Problem A. Matryoshka

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

N matryoshka dolls are sold in a shop. They are numbered 1 through N. The i-th matryoshka doll can be considered as a cylinder with radius R_i and height H_i.

The doll i can be put inside the doll j if both $R_i < R_j$ and $H_i < H_j$ are satisfied. The dolls can be nested: for example, the doll i can be put inside the doll j, and at the same time the doll j can be put inside the doll k (if $R_i < R_j < R_k$ and $H_i < H_j < H_k$). However, it is not allowed to put multiple dolls directly inside the same doll.

You are given Q queries. In the i-th query, suppose that you buy all the dolls whose radii are **at least** A_i and whose heights are **at most** B_i. Compute the minimum possible number of dolls that are not contained in any other dolls when you nest all the dolls you buy in the optimal way.

Input

$N \;\; Q$
$R_1 \;\; H_1$
$\vdots$
$R_N \;\; H_N$
$A_1 \;\; B_1$
$\vdots$
$A_Q \;\; B_Q$

- $1 \le N \le 200\,000$;
- $1 \le Q \le 200\,000$;
- $1 \le R_i \le 1\,000\,000\,000$ $(1 \le i \le N)$;
- $1 \le H_i \le 1\,000\,000\,000$ $(1 \le i \le N)$;
- $1 \le A_j \le 1\,000\,000\,000$ $(1 \le j \le Q)$;
- $1 \le B_j \le 1\,000\,000\,000$ $(1 \le j \le Q)$;

Output

Output Q lines. In the i-th line output the answer to the i-th query.

Examples

standard input	standard output

7 3
9 5
3 7
10 6
5 10
2 6
10 10
4 1
10 5
3 5
3 9
	0
	1
	2

10 8
14 19
9 16
11 2
7 18
20 16
9 5
10 9
20 6
4 17
13 8
7 14
9 3
9 13
4 19
12 4
19 16
18 10
7 14
	3
	1
	3
	5
	0
	2
	1
	3

Problem B. Solitaire

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

There is a $3 \times N$ grid, and some cells contain tokens. You can put a token on a cell X if it doesn't contain a token and it satisfies at least one of the following conditions:

- Two cells with tokens are vertically adjacent to X.
- Two cells with tokens are horizontally adjacent to X.

Compute the number of ways to put tokens in all empty cells and completely fill the grid with tokens, modulo $10^9 + 7$. Two ways are considered distinct if the order of putting tokens are distinct.

Input

- The first line contains the integer N.
- The following three lines contain N characters each, and each character is either an 'o' or an 'x'. These characters represent a cell with a token and a cell without a token, respectively.

- $1 \leq N \leq 2\,000$.
- Initially, at least one cell contains a token.
- Initially, at least one cell doesn't contain a token.

Output

Output the number of ways to put tokens in all empty cells, modulo 1 000 000 007.

Examples

standard input	standard output
3 oxo xxo oxo	14
10 ooxooxoxoo xooxxxoxxx oxoxoooooo	149022720
10 ooxoxxoxoo oxxxxxoxxx oxooxoxoxo	0
20 oxooxoxooxoxooxoxoxo oxxxoxoxxxooxxxxxoox oxooxoxooxooxooxoxoo	228518545

Problem C. Employment

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

A company wants to hire some employees.

There are N candidates for employees numbered 1 through N, and each candidate is assigned a score. Initially the score of the candidate i is A_i.

The company wants to hire all candidates whose scores are at least B and divides them into groups. If both candidates a and b ($a < b$) are hired, they will be assigned to the same group if and only if for all c that satisfies $a \leq c \leq b$, the candidate c is hired.

You are asked to handle the following two types of queries:

- Compute the number of groups for the given value of B.
- Update the score of the candidate C to D.

Input

- The first line contains the integers N and M.
- The i-th ($1 \leq i \leq N$) of the following N lines contains the integer A_i.
- The following M lines contain the information about queries. The first integer T in each line represents the type of the query.
  1. When $T = 1$, this query is of the first type, and this line contains two integers T and B.
  2. When $T = 2$, this query is of the second type, and this line contains three integers T, C, and D.

