    }
    for (i=1; i<=n; i++) fout << Bac[i] << ' ';
    fout.close();
    return 0;
}
```

BÀI TẬP 3 – KIỂM TRA HAI ĐỈNH KỀ NHAU

Cho **đơn** đồ thị **vô hướng** có **n** đỉnh được đánh số từ 1 đến **n**. Có **Q** câu hỏi, mỗi câu hỏi cần trả lời hai đỉnh u, v cho trước có kề nhau hay không?

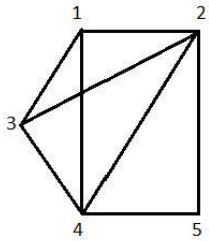
Dữ liệu vào: Cho từ file văn bản GRAPH3.INP có dạng:

- Dòng đầu ghi hai số nguyên dương **n**, **Q** ($1 \le n \le 1000$, $1 \le Q \le 10^5$).
- Dòng thứ **i** trong số **n** dòng tiếp theo ghi **n** số, số thứ **j** bằng 1 nếu đồ thị có cạnh nối từ đỉnh **i** đến đỉnh **j**, bằng 0 nếu không có cạnh nối từ **i** đến **j**.
- **Q** dòng tiếp theo, mỗi dòng ghi hai số nguyên dương tương ứng với một câu hỏi.

Kết quả: Ghi ra file văn bản GRAPH3.OUT gồm **Q** dòng tương ứng với **Q** câu hỏi, với mỗi câu hỏi ghi ra chữ YES nếu hai đỉnh kề nhau và ghi NO nếu hai đỉnh không kề nhau.

Ví dụ:

GRAPH3.INP	GRAPH3.OUT
5 3 0 1 1 1 0 1 0 1 1 1 1 1 0 1 0 1 1 1 0 1 0 1 0 1 0 3 5 1 4 2 5	NO YES YES



Code giải mẫu

```
#include <bits/stdc++.h>
using namespace std;
int a[1001][1001];
int n, Q, U, V, i, j;
int main()
{

    ifstream fin ("GRAPH3.INP");
    ofstream fout ("GRAPH3.OUT");
    fin >> n >> Q;
    for (i=1; i<=n; i++)
        for (j=1; j<=n; j++) fin >> a[i][j];
    for (i=1; i<=Q; i++)
    {
        fin >> U >> V;
        if (a[U][V]==1 || a[V][U]==1) fout << "YES";
        else fout << "NO";
        fout << endl;
    }
    fin.close();
    fout.close();
    return 0;
}
```

BÀI TẬP 4 – KIỂM TRA HAI ĐỈNH KỀ NHAU

Cho **đơn** đồ thị **vô hướng** có **n** đỉnh được đánh số từ 1 đến **n**. Có **Q** câu hỏi, mỗi câu hỏi cần trả lời hai đỉnh u, v cho trước có kề nhau hay không?

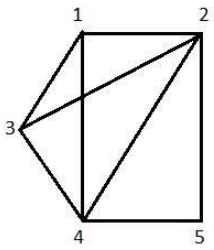
Dữ liệu vào: Cho từ file văn bản GRAPH4.INP có dạng:

- Dòng đầu ghi ba số nguyên dương **n**, **m**, **Q** ($1 \le n \le 10^9$, $1 \le m \le 10^5$, $1 \le Q \le 100$).
- **m** dòng tiếp theo, mỗi dòng ghi hai số u, v cho biết có cạnh nối từ đỉnh u đến đỉnh v.
- **Q** dòng tiếp theo, mỗi dòng ghi hai số nguyên dương tương ứng với một câu hỏi.

Kết quả: Ghi ra file văn bản GRAPH4.OUT gồm **Q** dòng tương ứng với **Q** câu hỏi, với mỗi câu hỏi ghi ra chữ YES nếu hai đỉnh kề nhau và ghi NO nếu hai đỉnh không kề nhau.

Ví dụ:

GRAPH4.INP	GRAPH4.OUT
5 8 3 1 2 1 3 1 4 2 3 2 4 2 5 3 4 4 5 3 5 1 4 2 5	NO YES YES



Code giải mẫu

```
#include <bits/stdc++.h>
using namespace std;
int U[100001], V[100001];
int n, m, Q, x, y, i, j;
int main()
{

