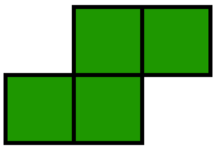
Problem A. Tetris Puzzle

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

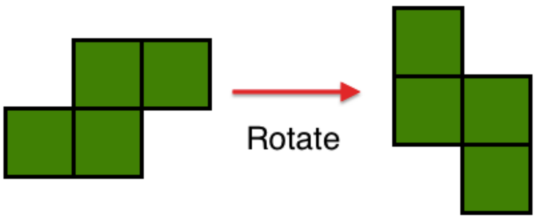
Teto has an infinite number of S-mino tiles. The following picture shows an S-mino tile:

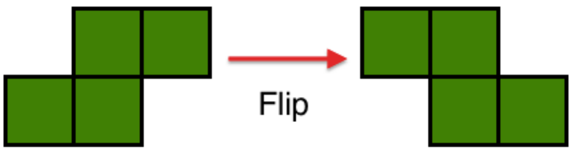


Initially, all tiles are arranged in this orientation. Teto wants to put some tiles on an $N \times N$ grid. The grid is initially empty. Also, Teto has a counter. Initially the couter is set to zero.

He can perform the following operations.

- Place an S-mino tile on the glid (the tile must completely fit within the grid). The tile must be aligned with the grid, and no two tiles can overlap.
- Rotate an S-mino tile clockwise by 90 degrees and increment the counter.
- Flip an S-mino tile with respect to the horizontal axis and increment the counter.
- All rotated tiles and flipped tiles must be placed on the grid.





You are given the state of the grid after Teto finishes placing tiles. For each (i, j), if $s_{i,j}$ is '`o`', the cell that is the i-th from the top and the j-th from the left is filled by one of the tiles. Otherwise, $s_{i,j}$ is '`.`', and this cell is empty.

Compute the parity of Teto's counter. It is guaranteed that the parity can be uniquely determined.

Input

N
$s_{1,1} \dots s_{1,N}$
$\vdots$
$s_{N,1} \dots s_{N,N}$

- $1 \le N \le 50$
- For each (i, j), $s_{i,j}$ is either '`o`' or '`.`'.

Output

Print the parity (0 or 1) of the counter.

Example

standard input	standard output
6 .oooo. oooooo oooooo oooooo oooooo .oooo.	0

Problem B. Shift and Paint

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 mebibytes

N balls are arranged in a row. You can perform an operation called *Shift*:

Shift Operation. Choose L consecutive balls, and move the rightmost chosen ball to the left of the leftmost chosen ball. In other words, if you choose balls at positions $i, i+1, \ldots, i+L-1$, these balls will go to positions $i+L-1, i, i+1, \ldots, i+L-2$ after the operation.

You want to color these balls with K colors. How many ways are there to color the balls? Two ways of coloring are considered equivalent if you can reach from one coloring to the other by repeating the *Shift* operations zero or more times. Compute the answer modulo 10^9+7.

Input

$N \ K \ L$

- $2 \le N \le 10^6$
- $1 \le K \le 10^6$
- $2 \le L \le N$

Output

Print the number of ways to color the balls, modulo 10^9+7.

Example

standard input	standard output
3 3 3	11
3 3 2	10

Note

In Sample 1, there are 11 ways to paint the balls:

$(1,1,1), (1,1,2), (1,1,3), (1,2,2), (1,2,3), (1,3,2), (1,3,3), (2,2,2), (2,2,3), (2,3,3)$, and $(3,3,3)$.

In Sample 2, here are 10 ways to paint the balls:

$(1,1,1), (1,1,2), (1,1,3), (1,2,2), (1,2,3), (1,3,3), (2,2,2), (2,2,3), (2,3,3)$, and $(3,3,3)$.

Problem C. Pianist

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

A pianist wants to play the piano.

The piano has 10^{100} keys, and the keys are labeled $A, B, C, D, E, F, G, A, B, \cdots$ from the left to the right. The pianist wants to satisfy the following constraints:

- He chooses an A key, and starts playing the piano by pressing the chosen key.
- After he presses the i-th key, he can press either the $i-1$-th key or the $i+1$-th key in the next step.
- After he presses an A key, he can also stop playing the piano.
- The first A key and the last A key he plays are not necessarily the same.
- He must press A keys (it doesn't matter which A keys) exactly C_1 times.
- Similarly, he must press B, C, D, E, F, G keys exactly $C_2, \ldots, C_7$ times, respectively.

Determine if it is possible to satisfy all the constraints above.

Input

C_1 C_2 C_3 C_4 C_5 C_6 C_7

- $0 \le C_i \le 10^{10}$
- $\sum C_i > 0$

Output

Print "`YES`" if it is possible to satisfy all the constraints, and "`NO`" otherwise.

