

Problem H. Hacks With Includes

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 1024 mebibytes

Short code is cool, reasonable, beautiful, and elegant. You love short code. Hence you want to make your code as short as possible. Several techniques are known to make your code shorter. Today, you focus on *includes* in your code.

There are N files you must include to your code. The N files are numbered as 1 through N . Some of them also include others. If file a includes file b and file b includes file c , including a into your code implies including b and c into your code. But including c does not necessarily imply including a or b unless c (indirectly) includes a or b .

You are given information about dependencies of N files, i -th of which describes file a_i includes b_i . From this information, your task is to determine the set of the minimum number of files you have to directly include in order to include all N files indirectly, and output file numbers in the minimum set in ascending order. If such a set is not uniquely determined, output a set with the minimum sum of the file numbers in a set.

Input

The first line of the input consists of two integers N and M , where N is the number of files ($1 \leq N \leq 30\,000$) and M is the number of dependency information ($0 \leq M \leq 5 \times 10^5$). The following M lines represents each dependency, the i -th of which contains two integers a_i and b_i , which means file a_i includes file b_i ($1 \leq a_i \leq N$, $1 \leq b_i \leq N$). There is no duplicate dependency information, i.e. $a_i \neq a_j$ or $b_i \neq b_j$ hold for $1 \leq i < j \leq M$.

Output

Determine the minimum number of files that must be directly included in your code to include all files indirectly, and print file numbers in such a file set in ascending order. If there are multiple sets with the minimum size, output a set with the minimum sum of file numbers.

Examples

standard input	standard output
4 3 2 1 2 4 3 1	2 3
5 6 2 1 2 4 3 1 3 2 5 1 5 2	3 5
9 11 1 3 2 4 2 6 4 1 5 3 5 6 5 8 6 8 7 4 8 1 8 2	5 7 9