Problem A. Building

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

There are N cities and $N-1$ bidirectional roads. They form a tree. Each city contains a building, and the height of the building in city i is H_i.

You decided to choose a simple path in the tree, choose a (not necessarily continuous) subsequence of cities on this path $i_1, i_2, \ldots, i_k$, and decorate bulidings in them. Additionally, they must satisfy the following constraints:

- The cities $i_1, i_2, \ldots, i_k$ must appear in the path in this relative order.
- $H_{i_1} < H_{i_2} < \cdots < H_{i_k}$

Compute the maximum possible number of buildings you can decorate.

Input

The 1-st line contains an integer N ($2 \leq N \leq 100\,000$).

The following N lines describe buildings. The i-th of them contains an integer H_i ($1 \leq H_i \leq 1\,000\,000\,000$).

The following $N-1$ lines describe roads. The i-th of them contains two integers A_i, B_i ($1 \leq A_i < B_i \leq N$) separated by a space, the two endpoints of the i-th road.

Output

Print the answer.

Example

standard input	standard output
7 4 2 5 3 1 8 7 1 2 2 3 3 4 4 5 3 6 6 7	4

Problem B. Fish

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	64 mebibytes

There are N fishes. The length of the i-th fish is L_i, and its color is C_i. Here, C_i is one of 'R', 'G', 'B', and they mean red, green, and blue, respectively.

Mr. JOI wants to buy a non-empty subset of the fishes. However, he is not allowed to buy two fishes X and Y at the same time, such that the length of X is at least twice as long as the length of Y.

Compute the number of triplets (r, g, b), such that it is possible to buy exactly r red fishes, g green fishes, and b blue fishes.

Input

The first line contains an integer N ($1 \le N \le 500\,000$). The $1+i$-th ($1 \le i \le N$) line contains an integer L_i and a character C_i, separated by a space ($1 \le L_i \le 1\,000\,000\,000$).

Output

Print the answer.

Example

standard input	standard output
4 10 R 4 G 8 B 5 B	6
10 26 B 10 B 16 G 20 R 6 R 5 G 13 G 40 R 8 R 33 R	13

Note

In the Sample 1

- It is not allowed to buy the 1st and the 2nd fishes simultaneously.
- It is not allowed to buy the 1st and the 4th fishes simultaneously.
- It is not allowed to buy the 2nd and the 3rd fishes simultaneously.

The possible triplets are $(r, g, b) = (1, 0, 0), (0, 1, 0), (0, 0, 1), (1, 0, 1), (0, 1, 1)$, and $(0, 0, 2)$.

Problem C. JOI Flag

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

We define level-K **JOI Flag** as follows:

- A level-0 **JOI Flag** is a 1×1 grid that contains one of three letters '`J`', '`O`', or '`I`'.
- For an integer $m > 0$, a level-m **JOI Flag** is a $2^m \times 2^m$ grid that can be divided into the following four $2^{m-1} \times 2^{m-1}$ grid: level-$m-1$ **JOI Flag**, a $2^{m-1} \times 2^{m-1}$ grid that consists of only '`J`', a $2^{m-1} \times 2^{m-1}$ grid that consists of only '`O`', and a $2^{m-1} \times 2^{m-1}$ grid that consists of only '`I`'.

For example, here is an example of level-2 **JOI Flag**:

```
OIJJ
JJJJ
OOII
OOII
```

And here is an example of level-3 **JOI Flag**:

```
IIIIIIOO
IIIIIIOO
IIIIJOJJ
IIIIOIJJ
JJJJOOOO
JJJJOOOO
JJJJOOOO
JJJJOOOO
```

You have a $2^K \times 2^K$ grid. Initially, N cells contain letters: the cell in the X_i-th column from left and the Y_i-th row from top contains a letter C_i. Other cells are empty.

Some cells may contain letters '`J`', '`O`', and '`I`'.

You want to convert it to a level-K **JOI Flag** by writing and modifying letters. You can freely write a letter into an empty cell, but it costs you a unit money to modify a cell that already contains a letter.

Compute the minimum cost required to convert it to a level-K **JOI Flag**.

