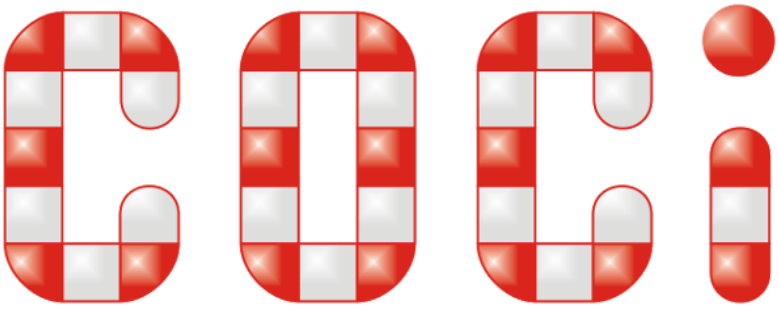

COCI 2017/2018

Round #2, November 4th, 2017

Tasks

Task	Time limit	Memory limit	Score
Košnja	1 s	64 MB	50
ZigZag	2 s	64 MB	80
Doktor	1 s	128 MB	100
San	1 s	64 MB	120
Usmjeri	2 s	256 MB	140
Garaža	4 s	256 MB	160
Total			650

Mirko wants to buy land on which he will build a house for his family. So far, he's seen K pieces of land. Each of them is in the shape of a rectangle and we can think of it as a matrix with N rows and M columns, $N \times M$ fields in total.

Mirko is aware that, before construction begins, the property needs to be regularly maintained and the lawn needs to be mowed. Because of this, Mirko bought a lawn mower. In order to mow the entire lawn of N rows and M columns, he needs to go over each field at least once. He can start from any field facing one of the four main directions (up, down, left, and right). His lawn mower can only go forwards (to the adjacent field facing the current direction) or make a 90 degree turn. Additionally, because of his own safety, Mirko can only use the lawn mower on his land, so he cannot leave the matrix.

Since making the lawn mower turn isn't simple, Mirko wants to mow the lawn with the minimal amount of turns. For each piece of land he saw so far, Mirko wants to know the minimal number of turns he can make so that the entire lawn is mowed. Help Mirko solve this problem.

INPUT

The first line of input contains the positive integer K ($1 \leq K \leq 50\ 000$), the number from the task.
Each of the following K lines contains two positive integers N and M ($1 \leq N, M \leq 1\ 000\ 000$), the numbers from the task.

OUTPUT

For each piece of land Mirko saw so far, output in a separate line the minimal amount of turns he can take so that the entire lawn is mowed.

SCORING

In test cases worth 50% of total points, Mirko will see only one piece of land. The dimensions of this piece of land will be smaller than 500.

SAMPLE TESTS

input	input	input
2 1 10 10 1	3 1 1 3 3 3 4	2 5 8 6 4
output	**output**	**output**
0 0	0 4 4	8 6

Clarification of the first test case:
The first piece of land can be mowed without making any turns if he starts from the field in the first column of the table, faced to the right and only going forwards. A similar idea applies for the second piece of land.

Zig and Zag are playing a word game. Zig says one letter, and Zag says a word that starts with that letter. However, the word needs to be from the allowed word list and such that Zag already said it the least amount of times. If the word choice is ambiguous, then Zag will choose the one that is lexicographically smaller (sooner in the alphabet). For each Zig's letter, it will be possible to choose a word.

Let there be a list consisting of exactly K distinct words and an array of N letters that Zig has given. Write a program that will, based on the input, output an array of N words that Zag said during the game.

INPUT

The first line of input contains positive integers K ($1 \leq K \leq 100\,000$) and N ($1 \leq N \leq 100\,000$) from the task.

Each of the following K lines contains a single word consisting of lowercase letters of the English alphabet not longer than 21 characters.

Each of the following N lines contains a single lowercase letter of the English alphabet.

OUTPUT

You must output N lines, each containing a single word from the task.

SCORING

In test cases worth 60% of total points, it will hold that N and K are smaller than 500.