Examples

standard input	standard output
2 1 1 1 1 1 1	YES
1 1 1 1 1 1 1	NO
3 1 0 10000000000 10000000000 0 1	NO
1 0 0 0 0 0 0	YES

Note

In Sample 1, he can play A, B, C, D, E, F, G, A.

Problem D. Driving

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

There are N cities and M bidirectional roads. The cities are numbered 1 through N, and the roads are numbered 1 through M. The road i connects cities A_i and B_i, and its length is 2^i. It is guaranteed that you can reach from any city to any city using these roads.

Snuke wants to go driving. He wants to start from city 1, pass through each road at least once, and return to city 1. Compute the shortest possible length of his route, modulo $10^9 + 7$.

Input

$$
\begin{array}{l}
N \;\; M \\
A_1 \;\; B_1 \\
\vdots \\
A_M \;\; B_M
\end{array}
$$

- $2 \le N \le 400000$
- $1 \le M \le 500000$
- $1 \le A_i, B_i \le N$
- $A_i \ne B_i$
- It is possible to reach from any city to any city using roads.
- No two roads connect the same pair of cities.

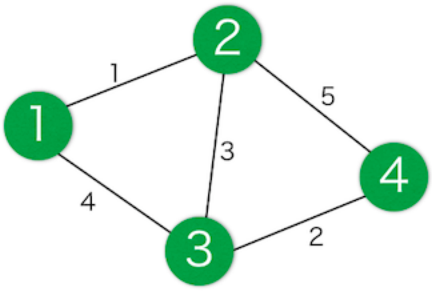
Output

Print the shortest possible length of Snuke's route, modulo $10^9 + 7$.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>4 5
1 2
3 4
2 3
1 3
2 4</pre></td><td><pre>70</pre></td></tr>
<tr><td><pre>6 10
4 6
4 5
3 6
5 2
3 2
1 2
3 4
6 1
2 4
1 3</pre></td><td><pre>2132</pre></td></tr>
</table>

Note



For example, in sample 1 one optimal route is $1 \to 2 \to 3 \to 4 \to 2 \to 3 \to 1$.

Problem E. Coins

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Niwango and Nikomoba play the following game.

N coins are arranged in a row, and they are numbered 1 through N. Initially, odd-indexed coins are heads up, and even-indexed coins are tails up. The value of the coin i is S_i.

The game consists of $N-1$ turns. The turns are numbered 1 through $N-1$. Niwango plays odd-indexed turns, and Nikomoba plays even-indexed turns. In the turn i, the player can flip at most one of the coins i and $i+1$. (The player is allowed not to flip any coins).

After the $N-1$ turns, Niwango takes all heads-up coins and Nikomoba takes all tails-up coins. The score of a player is the sum of values of all the coins he takes. Compute Niwango's score when both players play optimally.

Also, there will be Q updates of coin values. In the i-th update, the value of coin P_i decreases by D_i. This update applies for all later updates; for example, after the second update, both of the first two updates are applied. For each i, compute Niwango's score, assuming that they start playing the game after the i-th update.

Input

N
$S_1 \ \dots \ S_N$
Q
$P_1 \ \ D_1$
$\vdots$
$P_Q \ \ D_Q$

- $2 \leq N \leq 200000$
- $1 \leq S_i \leq 10^9$
- $0 \leq Q \leq 200000$
- $1 \leq P_i \leq N$
- $1 \leq D_i$
- At any time, each coin has a positive value.

Output

Print $Q+1$ lines. In the i-th line print Niwango's score when they start playing the game after $(i-1)$ updates of coin values.

Examples

standard input	standard output
4 1 2 3 4 1 4 3	7 5
8 3 1 4 1 5 9 2 6 5 3 3 6 6 8 4 1 1 6 2	19 16 16 12 11 11

Problem F. Forbidden Puzzle

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

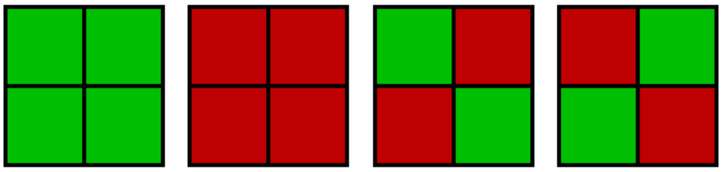
In this task you are asked to determine if a puzzle has at least one valid solution.

The puzzle contains an $N \times N$ grid. Initially, $4N - 4$ cells on the edge are colored either red or green, and the other cells are colored white. Each segment that divides two adjacent white cells are colored either black or gray.

