CEOI 2022

CENTRAL EUROPEAN OLYMPIAD IN INFORMATICS

VARAZDIN, CROATIA, JULY 24 - 30

Day 1

July 26th 2022

Tasks

Task	Time Limit	Memory Limit	Score
Abracadabra	3 seconds	512 MiB	100
Homework	1 second	512 MiB	100
Prize	3.5 seconds	1024 MiB	100
Total			300

REPUBLIC OF CROATIA
Ministry of Science and Education

CROATIAN ASSOCIATION OF TECHNICAL CULTURE

CROATIAN COMPUTER SCIENCE ASSOCIATION

Task: Abracadabra

Tin Golubić, also known as *Mr. Magic Man*, is one of Varaždin's most talented young magicians. His specialty is card magic, and this task is an homage to some of the truly impressive feats of magic we have witnessed over the years.

Tin's trick featured in this task involves a deck of N cards, where each card has a unique integer from 1 to N written on its face, and the total number of cards is even. Tin is going to perform what seems to be a series of *riffle shuffles*, and at any time an audience member may shout out a question: *"What was the number written on the face of the i-th card from the bottom after you performed t shuffles?"*. Naturally, Tin will immediately respond with the correct answer.

The secret behind the trick involves a mix of Tin's incredible mental abilities and his card-handling dexterity. First, he is going to perfectly remember the starting state of the deck, meaning he knows exactly which card is in which position initially.

Then, he will use a slightly nuanced version of a standard riffle shuffle that goes unnoticed by the audience. Similar to your typical riffle, Tin will take the bottom half of the cards in his left hand, and the top half of the cards in his right hand, having them face down at all times, and proceed to drop them one by one to form the new deck on the table. Instead of arbitrarily dropping a bottom card from one of his hands, he will always drop the bottom card with the smaller number written on its face. Additionally, once he has dropped all cards from one of his hands, he drops all the remaining cards from his other hand as well. The dropped cards are then collected, and the shuffle is complete.

Starting from the initial deck, Tin will repeatedly perform his shuffle on the current deck, obtaining a new ordering of the cards on which the next shuffle will be executed.

Your task is to write a program that simulates Tin's trick, i.e. given an initial state of the deck, you'll need to answer Q queries from the audience.

Input

The first line contains two space-separated integers N and Q from the task description. It is guaranteed that N is even.

The second line contains N space-separated positive integers, a permutation of the set $\{1, 2, \ldots, N\}$ representing the initial state of the deck from the bottom to the top.

The j-th of the next Q lines contains two space-separated integers t and i ($1 \leq i \leq N$), describing the j-th query from the audience. More precisely, the query asks about the number written on the i-th card from the bottom of the deck after completing t shuffles.

Output

Output Q lines, the j-th of which contains a single positive integer between 1 and N, the answer to the j-th query.

Scoring

In all subtasks it holds that $2 \leq N \leq 200\,000$, $1 \leq Q \leq 1\,000\,000$ and $0 \leq t \leq 10^9$.

Subtask	Score	Constraints
1	10	$N \leq 1000$
2	40	All queries have the same t value.
3	25	$N, Q \leq 100\,000$
4	25	No additional constraints.

Examples

input

```
6 3
1 5 6 2 3 4
1 2
0 4
1 5
```

output

```
2
2
5
```

input

```
6 6
2 1 5 4 6 3
0 1
1 1
0 3
1 3
0 6
10 6
```

output

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

input

```
10 10
7 5 2 9 10 8 4 3 6 1
3 1
3 2
3 3
3 4
3 5
3 6
3 7
3 8
3 9
3 10
```

output

```
2
3
6
1
7
5
8
4
9
10
```

Clarification of the third example:
The table below shows the state of the deck after each shuffle. All the queries have $t = 3$, so the output is precisely the state of the deck after 3 shuffles.

Number of shuffles	Deck (bottom to top)
0	7 5 2 9 10 8 4 3 6 1
1	7 5 2 8 4 3 6 1 9 10
2	3 6 1 7 5 2 8 4 9 10
3	2 3 6 1 7 5 8 4 9 10

Task: Homework

Little Helena recently finished her first year of primary school. She is a model student, has straight A's, and has a huge passion for mathematics. She is currently on a well-deserved vacation with her family, but she's starting to miss her daily math homework. Luckily, her older brother decided to quench her intellectual thirst, and gave her the following problem.