    ifstream fin ("GRAPH4.INP");
    ofstream fout ("GRAPH4.OUT");
    fin >> n >> m >> Q;
    for (i=1; i<=m; i++) fin >> U[i] >> V[i];
    for (i=1; i<=Q; i++)
    {
        fin >> x >> y;
        for (j=1; j<=m; j++)
        if ((U[j]==x && V[j]==y) || (U[j]==y && V[j]==x)) break;
        if (j<=m) fout << "YES";
        else fout << "NO";
        fout << endl;
    }
    fin.close();
    fout.close();
    return 0;
}
```

BÀI TẬP 5 – CHUYỂN ĐỔI DẠNG BIỂU DIỄN ĐỒ THỊ

Cho **đơn** đồ thị **vô hướng** có **n** đỉnh được đánh số từ 1 đến **n**. Đồ thị được cho dưới dạng ma trận kề.

Yêu cầu: Hãy xuất ra dạng biểu diễn danh sách cạnh của đồ thị.

Dữ liệu vào: Cho từ file văn bản GRAPH5.INP có dạng:

- Dòng đầu ghi số nguyên dương **n** ($1 \le n \le 1000$).
- Dòng thứ **i** trong số **n** dòng tiếp theo ghi **n** số, số thứ **j** bằng 1 nếu đồ thị có cạnh nối từ đỉnh **i** đến đỉnh **j**, bằng 0 nếu không có cạnh nối từ **i** đến **j**.

Kết quả: Ghi ra file văn bản GRAPH5.OUT có dạng:

- Dòng đầu ghi hai số nguyên dương **n**, **m** tương ứng là số đỉnh và số cạnh của đồ thị.
- **m** dòng tiếp theo, mỗi dòng ghi hai số u, v nếu có cạnh nối từ đỉnh u đến đỉnh v.

Ví dụ:

GRAPH5.INP	GRAPH5.OUT
5 0 1 0 0 1 1 0 1 1 0 0 1 0 0 0 0 1 0 0 1 1 0 0 1 0	5 5 1 2 1 5 2 3 2 4 4 5

Code giải mẫu

```
#include <bits/stdc++.h>
using namespace std;
int a[1001][1001], U[1002001], V[1002001];
int n, m, i, j;
int main()
{

    ifstream fin ("GRAPH5.INP");
    fin >> n;
    for (i=1; i<=n; i++)
        for (j=1; j<=n; j++) fin >> a[i][j];
    fin.close();
    ofstream fout ("GRAPH5.OUT");
    m=0;
    for (i=1; i<=n; i++)
        for (j=1; j<=n; j++)
            if (j>i && a[i][j]==1)
        {
            m++;
            U[m]=i;
            V[m]=j;
        }
    fout << n << ' ' << m << endl;
    for (i=1; i<=m; i++) fout << U[i] << ' ' << V[i] << endl;
    fout.close();
    return 0;
}
```

Giải thích: Vì có tối đa **n** đỉnh nên có thể có tối đa **n*n** cạnh giữa hai đỉnh bất kì khác nhau. Ta thường khai báo dư 1001 đỉnh nên bình phương $1001^2 = 1002001$ (cạnh).

BÀI TẬP 6 – CHUYỂN ĐỔI DẠNG BIỂU DIỄN ĐỒ THỊ

Cho **đơn** đồ thị **vô hướng** có **n** đỉnh **m** cạnh, các đỉnh được đánh số từ 1 đến **n**. Đồ thị được cho dưới dạng danh sách cạnh.

Yêu cầu: Hãy xuất ra dạng biểu diễn ma trận kề của đồ thị.

Dữ liệu vào: Cho từ file văn bản GRAPH6.INP có dạng:

- Dòng đầu ghi hai số nguyên dương **n**, **m**($1 \le \mathbf{n} \le 1000$, $1 \le \mathbf{m} \le 10^5$).
- **m** dòng tiếp theo, mỗi dòng ghi hai số u, v cho biết có cạnh nối từ đỉnh u đến đỉnh v.

Kết quả: Ghi ra file văn bản GRAPH6.OUT có dạng:

- Dòng đầu ghi số nguyên dương **n** là số đỉnh của đồ thị.
- Dòng thứ **i** trong số **n** dòng tiếp theo ghi **n** số, số thứ **j** bằng 1 nếu đồ thị có cạnh nối từ đỉnh **i** đến đỉnh **j**, bằng 0 nếu không có cạnh nối từ **i** đến **j**.

Ví dụ:

GRAPH6.INP	GRAPH6.OUT
5 5 1 2 1 5 2 3 2 4 4 5	5 0 1 0 0 1 1 0 1 1 0 0 1 0 0 0 0 1 0 0 1 1 0 0 1 0

Code giải mẫu

```
#include <bits/stdc++.h>
using namespace std;
int a[1001][1001];
int n, m, U, V, i, j;
int main()
{