- $1 \leq N \leq 200\,000$;
- $1 \leq M \leq 200\,000$;
- $1 \leq A_i \leq 1\,000\,000\,000$ ($1 \leq i \leq N$);
- $1 \leq T_j \leq 2$ ($1 \leq j \leq M$);
- $1 \leq B_j \leq 1\,000\,000\,000$ ($1 \leq j \leq M$);
- $1 \leq C_j \leq N$ ($1 \leq j \leq M$);
- $1 \leq D_j \leq 1\,000\,000\,000$ ($1 \leq j \leq M$;
- There will be at least one query of the first type.

Output

For each query of the first type, print the answer in a single line.

Example

standard input	standard output

5 4
8
6
3
5
4
1 5
2 4 1
1 5
1 3
	2
	1
	2

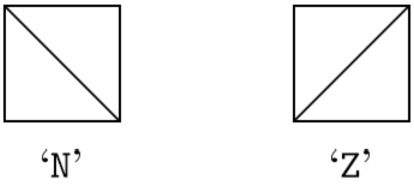
7 5
13
19
1
15
13
1
19
1 20
1 1
1 6
1 11
1 17
	0
	1
	3
	3
	2

10 5
8
10
15
2
2
8
5
12
11
4
1 5
2 8 4
1 12
2 5 11
1 16
	2
	1
	0

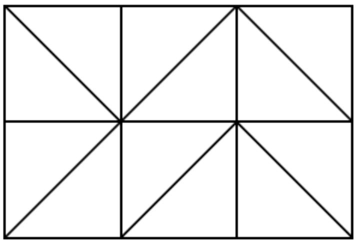
Problem D. Sandwich

Input file:	*standard input*
Output file:	*standard output*
Time limit:	40 seconds
Memory limit:	512 mebibytes

There is an $R \times C$ grid, and each cell contains two triangular sandwiches in one of the following two orientations:



The following picture shows an example of a grid:



Initially, no sandwiches are taken.

If a sandwich satisfies both of the following conditions, you can't take the sandwich:

- The other sandwich in the same cell is not taken.
- It touches with at least one untaken sandwich in a different cell by a side.

Otherwise you can take the sandwich.

Initially, no sandwiches are taken. For each cell in the grid, compute the minimum number of sandwiches you must take in order to take both sandwiches in the cell, including the two sandwiches in the cell.

Input

- The first line contains two integers R and C.
- The following R lines contain C characters each. Each character is either an 'N' or a 'Z', and it represents the orientation of the sandwiches (as shown in the picture above).
- $1 \leq R \leq 400$;
- $1 \leq C \leq 400$;

Output

Output R lines. In each line, output a space-separated list of C integers. Each of these integers is the minimum number of sandwiches you must take in order to take both sandwiches in the corresponding cell. If it's impossible to take both sandwiches in the cell, print -1 instead.

Example

standard input	standard output
2 3 NZN ZZN	10 8 2 8 6 4
2 2 NZ ZN	-1 -1 -1 -1

5 5
NZZZN
NNNZN
NNZNN
NZNNN
NZZZN
	10 12 14 16 2
	8 -1 -1 -1 4
	6 -1 -1 -1 6
	4 -1 -1 -1 8
	2 16 14 12 10

Problem E. Toilets

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

There are two toilets. One of them is for women only, and the other one is for both men and women.

$2N$ people are queued in a line to use toilets. They enter toilets according to the following rule:

- If the person at the front is a woman, she enters a vacant toilet. If both toilets are vacant, she enters the toilet for women only.
- If the person at the front is a man,
  - If the toilet for both men and women is vacant, he enters it.
  - Otherwise, if the toilet for women only is empty, (and if at least one woman is waiting in the queue), the woman who is closest to the front of the queue leaves the queue and enters the toilet.

Each person spends a minute in toilets. Before they start using the toilets, you want to change the order of the queue such that they can finish using the toilets in N minutes.

The *unfairness* of a reordering of the queue is defined as the minimum A such that for each person p in the queue, the number of people who overtake p during the reordering process is at most A. Note that we only count those who overtake p **during the reordering process**. Some people may overtake others when they enter toilets, but it doesn't affect the definition of unfairness.

You want to find the reordering that minimizes the unfairness under the constraint that they must be able to finish using the toilets in N minutes.

In order to compress the input, the input will be given in the following format. You are given M strings $S_1, \ldots, S_M$ and M integers $K_1, \ldots, K_M$. Let X be the string obtained by the concatenation of K_1 repetitions of S_1, followed by K_2 repetitions of S_2, …, followed by K_M repetitions of S_M. The i-th character of X represents the i-th person from the front in the initial queue, and it is either an 'M' (male) or an 'F' (female).

