Codeforces Round #439 (Div. 2)

A. The Artful Expedient

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Rock... Paper!

After Karen have found the deterministic winning (losing?) strategy for rock-paper-scissors, her brother, Koyomi, comes up with a new game as a substitute. The game works as follows.

A positive integer n is decided first. Both Koyomi and Karen independently choose n distinct positive integers, denoted by $x_1, x_2, ..., x_n$ and $y_1, y_2, ..., y_n$ respectively. They reveal their sequences, and repeat until **all of $2n$ integers become distinct**, which is the only final state to be kept and considered.

Then they count the number of ordered pairs (i, j) $(1 \le i, j \le n)$ such that the value x_i xor y_j equals to one of the $2n$ integers. Here xor means the bitwise exclusive or operation on two integers, and is denoted by operators ^ and/or xor in most programming languages.

Karen claims a win if the number of such pairs is even, and Koyomi does otherwise. And you're here to help determine the winner of their latest game.

Input

The first line of input contains a positive integer n $(1 \le n \le 2\ 000)$ — the length of both sequences.

The second line contains n space-separated integers $x_1, x_2, ..., x_n$ $(1 \le x_i \le 2 \cdot 10^6)$ — the integers finally chosen by Koyomi.

The third line contains n space-separated integers $y_1, y_2, ..., y_n$ $(1 \le y_i \le 2 \cdot 10^6)$ — the integers finally chosen by Karen.

Input guarantees that **the given $2n$ integers are pairwise distinct**, that is, no pair (i, j) $(1 \le i, j \le n)$ exists such that one of the following holds: $x_i = y_j$; $i \ne j$ and $x_i = x_j$; $i \ne j$ and $y_i = y_j$.

Output

Output one line — the name of the winner, that is, "Koyomi" or "Karen" (without quotes). Please be aware of the capitalization.

Examples

input

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

output

```
Karen
```

input

```
5
2 4 6 8 10
9 7 5 3 1
```

output

```
Karen
```

Note

In the first example, there are 6 pairs satisfying the constraint: $(1, 1)$, $(1, 2)$, $(2, 1)$, $(2, 3)$, $(3, 2)$ and $(3, 3)$. Thus, Karen wins since 6 is an even number.

In the second example, there are 16 such pairs, and Karen wins again.

B. The Eternal Immortality

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Even if the world is full of counterfeits, I still regard it as wonderful.

Pile up herbs and incense, and arise again from the flames and ashes of its predecessor — as is known to many, the phoenix does it like this.

The phoenix has a rather long lifespan, and reincarnates itself once every $a!$ years. Here $a!$ denotes the factorial of integer a, that is, $a! = 1 \times 2 \times ... \times a$. Specifically, $0! = 1$.

Koyomi doesn't care much about this, but before he gets into another mess with oddities, he is interested in the number of times the phoenix will reincarnate in a timespan of $b!$ years, that is, $\frac{b!}{a!}$. Note that when $b \geq a$ this value is always integer.

As the answer can be quite large, it would be enough for Koyomi just to know **the last digit of the answer in decimal representation**. And you're here to provide Koyomi with this knowledge.

Input

The first and only line of input contains two space-separated integers a and b ($0 \leq a \leq b \leq 10^{18}$).

Output

Output one line containing a single decimal digit — the last digit of the value that interests Koyomi.

Examples

input
```
2 4
```
output
```
2
```

input
```
0 10
```
output
```
0
```

input
```
107 109
```
output
```
2
```

Note

In the first example, the last digit of $\frac{4!}{2!} = 12$ is 2;

In the second example, the last digit of $\frac{10!}{0!} = 3628800$ is 0;

In the third example, the last digit of $\frac{109!}{107!} = 11772$ is 2.

C. The Intriguing Obsession

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

— This is not playing but duty as allies of justice, Nii-chan!

— Not allies but justice itself, Onii-chan!

With hands joined, go everywhere at a speed faster than our thoughts! This time, the Fire Sisters — Karen and Tsukihi — is heading for somewhere they've never reached — water-surrounded islands!

There are three clusters of islands, conveniently coloured red, blue and purple. The clusters consist of a, b and c distinct islands respectively.