SAMPLE TESTS

input

```
4 5
zagreb
split
zadar
sisak
z
s
s
z
z
```

output

```
zadar
sisak
split
zagreb
zadar
```

input

```
5 3
london
rim
pariz
moskva
sarajevo
p
r
p
```

output

```
pariz
rim
pariz
```

input

```
1 3
zagreb
z
z
z
```

output

```
zagreb
zagreb
zagreb
```

And the Mrs, Ms, says:
"I've been riding horses for fifteen years, and it is impossible to shoe a horse upside down!"
(...)

"Yes, that's upside down" - whispers Domagoj, looking at Mate's hand while playing, for the purposes of this task, a heavily modified version of the card game Hanabi. For the sake of simplicity, Mate is holding N cards in his hands, numbered from 1, 2, …, N in a certain order. (Each number from 1 to N appears exactly once.) As when playing the real game, he cannot voluntarily change the card order.

Just so the task is at least somewhat related to the story, Domagoj will point for Mate to a contiguous subarray of cards. (He can point to a single card too, but he will point to at least one card.) Mate will then "rotate" that contiguous subarray and put it back.

(The rotating can be thought of as taking all the cards in the given subarray and rotating all of them for 180 degrees. This means that the first and last card swap places, as well as the second and the second to last card, and so on.)

Like all of us, Domagoj is very fond of fixed points. In other words, cards whose numbers match their position in hand, counting from Domagoj's left side. Therefore, he'd like the number of fixed points to be as great as possible after rotating the given subarray.

Help Domagoj find out which contiguous subarray he needs to point out so that the number of fixed points in Mate's hand after rotating that subarray is maximal.

INPUT

The first line of input contains the positive integer N ($1 \leq N \leq 500\,000$), the number of cards in Mate's hand.
The following line contains the numbers on the cards in Mate's hand in the order that Domagoj sees them.

OUTPUT

You must output a single line containing A and B, the numbers on the cards that are the beginning and the end of the required contiguous subarray, in that order. If there are multiple options, output any of them.

SCORING

In test cases worth 30% of total points, it will hold $N \leq 500$.
In test cases worth an additional 30% of total points, it will hold $N \leq 5000$.

SAMPLE TESTS

input	input	input
4 3 2 1 4	2 1 2	7 3 6 5 7 4 1 2
output	**output**	**output**
3 1	1 1	3 2

Clarification of the test cases:

In the first test case, after rotating the contiguous subarray that starts with 3 and ends with 1, the cards will be ordered 1 2 3 4, and now all the cards are fixed points. In this example, the given output is the only correct output.

In the second test case, rotating any subarray consisting of only one card results with the same card order, the one that produces the maximal number of fixed points.

Have you ever dreamt that you were the main character in a computer game? The protagonist of this story, Branimir, is having that dream right now.

The world in Branimir's dream consists of N skyscrapers ordered from left to right. For the i^{th} skyscraper, we know the height H_i and the number of gold coins G_i on the roof of the skyscraper. The game begins with a jump on any of the skyscrapers and consists of several steps. In each step, Branimir can jump on a skyscraper to the **right** from the one he's currently on (it is possible for him to jump over a couple of them too) and if it **is not lower** than the current one. On each skyscraper roof he's on, Branimir will collect all the gold coins. Branimir can end the game after any number of steps (zero as well), but he must collect at least K gold coins in order to advance to the next level.

Branimir wants to know the number of different ways for him to play the game in order to advance to the next level. Two games are played in different ways if there is a skyscraper that was visited in one game, and not in the other.

INPUT

The first line of input contains positive integers N ($1 \leq N \leq 40$) and K ($1 \leq K \leq 4 \cdot 10^{10}$), the numbers from the task.
The i^{th} of the following N lines contains two positive integers, H_i and G_i ($1 \leq H_i, G_i \leq 10^9$), the numbers from the task.

OUTPUT

You must output the number of different ways to play the game from the task.

