A. Nastya Is Reading a Book

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

After lessons Nastya decided to read a book. The book contains n chapters, going one after another, so that one page of the book belongs to exactly one chapter and each chapter contains at least one page.

Yesterday evening Nastya did not manage to finish reading the book, so she marked the page with number k as the first page which was not read (i.e. she read all pages from the 1-st to the $(k-1)$-th).

The next day Nastya's friend Igor came and asked her, how many chapters remain to be read by Nastya? Nastya is too busy now, so she asks you to compute the number of chapters she has not completely read yet (i.e. the number of chapters she has not started to read or has finished reading somewhere in the middle).

Input

The first line contains a single integer n $(1 \leq n \leq 100)$ — the number of chapters in the book.

There are n lines then. The i-th of these lines contains two integers l_i, r_i separated by space ($l_1 = 1$, $l_i \leq r_i$) — numbers of the first and the last pages of the i-th chapter. It's guaranteed that $l_{i+1} = r_i + 1$ for all $1 \leq i \leq n-1$, and also that every chapter contains at most 100 pages.

The $(n+2)$-th line contains a single integer k ($1 \leq k \leq r_n$) — the index of the marked page.

Output

Print a single integer — the number of chapters which has not been completely read so far.

Examples

input
```
3
1 3
4 7
8 11
2
```
output
```
3
```

input
```
3
1 4
5 9
10 12
9
```
output
```
2
```

input
```
1
1 7
4
```
output
```
1
```

Note

In the first example the book contains 11 pages and 3 chapters — $[1;3]$, $[4;7]$ and $[8;11]$. Nastya marked the 2-nd page, so she finished in the middle of the 1-st chapter. So, all chapters has not been read so far, so the answer is 3.

The book in the second example contains 12 pages and 3 chapters too, but Nastya finished reading in the middle of the 2-nd chapter, so that the answer is 2.

B. Nastya Is Playing Computer Games

time limit per test: 1 second
memory limit per test: 256 megabytes

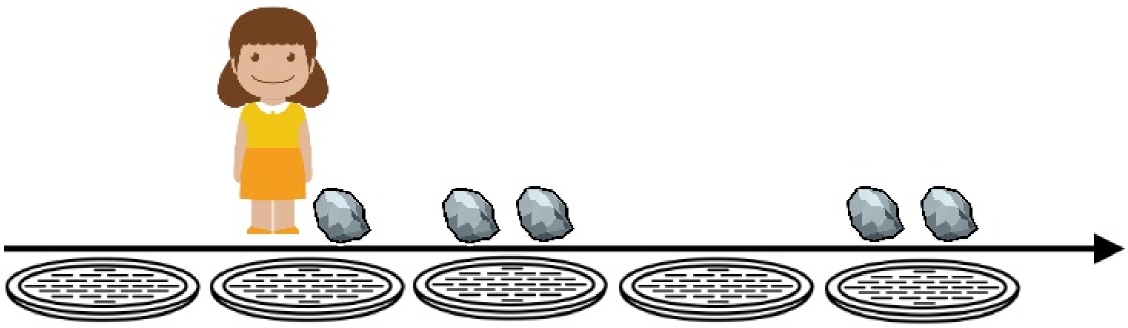
input: standard input
output: standard output

Finished her homework, Nastya decided to play computer games. Passing levels one by one, Nastya eventually faced a problem. Her mission is to leave a room, where a lot of monsters live, as quickly as possible.

There are n manholes in the room which are situated on one line, but, unfortunately, all the manholes are closed, and there is one stone on every manhole. There is exactly one coin under every manhole, and to win the game Nastya should pick all the coins. Initially Nastya stands near the k-th manhole from the left. She is thinking what to do.

In one turn, Nastya can do one of the following:

- if there is at least one stone on the manhole Nastya stands near, throw exactly one stone from it onto any other manhole (yes, Nastya is strong).
- go to a neighboring manhole;
- if there are no stones on the manhole Nastya stays near, she can open it and pick the coin from it. After it she must close the manhole immediately (it doesn't require additional moves).



The figure shows the intermediate state of the game. At the current position Nastya can throw the stone to any other manhole or move left or right to the neighboring manholes. If she were near the leftmost manhole, she could open it (since there are no stones on it).

Nastya can leave the room when she picks all the coins. Monsters are everywhere, so you need to compute the minimum number of moves Nastya has to make to pick all the coins.

Note one time more that Nastya can open a manhole only when there are no stones onto it.

