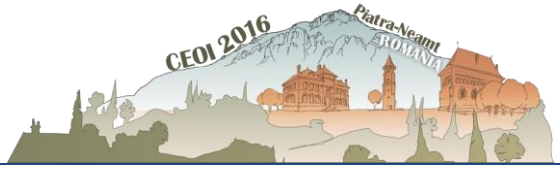

A garden is composed of a row of `N` cells numbered from `1` to `N`. Initially, all cells contain plants. A kangaroo arrived in the garden in cell numbered `cs`. Then he jumps from cell to cell, eating the plants as he goes. He will always finish in cell numbered `cf`, after visiting each of the `N` cells exactly once, including `cs` and `cf`. Obviously, the kangaroo will make `N-1` jumps.

The kangaroo doesn't want to be caught, so after each jump he changes the direction in which he jumps next: if he is currently in cell numbered `current` after he jumped there from a cell numbered `prev`, and will jump from `current` to cell numbered `next`, then:

- if `prev < current`, then `next < current`; else,
- if `current < prev`, then `current < next`.

Knowing the number `N` of cells in the garden, the starting cell `cs` from where the kangaroo starts to eat plants and the final cell `cf` where the kangaroo finishes, you should calculate the number of distinct routes the kangaroo can take while jumping through the garden.

Input format

The input file `kangaroo.in` will contain three space separated positive integers `N`, `cs`, `cf`.

Output format

In the output file `kangaroo.out` you should write a single integer, the number of distinct routes the kangaroo can take modulo `1000000007` ($10^9 + 7$).

Notes and constraints

- $2 \leq N \leq 2000$
- $1 \leq cs \leq N$
- $1 \leq cf \leq N$
- $cs \neq cf$
- For tests worth `6` points, $N \leq 8$.
- For tests worth `36` points, $N \leq 40$.
- For tests worth `51` points, $N \leq 200$.
- Any route is uniquely determined by the order in which cells are visited.
- We guarantee that for each test there is at least one route which follow the rules.
- The kangaroo can start jumping in any direction from `cs`.

Example

kangaroo.in	kangaroo.out	Note
4 2 3	2	The kangaroo starts from cell 2 and finishes in cell 3. The two correct routes are 2 -> 1 -> 4 -> 3 and 2 -> 4 -> 1 -> 3