Bridges have been built between some (possibly all or none) of the islands. A bridge bidirectionally connects two different islands and has length 1. For any two islands of the same colour, either they shouldn't be reached from each other through bridges, or the shortest distance between them is **at least 3**, apparently in order to prevent oddities from spreading quickly inside a cluster.

The Fire Sisters are ready for the unknown, but they'd also like to test your courage. And you're here to figure out the number of different ways to build all bridges under the constraints, and give the answer modulo $998\,244\,353$. Two ways are considered different if a pair of islands exist, such that there's a bridge between them in one of them, but not in the other.

Input

The first and only line of input contains three space-separated integers a, b and c ($1 \le a, b, c \le 5\,000$) — the number of islands in the red, blue and purple clusters, respectively.

Output

Output one line containing an integer — the number of different ways to build bridges, modulo $998\,244\,353$.

Examples

input
```
1 1 1
```
output
```
8
```

input
```
1 2 2
```
output
```
63
```

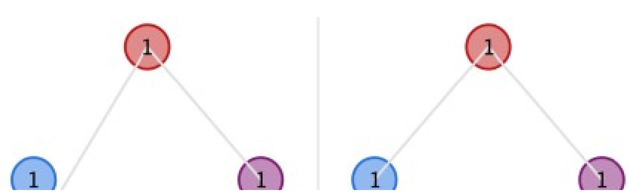
input
```
1 3 5
```
output
```
3264
```

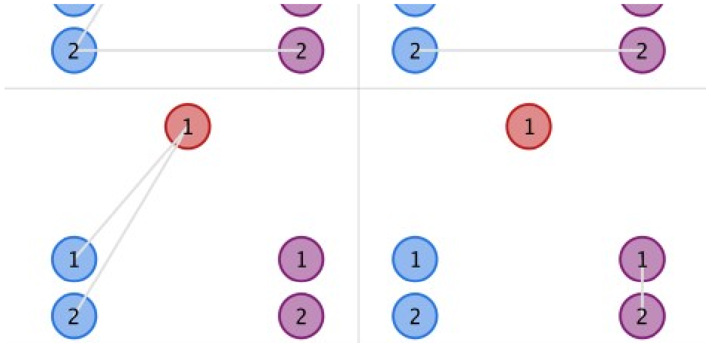
input
```
6 2 9
```
output
```
813023575
```

Note

In the first example, there are 3 bridges that can possibly be built, and no setup of bridges violates the restrictions. Thus the answer is $2^3 = 8$.

In the second example, the upper two structures in the figure below are instances of valid ones, while the lower two are invalid due to the blue and purple clusters, respectively.




2
2
2
2
1
1
1
1
2
2
1
1
1
2
2

D. The Overdosing Ubiquity

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

The fundamental prerequisite for justice is not to be correct, but to be strong. That's why justice is always the victor.

The Cinderswarm Bee. Koyomi knows it.

The bees, according to their nature, live in a tree. To be more specific, a *complete binary tree* with n nodes numbered from 1 to n. The node numbered 1 is the root, and the parent of the i-th $(2 \le i \le n)$ node is $\lfloor \frac{i}{2} \rfloor$. Note that, however, all edges in the tree are undirected.

Koyomi adds m extra undirected edges to the tree, creating more complication to trick the bees. And you're here to count the number of *simple paths* in the resulting graph, modulo $10^9 + 7$. A *simple path* is an alternating sequence of adjacent nodes and undirected edges, which begins and ends with nodes and does not contain any node more than once. Do note that a single node is also considered a valid *simple path* under this definition. Please refer to the examples and notes below for instances.

Input

The first line of input contains two space-separated integers n and m $(1 \le n \le 10^9, 0 \le m \le 4)$ — the number of nodes in the tree and the number of extra edges respectively.

The following m lines each contains two space-separated integers u and v $(1 \le u, v \le n, u \ne v)$ — describing an undirected extra edge whose endpoints are u and v.

Note that there may be multiple edges between nodes in the resulting graph.

Output

Output one integer — the number of *simple paths* in the resulting graph, modulo $10^9 + 7$.