Input

The first line contains two integers K, N, separated by a space ($1 \le K \le 30$, $1 \le N \le 1\,000$). The following N lines describe letters. The $i+1$-th line contains two integers X_i, Y_i and a letter C_i, separated by spaces ($1 \le X_i \le 2^K$, $1 \le Y_i \le 2^K$, C_i is one of '`J`', '`O`', '`I`', the pairs (X_i, Y_i) are pairwise distinct).

Output

Print the answer.

Examples

standard input	standard output
2 10 2 2 J 3 3 I 1 3 I 1 1 O 3 2 J 2 1 I 4 1 O 3 4 I 4 4 O 2 3 O	3
4 30 16 14 J 2 8 O 10 9 J 10 13 I 6 6 O 11 14 I 1 2 I 3 2 O 3 10 O 1 12 I 4 11 I 9 5 J 15 1 O 12 4 I 16 5 J 10 7 J 3 8 J 4 10 I 4 7 I 2 11 I 2 12 O 15 5 J 15 7 J 6 9 J 5 7 O 14 5 J 12 11 J 15 10 O 13 16 I 13 11 I	9

Note

In the first sample, the input is:

```
OI-O
-JJ-
IOI-
--IO
```

it can be modified to the following with 3 units of money:

```
OIJJ
JJJJ
OOII
OOII
```

Problem D. Constellation

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

You are given a photo of N stars. The coordinates of the i-th star in the photo is (X_i, Y_i).

You know the following information:

- Each star in the photo is either in constellation A or in constellation B.
- All stars in constellation A and or constellation B are in the photo.
- For some stars, you know their constellation. You are given $C_1, \ldots, C_N$. If $C_i = 0$, you don't know the constellation of the i-th star. If $C_i = 1$, you are sure that the i-th star is in constellation A. If $C_i = 2$, you are sure that the i-th star is in constellation B.
- Each constellation is a non-empty set of stars.

Furthermore, you know that you can add some line segments between two stars in the same constellation, and satisfy the following:

- All stars in the same constellation are connected by the segments, directly or indirectly.
- A segment that connects two stars in constellation A and a segment that connects two stars in constellation B never intersects.

How many possibilities for the set of stars in constellations A and B are there? Compute it modulo $1\,000\,000\,007\,(= 10^9 + 7)$.

Input

The 1-st line contains an integer N $(2 \le N \le 100\,000)$. The $i+1$-th line $(1 \le i \le N)$ contains three integers X_i, Y_i, C_i, separated by spaces $(0 \le X_i \le 10^9, 0 \le Y_i \le 10^9,\ 0 \le C_i \le 2$, no three points are collinear).

Output

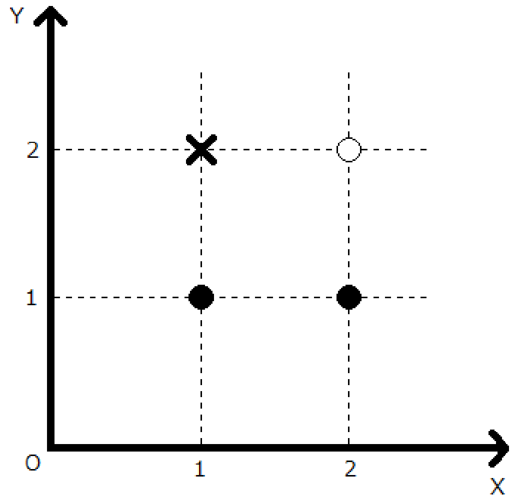
Print the answer.

Example

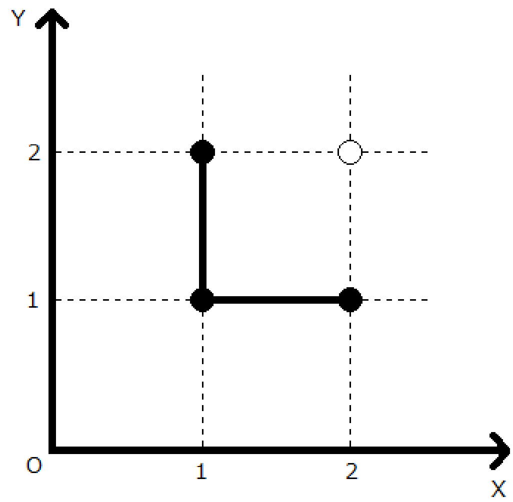
standard input	standard output
4 1 1 1 2 1 1 1 2 0 2 2 2	2

