Problem E. nanobots

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

small 8 points *2 minute timeout*	The contest is finished.
large 30 points *10 minute timeout*	The contest is finished.

Problem

Nanobots

A group of medical researchers is working on a new treatment to fight bacteria. In the treatment, special nanobots are transferred into the patient's body, where they locate and trap any harmful bacteria. This allows the patient to get better, and the researchers can later retrieve the trapped bacteria for further study.

Not any nanobot can trap any bacterium, though. A nanobot can be characterized by two traits: size and speed. A nanobot can only trap bacteria that are smaller than it (otherwise, the bacteria will not fit in the nanobot's cage) and also slower than it (otherwise, the bacteria can escape the nanobot). Formally, a nanobot with speed A and size B can trap a bacterium with speed C and size D if and only if A > C and B > D. The speeds and sizes of both nanobots and bacteria are in the inclusive range [1, `GetRange()`].

You have a group of nanobots and you want to know how effective they are at trapping bacteria. Your goal is to find how many of the `GetRange()`^{2} possible bacteria get trapped by the team of nanobots. Unfortunately, you cannot directly examine the speed and size of your nanobots. You can only experiment by introducing a bacterium with a specific size and speed, and watching whether this bacterium gets trapped by the team of nanobots or not. A team of nanobots will trap a bacterium with speed C and size D if and only if there is at least one nanobot in the team that has both speed strictly greater than C and size strictly greater than D.

You may carry out as many experiments of this sort as you want... within the allowed running time for the problem, of course! You can choose the speed and size of the bacteria in each experiment, and for each one, you receive one piece of data: whether or not the bacteria was trapped. Each experiment uses the full team of nanobots, and the same nanobot can catch bacteria in different experiments. Based on that information, you need to determine how many of the `GetRange()`^{2} possible bacteria would be trapped by the team of nanobots. (Because the speed can take any integer value in [1, `GetRange()`], and the same is true for the size, there are `GetRange()`^{2} possible bacteria.) Since the output can be a really big number, we only ask you to output the remainder of dividing the result by the prime 10^9+7 (1000000007).

Distributed Code Jam is not a licensed physician. Nothing in this problem statement should be construed as an attempt to offer medical advice. Distributed Code Jam is also not a licensed scientist. Nothing in this problem statement should be construed as an attempt to offer scientific advice. Using nanobots to fight harmful bacteria definitely sounds cool, though.

Input

The input library is called "nanobots"; see the sample inputs below for examples in your language. It defines three methods:

- **GetNumNanobots()**:
  - Takes no argument.
  - Returns a 64-bit integer: the number of nanobots on the team.
  - Expect each call to take 0.2 microseconds.
- **GetRange()**:
  - Takes no argument.
  - Returns a 64-bit integer: the maximum valid value for speeds and sizes of both bacteria and nanobots.
  - Expect each call to take 0.2 microseconds.
- **Experiment(c, d)**:
  - Takes exactly two 64-bit integer arguments: a size c and a speed d, 1 ≤ c,d ≤ `GetRange()`.
  - Returns a char: T if a bacteria with size c and speed d is trapped by the nanobots, or E if it escapes.
  - Expect each call to take 0.2 microseconds.

Output

Output one line with a single integer: how many of the different bacteria with both size and speed in the inclusive range [1, `GetRange()`] would be trapped by the nanobots, modulo the prime 10^9+7 (1000000007). Two bacteria are considered different if and only if they have different speed and/or different size.

Limits

Time limit: 18 seconds.
Memory limit per node: 256 MB.
Maximum number of messages a single node can send: 1000.
Maximum total size of messages a single node can send: 8 MB.
$2 \le$ GetRange() $\le 10^{12}$.
`Experiment(c, d)` is either uppercase T or uppercase E, for all c, d.
The results of `Experiment` are consistent with the same team of `GetNumNanobots()` nanobots across all nodes.
It is possible that multiple nanobots might have the same size and speed.

Small dataset

Number of nodes: 10.
$1 \le$ `GetNumNanobots()` $\le 10^5$.

Large dataset

Number of nodes: 100.
$1 \le$ `GetNumNanobots()` $\le 10^7$.

Sample

Input	Output

`See input files below.`	`For sample input 1:`
	`26`
	`For sample input 2:`
	`998496508`
	`For sample input 3:`
	`0`

Sample input libraries:
Sample input for test 1: nanobots.h [CPP] nanobots.java [Java]
Sample input for test 2: nanobots.h [CPP] nanobots.java [Java]
Sample input for test 3: nanobots.h [CPP] nanobots.java [Java]