A *valid expression* is defined recursively as follows:

- the string `?` is a valid expression which represents a number.
- if A and B are valid expressions, then so are `min(`A`,`B`)` and `max(`A`,`B`)`, where the former represents a function returning the smaller of its two arguments, while the latter represents a function returning the larger of its two arguments.

For example, expressions `min(min(?,?),min(?,?))` and `max(?,max(?,min(?,?)))` are valid according to the definition above, but expressions `??`, `max(min(?))` and `min(?,?,?)` are not.

Helena is given a valid expression containing a total of N question marks. Each question mark is to be replaced with a number from the set $\{1, 2, \ldots, N\}$ in such a way that each number from this set appears exactly once in the expression. In other words, the question marks are replaced by a permutation of the numbers from 1 to N.

Once the question marks have been replaced by numbers, the expression can be evaluated and its value will be an integer between 1 and N. Considering all the ways of assigning numbers to question marks, how many different values can Helena obtain after evaluating the expression?

Input

The first and only line contains a single valid expression.

Output

Output a single integer between 1 and N, the number of different values obtainable by evaluating the expression.

Scoring

In all subtasks it holds that $2 \leq N \leq 1\,000\,000$.

Subtask	Score	Constraints
1	10	$N \leq 9$
2	13	$N \leq 16$
3	13	Each function in the expression has at least one question mark as an argument.
4	30	$N \leq 1000$
5	34	No additional constraints.

Examples

input	input	input
`min(min(?,?),min(?,?))`	`max(?,max(?,min(?,?)))`	`min(max(?,?),min(?,max(?,?)))`
output	**output**	**output**
`1`	`2`	`3`

Clarification of the first example:
No matter how the numbers are assigned, the value of the resulting expression will always equal to the minimum of the set $\{1, 2, 3, 4\}$, which is 1. Therefore, there is only one possible value.

Clarification of the second example:
The numbers 3 and 4 can be obtained as `4=max(4,max(3,min(2,1)))` and `3=max(3,max(2,min(1,4)))`. It can be shown that values 1 and 2 are not attainable and so the answer is 2.

Task: Prize

"*Living on the edge!*" is a brand new TV game show that targets graph theory enthusiasts as its main audience. During each episode, the host presents a new problem for the contestants to solve. The contestant who solves the problem wins an all-inclusive trip to the Croatian coast, including a guided (Eulerian) tour of the famous Walls of Dubrovnik, as the grand prize.

Tomislav was fortunate enough to get accepted as a contestant in the next episode and immediately began his preparations. He spent countless nights at the library reading up on the most obscure theorems. One night, he accidentally fell asleep and began dreaming about his appearance on the show. Upon waking, he vividly remembered the presented problem and his inability to solve it. The problem went as follows.

The game show host drew two rooted *trees*[1], both consisting of N nodes labelled with integers from 1 to N. The trees themselves are labelled with integers 1 and 2. The host proceeded by stating that both trees are weighted, with positive weights, but the edge weights are purposefully kept secret. After that, Tomislav got the opportunity to choose any subset of node labels, as long as the size of that subset is exactly K.

After Tomislav chose the said subset, he was able to ask at most Q questions of the form (a, b), where a and b are node labels. For each question, the host answered with an ordered quadruple $(d_1(l_1, a), d_1(l_1, b), d_2(l_2, a), d_2(l_2, b))$, where $d_t(x, y)$ represents the *distance*[2] between nodes labelled x and y in tree t, and l_t represents the label of the *lowest common ancestor*[3] of nodes labelled a and b in tree t.

In order to win the prize, Tomislav had to answer a bunch of similar questions posed by the game show host. More specifically, he had to answer exactly T questions of the form (p, q), where p and q are node labels **belonging to Tomislav's chosen subset**. For each question, Tomislav had to respond with the distance between nodes p and q in both trees, i.e. he had to provide the ordered tuple $(d_1(p, q), d_2(p, q))$.

Your task is to help Tomislav with his preparations by writing a program that would solve the problem presented in his dream.

Interaction

This is an interactive task. Your program must communicate with a program made by the organizers which takes the role of the game show host. Of course, your program should take the role of Tomislav and ensure he wins the grand prize.

Your program should first read the parameters N, K, Q and T from the task description. These are given as four space-separated integers in the first line of the standard input.

Your program should proceed to read the description of two trees from the task statement. These descriptions are given in two lines, where the first line describes the first tree, while the second line describes the second tree.