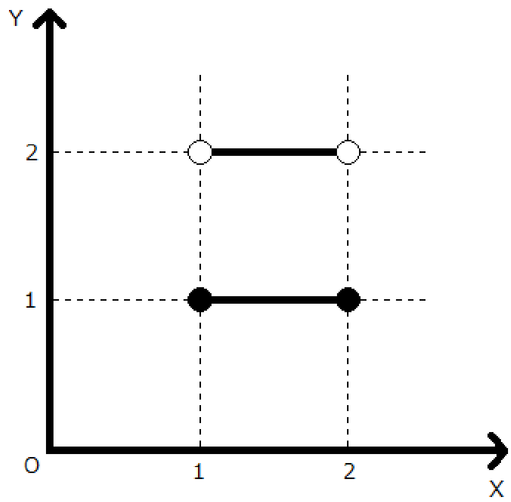
Note

Black represents A, white represents B, and $\times$ represents an unknown star.



In this example, $\times$ can be both A and B. For example, you can add segments as follows:





Problem E. Rotate

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	64 mebibytes

There is an $N \times N$ grid. We call the cell in the i-th row from top and the j-th column from left (i, j). Each cell contains a lowercase letter. Initially, the cell (i, j) contains A_{ij}.

You are asked to perform Q operations on this grid. In the k-th operation, choose an $S_k \times S_k$ subsquare of the grid whose upper-left corner is the cell (I_k, J_k), and rotate it counter-clockwise by 90 degrees.

Compute the final state of the grid after you perform all operations.

Input

The first line contains two integers N, Q, separated by a space ($2 \le N \le 1\,000$, $1 \le Q \le 2\,000$). The following N lines describe the grid. The j-th character of the i-th line among them is A_{ij}. The following Q lines describe operations. The k-th line among them contains three integers I_k, J_k, S_k ($1 \le I_k \le N - S_k + 1$, $1 \le J_k \le N - S_k + 1$, $2 \le S_k \le N$), separated by spaces.

Output

Print the answer in N lines, in the same format as the input.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td>4 1
abcd
efgh
ijkl
mnop
2 2 2</td><td>abcd
egkh
ifjl
mnop</td></tr>
</table>

Note

You rotate the following part counter-clockwise by 90 degrees:

```
fg
jk
```

Problem F. Fortune Telling

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	64 mebibytes

$M \times N$ white cards are arranged in M rows and N columns. Let (a, b) be the card in the i-th row from top, j-th column from left.

For each $i = 1, \cdots, K$, you do the following: for each (a, b) such that $A_i \le a \le B_i$ and $C_i \le b \le D_i$, invert the color (white to black, black to white) of (a, b).

After the operations, how many cards are white?

Input

The 1-st line contains three integers M, N, K, separated by a space ($1 \le M \le 10^9$, $1 \le N \le 10^9$, $1 \le K \le 10^5$). The $(i+1)$-th line ($1 \le i \le K$) contains 4 integers A_i, B_i, C_i, D_i ($1 \le A_i \le B_i \le M$, $1 \le C_i \le D_i \le N$), separated by spaces.

Output

Print the answer.

Example

standard input	standard output
6 5 3 2 4 1 4 4 6 3 5 1 2 3 5	11

Problem G. Kangaroo

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

There are N kangaroos numbered $1, 2, \cdots, N$. The size of the i-th kangaroo is A_i, and its pocket size is B_i. It is always that $A_i > B_i$.

Initially, no kangaroo contains other kangaroos in its pocket. They must repeat the following operations **while at least one possible operation exists**:

Choose a pair of kangaroos (i, j) such that:

- $A_i < B_j$
- Kangaroo i is not in other kangaroo's pocket.
- Kangaroo j doesn't contain any other kangaroos in its pocket.

When multiple such pairs exist, they can choose any. Then, put kangaroo i into kangaroo j's pocket (here, in case kangaroo i contains other kangaroos directly or indirectly, they will move together with kangaroo i).