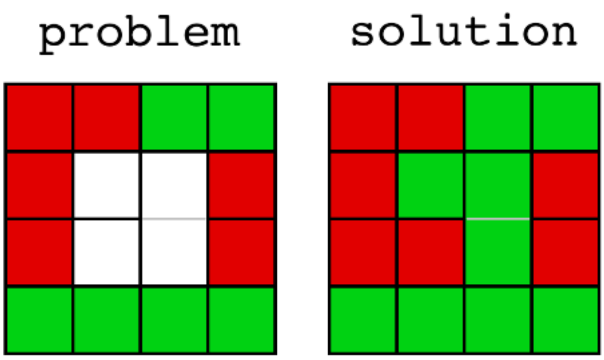
In order to solve this puzzle, you must color each white cell with red or green such that:

- Two adjacent cells that are separated by a gray line must be colored with the same color.
- The final grid must not contain "forbidden patterns".

There are four forbidden patterns. They are shown in the picture below:



The following picture shows one possible problem of the puzzle and a valid solution:



You are given the information of the puzzle as follows. The cell that is the i-th from the top and the j-th from the left is labelled as (i, j). For each edge cell (i, j), $s_{i,j}$ is either 'o' or 'x', and these characters represent green and red, respectively. For each non-edge cell (i, j), $s_{i,j}$ is an uppercase letter. If two adjacent non-edge cells are labeled with the same character, they are separated by a gray line. Otherwise they are separated by a black line.

Determine whether the given puzzle has at least one solution.

Input

N
$s_{1,1} \dots s_{1,N}$
$\vdots$
$s_{N,1} \dots s_{N,N}$

- $3 \le N \le 30$
- If (i, j) is one of the $4N - 4$ edge cells, $s_{i,j}$ is either 'o' or 'x'.

- If (i, j) is not an edge cell, $s_{i,j}$ is an uppercase letter.

Output

If the puzzle has at least one valid solution, print "YES". Otherwise print "NO".

Example

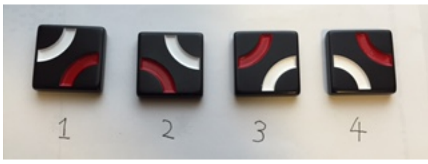
standard input	standard output
5 oooxx oBABx oABAx xBABo xooox	NO

Problem G. TRAX

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Snuke plays with the following tile (this tile is used in a game called TRAX):

Snuke wants to fill an $H \times W$ grid with these tiles. In each cell of the grid, he places a tile in one of the four orientations:



These orientations are numbered 1 through 4 as in the picture above.

The placement of tiles must satisfy the following constraints:

- A red arc must not touch a white arc.
- The grid must not contain cycles (see the examples below).
- For each $i (1 \le i \le N)$, in the cell that is R_i-th from the top and C_i-th from the left, the orientation of the tile must be D_i.

For example, the following picture shows two invalid placements of tiles. The placement on the left contains a white cycle, and the placement on the right contains a red cycle and a white large cycle.

Count the number of valid placements, modulo $10^9 + 7$.

Input

H W
N
R_1 C_1 D_1
$\vdots$
R_N C_N D_N

- $1 \le H, W \le 10^5$
- $0 \le N \le 10^5$

- $1 \leq R_i \leq H$
- $1 \leq C_i \leq W$
- $1 \leq D_i \leq 4$
- (R_i, C_i) are pairwise distinct.

Output

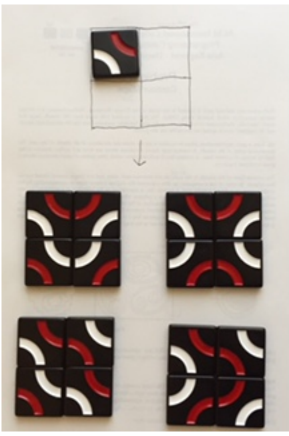
Print the number of valid placements, modulo $10^9 + 7$.

Examples

standard input	standard output
2 2 1 1 1 4	4
2 10 2 1 1 1 1 2 1	0
2015 1114 0	711460824
2 2 2 1 1 1 2 2 3	0
5 6 3 1 2 2 4 1 1 5 6 4	12
5 6 2 3 3 4 3 4 2	39

Note

For sample 1, these are the four valid placements:



Problem H. Pyramid Decoration

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

There are $L \times (L+1) \times (L+2)/6$ stones. Snuke stacked these stones like a tetrahedron (a 3D object that cosists of four triangular faces). The position of a stone is represented by a tuple of integers (i, j, k) $(1 \le k \le j \le i \le L)$. When $i < L$, the stone at (i, j, k) is stacked on top of three stones at $(i+1, j, k)$, $(i+1, j+1, k)$, and $(i+1, j+1, k+1)$.

