

Matrix Transform

Submit solution

All submissions
Best submissions

✓ Points: 100 (partial)② Time limit: 0.3s

■ Memory limit: 256M

Allowed languages

C, C++

For an undirected graph (with no self-loops or duplicate edges), the L matrix is an $n \times n$ matrix (where n is the number of vertices) defined as follows:

- For each vertex, the diagonal entry is equal to the degree of that vertex (i.e. the number of edges incident to it).
- \bullet For any two distinct vertices, the entry is -1 if there is an edge connecting them, and 0 if there is no edge.

One elegant way to compute the L is to start with the OI matrix of the graph. In the OI matrix, each column corresponds to an edge and has a -1 at the vertex where the edge starts (tail) and a +1 at the vertex where the edge ends (head), with all other entries being 0. Only one edge of u,v or v,u would be present in the given incidence matrix.

Your task is to take the OI matrix as input and compute the corresponding L of the graph.

We Challenge you to do this problem in couple of lines(in C, and single line in C++) using the given boiler plate code, after taking the input.

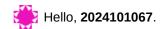
Input Format

- The first line contains two integers n and m, where:
 - \circ *n* is the number of vertices. (1 \leq *n* \leq 100)
 - m is the number of edges. $(0 \le m \le n^2)$
 - \circ The next n lines each contain m space-separated integers representing the OI matrix.
 - For each column (edge):
 - ullet A value of -1 indicates the tail (starting vertex) of that edge.
 - A value of +1 indicates the head (ending vertex) of that edge.

proudly powered by **DMOJ** | English (en)

1 of 4 3/23/25, 09:12





Output Format

- ullet Output the L matrix as an nxn matrix.
- ullet Print n lines, each with n space-separated integers.
- The entry in the i-th row and j-th column of the output should be:
 - \circ The degree of vertex i if i == j.
 - $\circ \ -1$ if vertices i and j are connected by an edge.
 - 0 otherwise.

Sample Testcase 1

Input

```
Copy
-1 0
1 -1
0 1
```

Input Explanation

- The first column represents an edge connecting vertex 1 and vertex 2.
- The second column represents an edge connecting vertex 2 and vertex 3.
- Vertex 1 is incident with 1 edge, vertex 2 with 2 edges, and vertex 3 with 1 edge.

Output

```
1 -1 0
-1 2 -1
0 -1 1
```

Output Explanation

Graph Interpretation

• First column:

The entries are:

- \circ Vertex 1: -1
- \circ Vertex 2: 1
- ∘ Vertex 3: 0

This indicates an edge between vertex 1 and vertex 2.

2 of 4 3/23/25, 09:12





- Vertex 1: 0
- \circ Vertex 2: -1
- Vertex 3: 1

This indicates an edge between vertex 2 and vertex 3.

Degree Computation

• Vertex 1:

Incident only in the first column \rightarrow degree is 1.

• Vertex 2:

Incident in both columns \rightarrow degree is 2.

Vertex 3:

Incident only in the second column \rightarrow degree is 1.

L Matrix Construction

• Diagonal entries:

Place the vertex degrees:

- $\circ L[1][1] = 1$
- L[2][2] = 2
- L[3][3] = 1

• Off-diagonal entries:

For each edge, put -1 at the positions corresponding to the connected vertices:

- \circ Edge between vertex 1 and vertex 2 gives L[1][2] = L[2][1] = -1.
- Edge between vertex 2 and vertex 3 gives L[2][3] = L[3][2] = -1.
- All other off-diagonals are 0.

Sample Testcase 2

Input

Copy
-1 0 0
1 -1 0
0 1 -1
0 0 1

Input Explanation

- The first column corresponds to an edge between vertex 1 and vertex 2.
- The second column corresponds to an edge between vertex 2 and vertex 3.

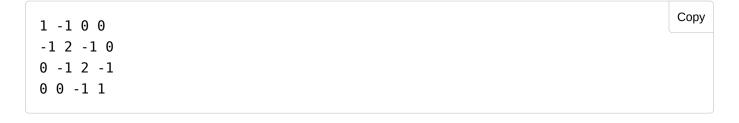
3 of 4



Hello, **2024101067**.

vertex 4 has degree 1.

Output



Clarifications

Request clarification

No clarifications have been made at this time.

4 of 4 3/23/25, 09:12