Compute the number of possible final states, modulo $1\,000\,000\,007\,(= 10^9 + 7)$.

Input

The 1-st line contains an integer N ($1 \le N \le 300$). The $i + 1$-th line ($1 \le i \le N$) contains two integers A_i, B_i separated by a space ($1 \le B_i < A_i \le 1\ 000\ 000\ 000$).

Output

Print the answer.

Example

standard input	standard output
5 4 3 3 1 6 5 2 1 4 2	4
20 7 6 7 3 10 1 7 2 10 7 10 7 8 6 3 2 5 4 7 2 3 2 10 9 9 4 7 2 8 6 5 4 8 6 7 4 10 5 9 3	21060

Note

For Sample 1, there are four possible final states:

- Kangaroo 4 is in kangaroo 3's pocket.
- Kangaroo 4 is in kangaroo 1's pocket, and kangaroo 1 is in kangaroo 3's pocket.
- Kangaroo 4 is in kangaroo 1's pocket, and kangaroo 2 is in kangaroo 3's pocket.
- Kangaroo 4 is in kangaroo 1's pocket, and kangaroo 5 is in kangaroo 3's pocket.

Problem H. Sokoban

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

In Sokoban puzzle, you are given an $M \times N$ grid. One cell contains a box, and your objective is to move the box to a goal by moving a player in the grid. Some cells in the grid are walls, and each of the player, the box, the goal are always in one cell of the grid without walls. (The player and the box never go out of the grid).

You can perform the following operations:

- Move the player to one of its 4-adjacent cells that doesn't contain walls nor the box.
- If the player and the box are adjacent, and the cell that is adjacent to the box in the opposite direction to the player is a cell without a wall, the player can "push" the box: move the box to the cell in the opposite direction to the player and the player moves to the cell that previously contained the box.

Here are two examples of Sokoban problems. (`#` are walls, `@` is a player, `O` is a box, `X` is a goal, and `.` are other cells.)

```
..#@.
.X.O.
##..#
```

This problem can be solved as follows:

1. Move the player to right.
2. Move the player to down.
3. The player pushes the box to left.
4. The player pushes the box to left.

The following problem can't be solved:

```
..#..
.X.O.
##.@#
```

You are given a grid with walls and a goal. You want to put a player and a box in the grid and construct a solvable problem. Here, they must be put in different cells without walls and the goal. Count the number of ways to do so.

Input

The 1-st line contains two integers M, N, separated by a space ($1 \leq M, N \leq 1\,000$). The following M lines describe the grid. Each line contains N characters. Each character is one of '`#`' (wall), '`X`' (goal), or '`.`' (other cells, possibly an initial place for a player or a box).

The grid contains exactly one '`X`'.

Output

Print the answer.

Examples

standard input	standard output

`3 5`
`..#..`
`.X...`
`##..#`
	`9`

`2 3`
`.X.`
`...`
	`0`

`4 7`
`.#.#.##`
`##.#..#`
`....X..`
`##.#...`
	`24`

Note

In Sample 1, the following 9 problems are possible:

```
..#@.   ..#.@   ..#..   ..#..   ..#..   ..#..   ..#@.   ..#.@   ..#..
.XO..   .XO..   .XO@.   .XO.@   .XO..   .XO..   .X.O.   .X.O.   .X.O@
##..#   ##..#   ##..#   ##..#   ##@.#   ##.@#   ##..#   ##..#   ##..#
```

Problem I. Chinese

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	64 mebibytes

In a Chinese restaurant, N people are sitting around a round table equidistantly, numbered 1 through N in counter-clockwise order. Also, there are N dishes. For each i $(1 \le i \le N)$, the dish i is put in front of the person i.

The dishes are put on a rotatable machine. With cost c, it is possible to rotate the positions of all dishes at once by $(360/N)c$ degrees, either clockwise or counter-clockwise. For example, from the initial position, if you spend cost 1 and rotate the dishes counter-clockwise, dish N will come in front of person 1, and for each i $(2 \le i \le N)$ dish $(i-1)$ will come in front of person i.