SCORING

In test cases worth 40% of total points, it will hold $N \leq 20$.

SAMPLE TESTS

input	input	input
4 6 2 1 6 3 7 2 5 6	2 7 4 6 3 5	4 15 5 5 5 12 6 10 2 1
output	**output**	**output**
3	0	4

Clarification of the first test case:
The following three games will take Branimir to the next level (the numbers represent the labels of the skyscrapers he visited): {1, 2, 3}, {1, 4} and {4}.

We are given a tree[1] with N nodes denoted with different positive integers from 1 to N.
Additionally, you are given M node pairs from the tree in the form of $(a_1, b_1), (a_2, b_2), \ldots, (a_M, b_M)$.
We need to direct each edge of the tree so that for each given node pair (a_i, b_i) there is a path from a_i to b_i or from b_i to a_i. How many different ways are there to achieve this?
Since the solution can be quite large, determine it modulo $10^9 + 7$.

INPUT

The first line of input contains the positive integers N and M ($1 \leq N, M \leq 3 \cdot 10^5$), the number of nodes in the tree and the number of given node pairs, respectively.
Each of the following N - 1 lines contains two positive integers, the labels of the nodes connected with an edge.
The i^{th} of the following M lines contains two different positive integers a_i and b_i, the labels of the nodes from the i^{th} node pair. All node pairs will be mutually different.

OUTPUT

You must output a single line containing the total number of different ways to direct the edges of the tree that meet the requirement from the task, modulo $10^9 + 7$.

SCORING

In test cases worth 20% of total points, the given tree will be a chain. In other words, node i will be connected with an edge to node $i + 1$ for all $i < N$.
In additional test cases worth 40% of total points, it will hold $N, M \leq 5 \cdot 10^3$.

SAMPLE TESTS

input

```
4 1
1 2
2 3
3 4
2 4
```

output

```
4
```

input

```
7 2
1 2
1 3
4 2
2 5
6 5
5 7
1 7
2 6
```

output

```
8
```

input

```
4 3
1 2
1 3
1 4
2 3
2 4
3 4
```

output

```
0
```

[1] *A tree* is a graph that consists of N nodes and N - 1 edges such that there exists a path from each node to each other node.

Lately, Slavko's been studying sequences of natural numbers. He finds a sequence interesting if the greatest common divisor of all the elements from the sequence is greater than 1.

Yesterday, he found a sequence consisting of N natural numbers in his garage. Since he was really bored, he decided to keep himself occupied by asking simple queries. Each query can be one of the two types:

1. Change the value at position X in the sequence to V.
2. Determine the number of interesting contiguous subarrays contained in the interval [L, R] of the sequence.

INPUT

The first line of input contains the numbers N and Q ($1 \leq N, Q \leq 10^5$), representing the number of elements in the sequence and the number of queries, respectively.
The following line contains N natural numbers A_i ($1 \leq A_i \leq 10^9$) that represent the numbers in the initial sequence.
Each of the following Q lines contains a query of the following form:

- The first number in the line can be 1 or 2 and represents the type of the query.
- If the query is of type 1, two numbers follow, X ($1 \leq X \leq N$) and V ($1 \leq V \leq 10^9$) from the task.
- If the query is of type 2, two numbers follow, L and R ($1 \leq L \leq R \leq N$) that represent the left and right interval boundary.

OUTPUT

For each query of type 2, output the number of interesting contiguous subarrays from the task.

SAMPLE TESTS

input

```
5 1
8 4 3 9 1
2 2 5
```

output

```
4
```

input

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

output

```
6
1
```

input

```
4 3
2 2 2 2
2 1 4
1 2 3
2 1 4
```

output

```
10
5
```

Clarification of the first test case:

The interval from the 2nd to the 5th position consists of numbers (4, 3, 9, 1). In it, the following are interesting contiguous subarrays (denoted with square brackets):

[4] 3 9 1, 4 ***[3]*** 9 1, 4 3 ***[9]*** 1, 4 ***[3 9]*** 1