Water park (`waterslide`)

There's a new water park in the city and Edoardo, the owner, cannot wait to open! However, he is in doubt on where to put the life guards.

To access a pool you have to take a ride (that is, a path from the launch pad) through some slides into a pool. While going down, you may encounter a junction. Note that there could be junctions where many slides enter and many exit: in this case, you will choose one of the many exits *uniformly at random*. Obviously, there cannot be any cyclic paths since water always flows down.

Figure 1: A nice waterslide.

Edoardo knows very well that most of the accidents happen where the most people are. You are given a description of the junctions and slides connecting them. Help Edoardo find which is the ending pool where most of the people will arrive!

> ☞ Among the attachments of this task you may find a template file `waterslide.*` with a sample incomplete implementation.

Input

The first line contains the number of junctions N, the number of slides M and the number of pools P. The launch pad has index 0, the ending pools have a number between $N - P$ and $N - 1$. The next M lines contain two integers A_i, B_i each: the starting and ending junction of each slide (respectively).

Output

You need to write a single line with an integer: the index of the pool where most of the people will arrive.

Constraints

- $1 \leq P < N \leq 100\,000$.
- $1 \leq M \leq 200\,000$.
- $0 \leq A_i, B_i < N$ for all $i = 0 \ldots M - 1$.

- The network of slides is always connected.
- There are no slides starting from pools.
- Every path has to eventually reach a pool.
- If multiple equivalent solutions are possible, you can print any one of them.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

- **Subtask 1** [**0 points**]: Examples.
- **Subtask 2** [**15 points**]: Exactly one slide enter and one exits each junction.
- **Subtask 3** [**30 points**]: $N \leq 1000$.
- **Subtask 4** [**30 points**]: Exactly one slide enter each junction.[1]
- **Subtask 5** [**25 points**]: No additional limitations.

Examples

input.txt	output.txt
5 6 2 0 1 0 2 1 2 2 3 2 4 0 4	4
7 7 3 0 6 0 1 1 5 1 2 2 4 2 3 3 4	6

Explanation

In the **first sample case**, one third of the people will arrive at pool 3 and two thirds at pool 4.

In the **second sample case**, $\frac{1}{4}$ will arrive at pool 4, $\frac{1}{4}$ at pool 5 and $\frac{1}{2}$ at pool 6.

[1] For this purpose, pools do not count as junctions, so many slides might enter in them.