Person 1 wants to eat dish k, and for each i $(2 \le i \le N)$ person i wants to eat dish A_i. The following constraints must be satisfied:

- When a person eats a dish the dish must be in front of the person. (Otherwise, they must rotate the dishes first.)
- Person 1 must eat first.
- Other people can eat in any order.

You don't know the value of k. For each k, compute the minimum possible total cost.

Input

The 1-st line contains an integer N $(2 \le N \le 10^5)$. The i-th $(2 \le i \le N)$ line contains an integer A_i $(1 \le A_i \le N)$.

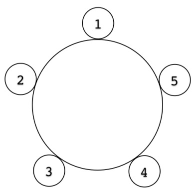
Output

Print N lines. The k-th line $(1 \le k \le N)$ should correspond to the answer when the 1-st person wants to eat the k-th dish.

Example

standard input	standard output
5 3 5 3 2	4 4 5 6 4

Note



For example, when $k = 3$, one optimal solution is as follows:

- Person 1 rotates the dishes clockwise with cost 2 and eats dish 3.

- Person 3 eats dish 5.
- Person 5 eats dish 2.
- Person 2 rotates the dishes counter-clockwise with cost 1 and eats dish 3.
- Person 4 rotates the dishes counter-clockwise with cost 2 and eats dish 3.

Problem J. Invitation

Input file:	*standard input*
Output file:	*standard output*
Time limit:	7 seconds
Memory limit:	128 mebibytes

There are A dogs (numbered $1, 2, \ldots, A$) and B cats (numbered $1, 2, \ldots, B$). You want to invite all of them to a party.

There are N groups of dogs and cats. The i-th groups contains $Q_i - P_i + 1$ dogs between P_i-th and Q_i-th, inclusive, and $S_i - R_i + 1$ cats between R_i-th and S_i-th. Also, it has a parameter T_i. The groups are not necessarily disjoint. Some animals may not be in any group.

You have already invited dog C. You want to invite the remaining dogs and cats by repeating the following process:

- If all dogs and cats are already invited, the invitation succeeds and you terminate the process.
- For each animal that hasn't been invited yet, compute its **happiness**. The happiness of an animal X is the maximum t such that there is a group with a parameter t that contains both X and at least one of invited animals. If such t doesn't exist, the happiness is defined as 0.
- Choose an animal with the maximum happiness. In case of tie, you prefer dogs, and if there are still ties, you choose the one with the smallest index.
- In case the chosen animal has the happiness 0, the invitation fails and you terminate the process. Otherwise, you invite the chosen animal.

Will the invitation succeed? If it will succeed, print the sum of happiness values of chosen animals at each step. Otherwise, print -1.

Input

The 1-st line contains three integers A, B, C ($1 \leq A, B \leq 1\,000\,000\,000$, $1 \leq C \leq A$), separated by spaces. The 2-nd line contains an integer N ($1 \leq N \leq 100\,000$). The $2 + i$-th line ($1 \leq i \leq N$) contains five integers P_i, Q_i, R_i, S_i, T_i ($1 \leq P_i \leq Q_i \leq A$, $1 \leq R_i \leq S_i \leq B$, $1 \leq T_i \leq 1\,000\,000\,000$), separated by spaces.

Output

Print the answer.

Examples

standard input	standard output
5 6 3 4 2 4 1 3 20 1 2 2 4 40 4 5 2 3 30 4 4 4 6 10	280
10 10 1 2 1 5 1 5 3 6 10 6 10 4	-1

Note

In Sample 1:

- You initially start with a dog 3.
- In the first step, the animals have the following happiness. Dog 2: 20, Dog 4: 20, Cat 1: 20, Cat 2: 20, Cat 3: 20. You invite Dog 2.
- In the second step, Dog 1: 40, Dog 4: 20, Cat 1: 20, Cat 2: 40, Cat 3: 40, Cat 4: 40. You invite Dog 3.
- And so on.

Type	Index	Happiness (when invited)
Dog	3	—
Dog	2	20
Dog	1	40
Cat	2	40
Cat	3	40
Cat	4	40
Dog	4	30
Dog	5	30
Cat	1	20
Cat	5	10
Cat	6	10

In Sample 2, you can invite dogs $1, 2, 3, 4, 5$ and cats $1, 2, 3, 4, 5$, but fail to invite dog 6.