Each tree is given as a sequence of N space-separated integers $p_1, p_2, \ldots, p_N$, where $p_i \in \{-1, 1, 2, \ldots, N\}$ represents the parent of node labelled i in the tree, or is equal to -1 if the tree is rooted in the node labelled i.

Your program should then output K different space-separated integers $x_1, x_2, \ldots, x_K$ $(1 \le x_i \le N)$, representing the subset of node labels Tomislav should choose, and *flush* the output afterwards.

Your program may now ask up to Q questions by writing `'? a b'` $(1 \le a, b \le N)$ to the standard output. When your program has finished asking the questions, it should write a single character `'!'` in its own line, and *flush* the output.

[1] simple, connected, acyclic graphs
[2] sum of edge weights on the unique path between the two nodes
[3] node furthest from the root that has both a and b as (not necessarily direct) descendants

After that, your program may obtain the answers to the posed queries by repeatedly reading a line consisting of four space-separated integers $d_1(l_1, a)$, $d_1(l_1, b)$, $d_2(l_2, a)$ and $d_2(l_2, b)$ from the task description.

Your program should proceed by reading all T of the host's questions from the standard input. Each question is given in a single line by two space-separated integers p and q (where $p, q \in \{x_1, x_2, \ldots, x_K\}$) from the task description.

After your program has read all T questions, it should answer each of them by outputting two space-separated integers $d_1(p, q)$, and $d_2(p, q)$ in its own line. After printing all of the answers, your program should *flush* the output one last time.

Note: You can download the sample source code from the judging system that correctly interacts with the program made by the organizers (including the output *flush*), and solves the first example.

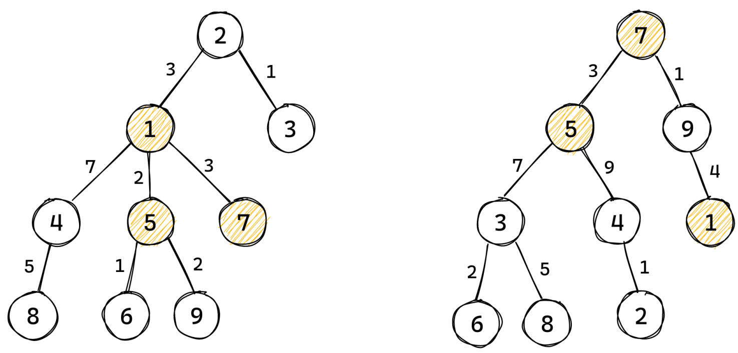
Scoring

It is guaranteed that the hidden edge weights are positive integers not greater than 2 000. Also, in all subtasks it holds that $2 \le K \le 100\,000$ and $1 \le T \le \min(K^2, 100\,000)$.

Subtask	Score	Constraints
1	10	$N = 500\,000$, $Q = K - 1$, trees are identical (including hidden edge weights)
2	25	$N = 500\,000$, $Q = 2K - 2$
3	19	$N = 500\,000$, $K = 200$, $Q = K - 1$
4	22	$N = 1\,000\,000$, $K = 1\,000$, $Q = K - 1$
5	24	$N = 1\,000\,000$, $Q = K - 1$

Example

Output	Input
	9 3 2 3
	2 -1 2 1 1 5 1 4 5
	9 4 5 5 7 3 -1 3 7
1 5 7	
? 1 5	
? 1 7	
!	
	0 2 5 3
	0 3 5 0
	1 7
	7 5
	5 1
3 5	
5 3	
2 8	



1 2

Clarification: In this example, the progam choose the subset $\{1, 5, 7\}$. Then, it asked questions $(1, 5)$ and $(1, 7)$. For the first question, the lowest common ancestors of 1 and 5 are $l_1 = 1$ and $l_2 = 7$, and the answer is $(d_1(1, 1) = 0, d_1(1, 5) = 2, d_2(7, 1) = 5, d_2(7, 5) = 3)$. For the second question, the lowest common ancestors of 1 and 7 are $l_1 = 1$ and $l_2 = 7$, and the answer is $(d_1(1, 1) = 0, d_1(1, 7) = 3, d_2(7, 1) = 5, d_2(7, 7) = 0)$. Finally, the program was asked questions $(1, 7)$, $(7, 5)$, and $(5, 1)$. The answers to these questions are $(d_1(1, 7) = 3, d_2(1, 7) = 5)$, $(d_1(7, 5) = 5, d_2(7, 5) = 3)$, and $(d_1(5, 1) = 2, d_2(5, 1) = 8)$.