**Set A**

**Final Lab Examination**

**CSE221-Algorithms**

**Duration: 1 Hour 10 Minutes**

**Total Marks: 20**

| **Tasks**   1. Taking correct from inputs from a file 2. Making an adjacency list or a matrix from the input 3. Applying correct algorithm and solving the first problem with right approach 4. Tweaking the solution of the main problem to solve the second problem 5. Generating the output of the first problem and storing in a file 6. Generating the output of the second problem and storing in a file | **Marks**  3  3  6  4  2  2 |
| --- | --- |

**Part A [Read Part B before solving Part A ]**

Suppose, you are hired as a program coordinator of a corporate office. You are assigned to make a sequence of tasks to attain program objectives. There are some tasks which have prerequisites. Suppose, task B can only be completed after task A has been completed. Task A has no prerequisites. Task C and D can only be completed after task B has been completed.

The sequence of the tasks may look like this:



In this problem, there are **N** tasks in the curriculum. There are **M** prerequisite requirements of the form "Task **A** has to be completed before task **B**".

Your task is to find an order in which you can complete the tasks.

**Input:**

The first input line has two integers **N** (1 <= N <= 1000) and **M** (1 <= M <= N2) - the number of tasks and prerequisite requirements. The tasks are numbered 1,2,3,…,N.

Next, there are M lines describing the requirements. Each line has two integers **A**, **B** (1 <= A, B <= N)- task A has to be completed before task B.

**Output:**

Print an order in which you can complete the tasks. Please note, there could be multiple correct sequences. You can print any valid order that includes all the tasks.

If there are no solutions, print "IMPOSSIBLE".

| Sample Input 1 | Sample Output 1 |
| --- | --- |
| 5 3  3 1  1 2  4 5 | 3 4 1 5 2 |
| Sample Input 2 | Sample Output 2 |
| 6 6  1 2  2 3  4 3  4 5  5 6  6 4 | IMPOSSIBLE |
| Sample Input 3 | Sample Output 3 |
| 8 10  1 2  1 4  2 4  2 5  2 3  4 6  4 5  6 5  5 3  7 8 | 1 7 2 8 4 6 5 3 |

**Part B**

You have successfully made a sequence of the tasks. Now, the head of the department wants you to make some modifications so that the sequence you made is lexicographically smallest.

The problem statement of this problem is the same as **Part A.**

The only difference is that in this task, you have to find the lexicographically smallest valid task sequence.

If you have two sequences, for example: A: 3 → 1 → 2 → 4 and B: 3 → 1 → 4 → 2. Then path A is lexicographically smaller than path B.

**Input:**

The first input line has two integers **N** (1 <= N <= 1000) and **M** (1 <= M <= N2) - the number of tasks and prerequisite requirements. The tasks are numbered 1,2,3,…,N.

Next, there are M lines describing the requirements. Each line has two integers **A**, **B** (1 <= A, B <= N)- task A has to be completed before task B.

**Output:**

Print the lexicographically smallest valid task sequence in which you can complete the tasks.

If there are no solutions, print "IMPOSSIBLE".

| Sample Input 1 | Sample Output 1 |
| --- | --- |
| 5 3  3 1  1 2  4 5 | 3 1 2 4 5 |
| Sample Input 2 | Sample Output 2 |
| 6 6  1 2  2 3  4 3  4 5  5 6  6 4 | IMPOSSIBLE |
| Sample Input 3 | Sample Output 3 |
| 8 10  1 2  1 4  2 4  2 5  2 3  4 6  4 5  6 5  5 3  7 8 | 1 2 4 6 5 3 7 8 |