Input

N
M
$S_1 \;\; K_1$
$\vdots$
$S_M \;\; K_M$

- $1 \leq N \leq 1\,000\,000\,000\,000\,000\,000 \; (= 10^{18})$;
- $1 \leq M \leq 100\,000$;
- $1 \leq K_i \leq 2N \; (1 \leq i \leq M)$;
- $1 \leq |S_i| \leq 2N \; (1 \leq i \leq M)$;
- Each charater in S_i $(1 \leq i \leq M)$ is either an '`M`' or an '`F`';
- $|S_1| + |S_2| + \cdots + |S_M| \leq 200\,000$;
- The input defines a string of length $2N$;

Output

Print the minimum possible unfairness of the reordering under the constraint that they must be able to finish using the toilets in N minutes. If such reordering doesn't exist, print -1 instead.

Examples

standard input	standard output
6 1 FFFMMMMMMFFF 1	2
6 1 MMFFMMMMFFMF 1	-1
6 1 MFFFMFMMFFFM 1	0
6 4 M 1 F 2 FM 2 MFFFM 1	0

Problem F. Sushi

Input file:	*standard input*
Output file:	*standard output*
Time limit:	20 seconds
Memory limit:	512 mebibytes

N people numbered 1 through N are arranged in a circle. Initially the person i has a dish with x_i sushis.

You are asked to handle Q queries. The i-th query has three parameters s_i, t_i, and p_i. In this query, you need to do the following:

- You have a dish with p_i sushis.
- For each $k = s_i, s_i + 1, \ldots, t_i$, (when $s_i > t_i$, for each $k = s_i, \ldots, N, 1, \ldots, t_i$), in this order, if your dish has less sushis than the k-th person's dish, swap the dishes.
- You eat all sushis on the dish.

Compute the number of sushis you eat in each query.

Note that these queries will be performed in order. After each query, the number of sushis on dishes will be changed.

Input

N Q
x_1
$\vdots$
x_N
s_1 t_1 p_1
$\vdots$
s_Q t_Q p_Q

- $1 \leq N \leq 400\,000$.
- $1 \leq Q \leq 25\,000$.
- $1 \leq x_i \leq 1\,000\,000\,000$ $(1 \leq i \leq N)$.
- $1 \leq s_i \leq N$ $(1 \leq i \leq Q)$.
- $1 \leq t_i \leq N$ $(1 \leq i \leq Q)$.
- $1 \leq p_i \leq 1\,000\,000\,000$ $(1 \leq i \leq Q)$.

Output

Output Q lines. In the i-th line output the answer of the i-th query.

Examples

standard input	standard output

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

4 2
5
2
4
7
1 4 3
1 4 1
	7
	5

10 10
19
5
8
17
14
3
9
10
7
6
1 8 4
7 3 2
5 9 10
4 8 3
10 3 6
8 7 4
6 6 3
2 9 12
6 3 7
9 6 3
	19
	10
	14
	17
	8
	10
	3
	12
	7
	9

Problem G. Telegraph

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

There are N islands numbered 1 through N.

Each island has a receiver, and it can receive messages from another island. When it directs to the island i, it can receive messages from the island i. However, it is impossible to receive messages from multiple islands.

Initially, the receiver at the island i directs to the island A_i. By paying C_i dollars, you can change the direction of this receiver (the cost doesn't depend on the new direction).

You want to change the directions of some receivers such that for any (i, j), it is possible to send messages from the island i to the island j. Compute the minimum cost required to satisfy this condition.

Input

N
$A_1 \;\; C_1$
$\vdots$
$A_N \;\; C_N$

- $2 \le N \le 100\,000$;
- $1 \le A_i \le N$ $(1 \le i \le N)$;
- $A_i \ne i$ $(1 \le i \le N)$;
- $1 \le C_i \le 1\,000\,000\,000$ $(1 \le i \le N)$;

Output

Print the minimum cost required to satisfy the condition.

Examples

standard input	standard output

4
2 2
1 4
1 3
3 1
	4

4
2 2
1 6
1 3
3 1
	5

4
2 2
1 3
4 2
3 3
	4

3
2 1
3 1
1 1
	0

Problem H. Dangerous Skating

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

There is an $H \times W$ grid. Each cell of the grid is either an ice block or a field. It is guaranteed that all cells on the edges are ice blocks.