Input
The first and only line contains two integers n and k, separated by space ($2 \leq n \leq 5000$, $1 \leq k \leq n$) — the number of manholes and the index of manhole from the left, near which Nastya stays initially. Initially there is exactly one stone near each of the n manholes.

Output
Print a single integer — minimum number of moves which lead Nastya to pick all the coins.

Examples

input
```
2 2
```
output
```
6
```

input
```
4 2
```
output
```
13
```

input
```
5 1
```
output
```
15
```

Note
Let's consider the example where $n = 2$, $k = 2$. Nastya should play as follows:

- At first she throws the stone from the second manhole to the first. Now there are two stones on the first manhole.
- Then she opens the second manhole and pick the coin from it.
- Then she goes to the first manhole, throws two stones by two moves to the second manhole and then opens the manhole and picks the coin from it.

So, 6 moves are required to win.

C. Nastya Is Transposing Matrices

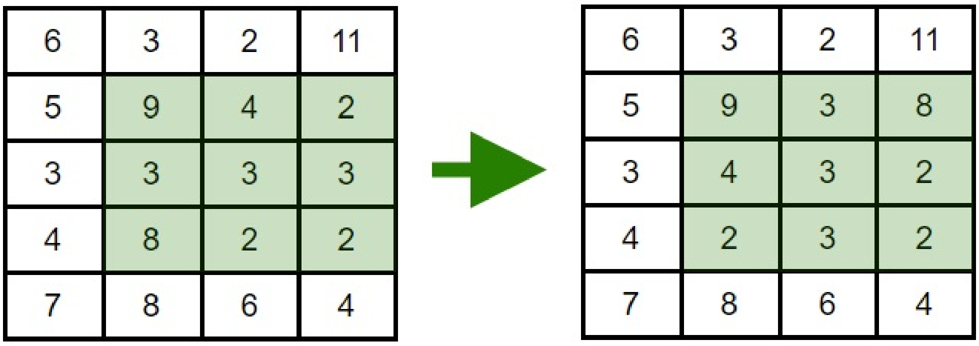
time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Nastya came to her informatics lesson, and her teacher who is, by the way, a little bit famous here gave her the following task.

Two matrices A and B are given, each of them has size $n \times m$. Nastya can perform the following operation to matrix A unlimited number of times:

- take any square *square submatrix* of A and transpose it (i.e. the element of the submatrix which was in the i-th row and j-th column of the submatrix will be in the j-th row and i-th column after transposing, and the transposed submatrix itself will keep its place in the matrix A).

Nastya's task is to check whether it is possible to transform the matrix A to the matrix B.



Example of the operation

As it may require a lot of operations, you are asked to answer this question for Nastya.

A *square submatrix* of matrix M is a matrix which consist of all elements which comes from one of the rows with indeces $x, x+1, \dots, x+k-1$ of matrix M and comes from one of the columns with indeces $y, y+1, \dots, y+k-1$ of matrix M. k is the size of *square submatrix*. In other words, *square submatrix* is the set of elements of source matrix which form a solid square (i.e. without holes).

Input

The first line contains two integers n and m separated by space ($1 \leq n, m \leq 500$) — the numbers of rows and columns in A and B respectively.

Each of the next n lines contains m integers, the j-th number in the i-th of these lines denotes the j-th element of the i-th row of the matrix A ($1 \leq A_{ij} \leq 10^9$).

Each of the next n lines contains m integers, the j-th number in the i-th of these lines denotes the j-th element of the i-th row of the matrix B ($1 \leq B_{ij} \leq 10^9$).

Output

Print "YES" (without quotes) if it is possible to transform A to B and "NO" (without quotes) otherwise.

You can print each letter in any case (upper or lower).

Examples

input

```
2 2
1 1
6 1
1 6
1 1
```

output

```
YES
```

input

```
2 2
4 4
4 5
5 4
4 4
```

output

```
NO
```

input

```
3 3
1 2 3
```

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

output

```
YES
```

Note

Consider the third example. The matrix A initially looks as follows.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Then we choose the whole matrix as transposed submatrix and it becomes

$$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

Then we transpose the submatrix with corners in cells $(2, 2)$ and $(3, 3)$.

$$\begin{bmatrix} 1 & 4 & 7 \\ 2 & \mathbf{5} & \mathbf{8} \\ 3 & \mathbf{6} & \mathbf{9} \end{bmatrix}$$

So matrix becomes

$$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 6 \\ 3 & 8 & 9 \end{bmatrix}$$

and it is B.