    ifstream fin ("GRAPH6.INP");
    fin >> n >> m;
    for (i=1; i<=m ; i++)
    {
        fin >> U >> V;
        a[U][V]=1;
        a[V][U]=1;
    }
    fin.close();
    ofstream fout ("GRAPH6.OUT");
    fout << n << endl;
    for (i=1; i<=n; i++)
    {
        for (j=1; j<=n; j++) fout << a[i][j] << ' ';
        fout << endl;
    }
    fout.close();
    return 0;
}
```

BÀI TẬP 7 – DANH SÁCH KỀ - Tìm các đỉnh kề

Cho **đơn** đồ thị **vô hướng** có **n** đỉnh được đánh số từ 1 đến **n**.

Yêu cầu:

- Tổ chức lưu trữ đồ thị bằng danh sách kề.
- Với mỗi đỉnh v của đồ thị, hãy cho biết những đỉnh kề với đỉnh v.

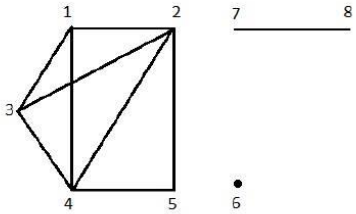
Dữ liệu vào: Cho từ file văn bản GRAPH7.INP có dạng:

- Dòng đầu ghi số nguyên dương **n** ($1 \le n \le 1000$).
- Dòng thứ **i** trong số **n** dòng tiếp theo ghi **n** số, số thứ **j** bằng 1 nếu đồ thị có cạnh nối từ đỉnh **i** đến đỉnh **j**, bằng 0 nếu không có cạnh nối từ **i** đến **j**.

Kết quả: Ghi ra file văn bản GRAPH7.OUT gồm n dòng, dòng thứ v ghi số đầu tiên là số đỉnh kề với đỉnh v, tiếp theo là danh sách các đỉnh kề với đỉnh v theo thứ tự từ nhỏ đến lớn.

Ví dụ:

GRAPH7.INP	GRAPH7.OUT
8	3 2 3 4
0 1 1 1 0 0 0 0	4 1 3 4 5
1 0 1 1 1 0 0 0	3 1 2 4
1 1 0 1 0 0 0 0	4 1 2 3 5
1 1 1 0 1 0 0 0	2 2 4
0 1 0 1 0 0 0 0	0
0 0 0 0 0 0 0 0	1 8
0 0 0 0 0 0 0 1	1 7
0 0 0 0 0 0 1 0	



Code giải mẫu

```
#include <iostream>
#include <fstream>
#include <vector>
#include <algorithm>
using namespace std;
vector < vector<int> > Ke(1001);
int n, k, i, j;
int main()
{
    ifstream fin ("GRAPH7.INP");
    fin >> n;
    for (i=1; i<=n; i++)
        for (j=1; j<=n; j++)
    {
        fin >> k;
        if (k==1) Ke[i].push_back(j);
    }
    fin.close();
    ofstream fout ("GRAPH7.OUT");
    for (i=1; i<=n; i++)
    {
        fout << Ke[i].size() << ' ';
        for (j=0; j<Ke[i].size(); j++) fout << Ke[i][j] << ' ';
        fout << endl;
    }
    fout.close();
    return 0;
}
```

BÀI TẬP 8 – DANH SÁCH KỀ - Kiểm tra hai đỉnh kề nhau

Cho **đơn** đồ thị **vô hướng** có **n** đỉnh **m** cạnh, các đỉnh được đánh số từ 1 đến **n**. Có **Q** câu hỏi, mỗi câu hỏi cần trả lời hai đỉnh u, v cho trước có kề nhau hay không?

Yêu cầu:

- Tổ chức lưu trữ đồ thị bằng danh sách kề.
- Trả lời Q câu hỏi.

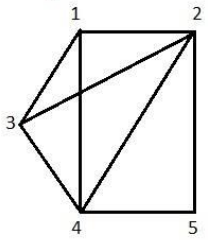
Dữ liệu vào: Cho từ file văn bản GRAPH8.INP có dạng:

- Dòng đầu ghi ba số nguyên dương **n**, **m**, **Q** ($1 \le \mathbf{n} \le 10^5$, $1 \le \mathbf{m} \le 10^5$, $1 \le \mathbf{Q} \le 100$).
- **m** dòng tiếp theo, mỗi dòng ghi hai số u, v cho biết có cạnh nối từ đỉnh u đến đỉnh v.
- **Q** dòng tiếp theo, mỗi dòng ghi hai số nguyên dương tương ứng với một câu hỏi.

Kết quả: Ghi ra file văn bản GRAPH8.OUT gồm **Q** dòng tương ứng với **Q** câu hỏi, với mỗi câu hỏi ghi ra chữ YES nếu hai đỉnh kề nhau và ghi NO nếu hai đỉnh không kề nhau.

Ví dụ:

GRAPH8.INP	GRAPH8.OUT
5 8 3 1 2 1 3 1 4 2 3 2 4 2 5 3 4 4 5 3 5 1 4 2 5	NO YES YES



Code giải mẫu

```
#include <iostream>
#include <fstream>
#include <vector>
#include <algorithm>
using namespace std;
vector < vector <int> > Ke(100001);
int n, m, Q, i, j, u, v;
int main()
{
    ifstream fin ("GRAPH8.INP");
    ofstream fout ("GRAPH8.OUT");
    fin >> n >> m >> Q;
    for (i=1; i<=m ;i++)
    {
        fin >> u >> v;
        Ke[u].push_back(v);
        Ke[v].push_back(u);
    }
    for (i=1; i<=Q; i++)
    {
        fin >> u >> v;
        for (j=0; j<Ke[u].size(); j++)
            if (Ke[u][j]==v) break;
        if (j<Ke[u].size()) fout << "YES";
        else fout << "NO";
        fout << endl;
    }
    fin.close();
    fout.close();
    return 0;
}
```