In this task we label the cell that is the r-th from the top and the c-th from the left as (r, c). Initially, the cell (r, c) is $s_{r,c}$, where a '`.`' represents a field and a '`#`' represents an ice block.

Initially, Mr. Joi is standing on the cell (r_1, c_1). In each step, he chooses one of the four cardinal directions, and keeps moving in this direction until he hits an ice block (and he stops one cell behind the ice block he hits). Surprisingly, after each step, the cell he starts moving will be changed to an ice block.

For example, consider the following grid. '`J`' represents the position of Mr. Joi, and '`I`' represents an ice block.

I	I	I	I	I
I	J			I
I				I
I				I
I	I	I	I	I

If he moves to the right, the grid will be as follows after a step:

I	I	I	I	I
I	I		J	I
I				I
I				I
I	I	I	I	I

Compute the minimum number of steps required to stop at the cell (r_2, c_2). Note that it is not enough to pass through this cell; he must stop at this cell.

Input

$H \ \ W$
$s_{1,1} \cdots s_{1,W}$
$\vdots$
$s_{H,1} \cdots s_{H,W}$
$r_1 \ \ c_1$
$r_2 \ \ c_2$

- $3 \leq H \leq 1\,000$;
- $3 \leq W \leq 1\,000$;
- $1 \leq r_1 \leq H$;
- $1 \leq c_1 \leq W$;
- $1 \leq r_2 \leq H$;
- $1 \leq c_2 \leq W$;
- $s_{i,j}$ is either a '`.`' or a '`#`'.

- Each cell on the edge is an ice field.
- The cells (r_1, c_1) and (r_2, c_2) are fields.

Output

Print the minimum number of steps required to stop at the cell (r_2, c_2). If this is impossible, print -1 instead.

Examples

standard input	standard output
5 5 ##### #...# #...# #...# ##### 2 2 3 3	4
8 6 ###### #..#.# ##...# #....# #.#..# #....# ##...# ###### 4 3 6 4	5
5 5 ##### #.#.# #.#.# #.#.# ##### 2 2 4 4	-1
3 3 ### #.# ### 2 2 2 2	0

Problem I. Worst Reporter 2

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

N contestants have participated in a programming competition. A reporter reported the following information:

- Two hours after the beginning of the contest, no two contestants had the same score. The i-th ($1 \le i \le N$) ranked contestant comes from the country A_i, and his score was B_i.
- Five hours after the beginning of the contest, no two contestants had the same score. The i-th ($1 \le i \le N$) ranked contestant comes from the country C_i, and his score was D_i.
- The score of each contestant never decreased.

However the reporter may have made mistakes in the information about countries, and the information may be inconsistent. You want to make the information consistent by changing the minimum number of values from $A_1, \ldots, A_N, C_1, \ldots, C_N$. Note that you are not allowed to change the values of $B_1, \ldots, B_N, D_1, \ldots, D_N$.

Compute the minimum number of changes required to make the information consistent.

Formally, the information is consistent if there exists a permutation $(x_1, x_2, \ldots, x_N)$ of $(1, 2, \ldots, N)$ such that for each $i = 1, 2, \ldots, N$, both $A_i = C_{x_i}$ and $B_i \le D_{x_i}$ are satistied.

Input

N
$A_1 \;\; B_1$
$\vdots$
$A_N \;\; B_N$
$C_1 \;\; D_1$
$\vdots$
$C_N \;\; D_N$

- $2 \le N \le 200\,000$;
- $1 \le A_i \le N$ ($1 \le i \le N$);
- $0 \le B_i \le 1\,000\,000\,000$ ($1 \le i \le N$);
- $B_i > B_{i+1}$ ($1 \le i \le N - 1$);
- $1 \le C_i \le N$ ($1 \le i \le N$);
- $0 \le D_i \le 1\,000\,000\,000$ ($1 \le i \le N$);
- $D_i > D_{i+1}$ ($1 \le i \le N - 1$);
- By changing some of the values $A_1, \ldots, A_N, C_1, \ldots, C_N$, you can make the standings consistent.

Output

Output the minimum number of changes required to make the information consistent.

Examples

standard input	standard output

3
3 500
2 200
1 100
1 1000
3 700
3 400
	1

3
3 3
3 2
1 1
3 4
3 2
1 1
	0

6
1 70
4 50
1 30
2 20
1 10
3 0
6 100
2 90
1 80
2 60
4 40
1 10
	3