Snuke colors these stones in the following way. First, he colors all stones in the bottommost layer. For each i $(1 \le i \le N)$, he colors the stone at (L, A_i, B_i) black, and colors the other stones on the bottommost layer white. Then, he colors the other stones from bottom to top. He colors the stone at (i, j, k) black if the number of black stones below it ($(i+1, j, k)$, $(i+1, j+1, k)$, and $(i+1, j+1, k+1)$) is either zero or two. Otherwise he paints this stone white.

Determine the color of the topmost stone.

Input

L N
A_1 B_1
$\vdots$
A_N B_N

- $2 \le L \le 10^9$
- $0 \le N \le min(1000, L \times (L+1)/2)$
- $1 \le B_i \le A_i \le L$
- The tuples (A_i, B_i) are pairwise distinct.

Output

Print "`Iori`" if the topmost stone is black. Otherwise print "`Yayoi`".

Example

standard input	standard output
2 2 2 1 1 1	Iori
3 0	Yayoi

Problem I. 2D Pyramid

Input file:	*standard input*
Output file:	*standard output*
Time limit:	6 seconds
Memory limit:	512 mebibytes

There is a pyramid of stones. The pyramid consists of 10^6 layers, and the i-th layer ($1 \le i \le 10^6$) contains i stones. The j-th stone in the i-th layer is labeled (i, j).

In order to take the stone (i, j) from the pyramid, you must take the stones $(i-1, j-1)$ and $(i-1, j)$ first (if such stones exist).

You want to take two stones (A, B) and (C, D). You also want to do this by taking the minimum possible number of stones. In how many ways can you do this? Two ways are considered different if for some i, the i-th stone you take from the pyramid is different.

You are given T queries. The parameters for the i-th query are given by (A_i, B_i, C_i, D_i). For each query, compute the answer modulo $10^9 + 7$.

Input

T
$A_1 \;\; B_1 \;\; C_1 \;\; D_1$
$\vdots$
$A_T \;\; B_T \;\; C_T \;\; D_T$

- $1 \le T \le 300000$
- $1 \le B_i \le A_i \le 10^6$
- $1 \le D_i \le C_i \le 10^6$
- It is guaranteed that you can take both (A_i, B_i) and (C_i, D_i) by taking at most 10^6 stones.
- For each i, (A_i, B_i) and (C_i, D_i) are distinct.

Output

Print T lines. In the i-th line, print the answer of the i-th query.

Example

standard input	standard output
6 2 1 2 2 1 1 1000000 1000000 3 2 5 3 5 2 4 3 2015 55 1700 1300 100 50 1000 500	2 1 42 252 926737422 143485143

Note

For the first query, there are two ways to take the stones:

- Take stones in the order $(1, 1), (2, 2), (2, 1)$.
- Take stones in the order $(1, 1), (2, 1), (2, 2)$.

Problem J. Foot Game

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

A group of X people wants to play the following game.

- There are N buttons numbered 1 through N.
- In order to press a button, one person must stand on the button.
- A person can't press multiple buttons simultaneously.
- If they keep pressing the button i during the time interval $[S_i, T_i)$, they score a point. Note that the button is not necessarily pressed by the same person during this time interval.
- They can move between buttons instantly; for example, a person can press a button during the time interval $[1, 2)$, and press another button during the time interval $[2, 3)$.

Compute the minimum possible X that enables them to score at least $N - 1$ points.

Input

N
$S_1 \ T_1$
$\vdots$
$S_N \ T_N$

- $2 \le N \le 10^5$
- $1 \le S_i < T_i \le 10^5$

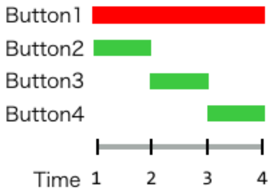
Output

Print the minimum possible X that enables them to score at least $N - 1$ points.

Examples

standard input	standard output
4 1 4 1 2 2 3 3 4	1
5 5 11 2 4 3 4 2 7 5 7	2
4 1 2 1 2015 2015 100000 99999 100000	2

Note



In Sample 1, one person can press buttons 2, 3, and 4, and score three points.

Problem K. Pyramid Game

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

There are N piles of stones. The i-th (1-based) pile contains A_i stones.

Iori and Yayoi play a game with these piles. Iori takes the first turn, and they take turns alternately.

In each turn, the player can perform one of the following operations:

- Remove a stone from a pile.
- Remove one stone each from all the N piles. This operation is possible only when all piles contain at least one stone.

If a player can't perform any operations in her turn, she loses. Determine the winner of the game when they both play optimally.

Input

N
$A_1 \ A_2 \cdots \ A_N$

- $1 \le N \le 50$
- $1 \le A_i \le 50$

Output

Print the name of the winner ("`Iori`" or "`Yayoi`") when they play optimally.

Example

standard input	standard output
2 1 1	Iori
1 50	Yayoi

Note

In Sample 1, Iori can take both stones in her first turn.