D. Nastya Is Buying Lunch

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

At the big break Nastya came to the school dining room. There are n pupils in the school, numbered from 1 to n. Unfortunately, Nastya came pretty late, so that all pupils had already stood in the queue, i.e. Nastya took the last place in the queue. Of course, it's a little bit sad for Nastya, but she is not going to despond because some pupils in the queue can agree to change places with some other pupils.

Formally, there are some pairs u, v such that if the pupil with number u stands directly in front of the pupil with number v, Nastya can ask them and they will change places.

Nastya asks you to find the maximal number of places in queue she can move forward.

Input

The first line contains two integers n and m ($1 \leq n \leq 3 \cdot 10^5$, $0 \leq m \leq 5 \cdot 10^5$) — the number of pupils in the queue and number of pairs of pupils such that the first one agrees to change places with the second one if the first is directly in front of the second.

The second line contains n integers p_1, p_2, ..., p_n — the initial arrangement of pupils in the queue, from the queue start to its end ($1 \leq p_i \leq n$, p is a permutation of integers from 1 to n). In other words, p_i is the number of the pupil who stands on the i-th position in the queue.

The i-th of the following m lines contains two integers u_i, v_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i$), denoting that the pupil with number u_i agrees to change places with the pupil with number v_i if u_i is directly in front of v_i. It is guaranteed that if $i \neq j$, than $v_i \neq v_j$ or $u_i \neq u_j$. Note that it is possible that in some pairs both pupils agree to change places with each other.

Nastya is the last person in the queue, i.e. the pupil with number p_n.

Output

Print a single integer — the number of places in queue she can move forward.

Examples

input

```
2 1
1 2
1 2
```

output

```
1
```

input

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

output

```
2
```

input

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

output

```
1
```

Note

In the first example Nastya can just change places with the first pupil in the queue.

Optimal sequence of changes in the second example is

- change places for pupils with numbers 1 and 3.
- change places for pupils with numbers 3 and 2.
- change places for pupils with numbers 1 and 2.

The queue looks like $[3, 1, 2]$, then $[1, 3, 2]$, then $[1, 2, 3]$, and finally $[2, 1, 3]$ after these operations.

E. Nastya Hasn't Written a Legend

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In this task, Nastya asked us to write a formal statement.

An array a of length n and an array k of length $n-1$ are given. Two types of queries should be processed:

- increase a_i by x. Then if $a_{i+1} < a_i + k_i$, a_{i+1} becomes exactly $a_i + k_i$; again, if $a_{i+2} < a_{i+1} + k_{i+1}$, a_{i+2} becomes exactly $a_{i+1} + k_{i+1}$, and so far for a_{i+3}, ..., a_n;
- print the sum of the contiguous subarray from the l-th element to the r-th element of the array a.

It's guaranteed that initially $a_i + k_i \le a_{i+1}$ for all $1 \le i \le n-1$.

Input

The first line contains a single integer n ($2 \le n \le 10^5$) — the number of elements in the array a.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($-10^9 \le a_i \le 10^9$) — the elements of the array a.

The third line contains $n-1$ integers $k_1, k_2, \ldots, k_{n-1}$ ($-10^6 \le k_i \le 10^6$) — the elements of the array k.

The fourth line contains a single integer q ($1 \le q \le 10^5$) — the number of queries.

Each of the following q lines contains a query of one of two types:

- if the query has the first type, the corresponding line contains the character '+' (without quotes), and then there are two integers i and x ($1 \le i \le n$, $0 \le x \le 10^6$), it means that integer x is added to the i-th element of the array a as described in the statement.
- if the query has the second type, the corresponding line contains the character 's' (without quotes) and then there are two integers l and r ($1 \le l \le r \le n$).

Output

For each query of the second type print a single integer in a new line — the sum of the corresponding subarray.

Examples

input

```
3
1 2 3
1 -1
5
```

```
s 2 3
+ 1 2
s 1 2
+ 3 1
s 2 3
```

output

```
5
7
8
```

input

```
3
3 6 7
3 1
3
+ 1 3
+ 2 4
s 1 3
```

output

```
33
```

Note

In the first example:

- after the first change $a = [3, 4, 3]$;
- after the second change $a = [3, 4, 4]$.

In the second example:

- after the first change $a = [6, 9, 10]$;
- after the second change $a = [6, 13, 14]$.