Examples

input
```
3 0
```
output
```
9
```

input
```
3 1
2 3
```
output
```
15
```

input
```
2 4
1 2
2 1
1 2
2 1
```
output
```
12
```

Note

In the first example, the paths are: (1); (2); (3); $(1, 2)$; $(2, 1)$; $(1, 3)$; $(3, 1)$; $(2, 1, 3)$; $(3, 1, 2)$. (For the sake of clarity, the edges between nodes are omitted since there are no multiple edges in this case.)

In the second example, the paths are: (1); $(1, 2)$; $(1, 2, 3)$; $(1, 3)$; $(1, 3, 2)$; and similarly for paths starting with 2 and 3. $(5 \times 3 = 15$ paths in total.)

In the third example, the paths are: (1); (2); any undirected edge connecting the two nodes travelled in either direction. $(2 + 5 \times 2 = 12$ paths in total.)

E. The Untended Antiquity

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Adieu l'ami.

Koyomi is helping Oshino, an acquaintance of his, to take care of an open space around the abandoned Eikou Cram School building, Oshino's makeshift residence.

The space is represented by a rectangular grid of $n \times m$ cells, arranged into n rows and m columns. The c-th cell in the r-th row is denoted by (r, c).

Oshino places and removes barriers **around** rectangular areas of cells. Specifically, an action denoted by "$1\ r_1\ c_1\ r_2\ c_2$" means Oshino's placing barriers around a rectangle with two corners being (r_1, c_1) and (r_2, c_2) and sides parallel to squares sides. Similarly, "$2\ r_1\ c_1\ r_2\ c_2$" means Oshino's removing barriers around the rectangle. **Oshino ensures that no barriers staying on the ground share any common points, nor do they intersect with boundaries of the $n \times m$ area.**

Sometimes Koyomi tries to walk from one cell to another carefully without striding over barriers, in order to avoid damaging various items on the ground. "$3\ r_1\ c_1\ r_2\ c_2$" means that Koyomi tries to walk from (r_1, c_1) to (r_2, c_2) without crossing barriers.

And you're here to tell Koyomi the feasibility of each of his attempts.

Input

The first line of input contains three space-separated integers n, m and q $(1 \le n, m \le 2\,500,\ 1 \le q \le 100\,000)$ — the number of rows and columns in the grid, and the total number of Oshino and Koyomi's actions, respectively.

The following q lines each describes an action, containing five space-separated integers t, r_1, c_1, r_2, c_2 $(1 \le t \le 3,\ 1 \le r_1, r_2 \le n,\ 1 \le c_1, c_2 \le m)$ — the type and two coordinates of an action. Additionally, the following holds depending on the value of t:

- If $t = 1$: $2 \le r_1 \le r_2 \le n - 1$, $2 \le c_1 \le c_2 \le m - 1$;
- If $t = 2$: $2 \le r_1 \le r_2 \le n - 1$, $2 \le c_1 \le c_2 \le m - 1$, the specified group of barriers exist on the ground before the removal.
- If $t = 3$: no extra restrictions.

Output

For each of Koyomi's attempts (actions with $t = 3$), output one line — containing "`Yes`" (without quotes) if it's feasible, and "`No`" (without quotes) otherwise.

Examples

input

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

output

```
No
Yes
```

input

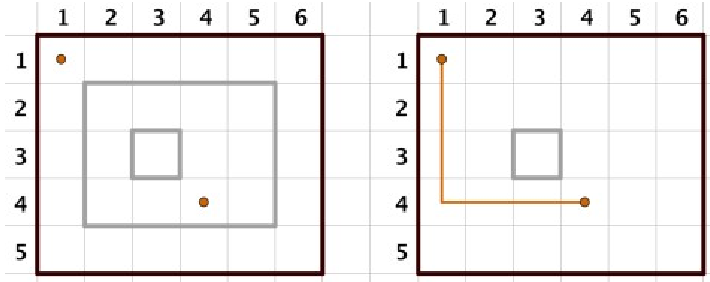
```
2500 2500 8
1 549 1279 1263 2189
1 303 795 1888 2432
1 2227 622 2418 1161
3 771 2492 1335 1433
1 2017 2100 2408 2160
3 48 60 798 729
1 347 708 1868 792
3 1940 2080 377 1546
```

output

```
No
Yes
No
```

Note

For the first example, the situations of Koyomi's actions are illustrated below.


1
2
3
4
5
6
1
2
3
4
5
1
2
3
4
5
6
1
2
3
4
5