Task: DRO
Bike paths

Olimpiada Informatyczna

XXV OI, Stage II, Day one. Source file `dro.*` Available memory: 64 MB. *14.02.2018*

King Byteasar finally surrendered to Byteburg citizens' demands and earmarked a fraction of the budget surplus for paving bike paths within the city. The royal advisor for road infrastructure has already come up with a network design project, to which he applied many modifications requested by his majesty himself. The network consists of *one-way* segments linking certain intersections. A path from intersection u to another intersection v is a sequence of distinct intersections $u = v_0, v_1, \ldots, v_k = v$ such that each successive two v_i, v_{i+1} (for $0 \leq i < k$) are linked by a segment directed from v_i to v_{i+1}.

The king demanded that the network be *just*, meaning that it should satisfy the following property: If it is *not* possible to reach intersection u from intersection v (i.e., there is no path from v to u), then there can be *at most one* path from u to v. The king believes that this should ensure that people living next to intersection v will not envy those living next to intersection u.

Members of the Citizens Bicycle Committee have recently obtained a copy of the just network design, which is not to their liking at all. They insist that the proposed network overly restricts the movement by bicycles in the city. They are planning to publish a report on the issue, for which they need some hard data. Your task is to determine the reachability degree, i.e., for each intersection v, you should calculate the number of intersections reachable from v.

Input

In the first line of the standard input, there are two integers n and m ($n \geq 2$, $m \geq 1$), separated by a single space, specifying the number of intersections in Byteburg and the number of segments in the network design. The intersections are numbered from 1 to n.

The m lines that follow specify the network: Each line contains two integers a and b ($1 \leq a, b \leq n$, $a \neq b$), separated by a single space, indicating that there is a segment directed from intersection a to intersection b. Every ordered pair (a, b) will appear on input at most once. It is guaranteed that the input network is just.

Output

Exactly n lines should be printed to the standard output, the i-th of which (for $i = 1, \ldots, n$) should contain a single integer: the number of intersections reachable from intersection i.

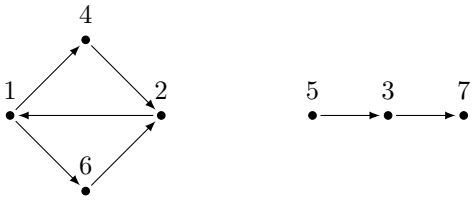
Example

For the input data:

```
7 7
1 4
1 6
4 2
6 2
2 1
5 3
3 7
```

the correct result is:

```
3
3
1
3
2
3
0
```



Sample grading tests:

`1ocen:` $n = 25$, $m = 600$, every intersection can be reached from every other;

`2ocen:` $n = 55$, $m = 54$, an isolated intersection and disjoint cycles of lengths $2, \ldots, 10$;

`3ocen:` $n = 50\,000$, $m = 49\,999$, a single path leads through all intersections;

`4ocen:` $n = m = 50\,000$, all intersection are arranged along a single cycle.

Grading

The set of tests consists of the following subsets. Within each subset, there may be several unit tests.

Subset	Property	Score
1	$n \leq 60$	12
2	$n, m \leq 5000$	8
3	$n \leq 50\,000$, $m \leq 100\,000$, moreover: if $u > v$, then there is no path from u to v	18
4	$n \leq 50\,000$, $m \leq 100\,000$, moreover: if there is any path from u to v, then there is no path from v to u	18
5	$n \leq 50\,000$, $m \leq 100\,000$	44