ProCo 2017
Advanced Division
Round 1

Problem A. Traveling

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Moana wants to travel from Motunui to Lalotai. To do this she has to cross a narrow channel filled with rocks. The channel is represented by a 3 by n grid. Each cell contains a single bit: 1 if there is a rock at that cell and 0 if there isn't.

Moana's raft can only move to vertically or horizontally adjacent cells. Given the locations of rocks, determine if she can cross the channel, that is, if she can start from some cell on the left and get to some cell on the right without hitting any rocks.

Input

The first line contains n ($2 \leq n \leq 2 \cdot 10^4$), the width of the river in cells. The next 3 lines contain n space separated integers (either 0 or 1), denoting whether or not there is a rock at the cell.

Output

Print "YES" if Moana can cross the channel safely. Otherwise print "NO" (without the quotes).

Examples

standard input	standard output

`3`
`1 0 1`
`0 0 1`
`1 0 0`
	`YES`

`2`
`0 1`
`0 1`
`1 1`
	`NO`

Note

In the first sample, Moana can start in the middle cell on the left and go right, down, and then finally to the right.

In the second sample, all the cells on the right side have rocks, so there is no way to end on any cell on the right.

Problem B. Musical chairs

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

There are n people in Moana's tribe ($1 \leq n \leq 10^6$), and they have decided to play musical chairs. However, standard musical chairs is really boring, so they decided to play a very special variant of musical chairs.

The game proceeds in rounds. There are n chairs in a row, and every chair has a person sitting in it. Each person and each chair has a unique label from 1 to n.

In each round, there are 4 steps. Everyone starts seated in some chair.

1. Each person writes down the chair number they are sitting on a piece of paper. Let the chair number written down by person i be c_i.
2. Each person stands up, walks to the chair corresponding to their own personal number, and puts the paper they wrote on under the chair. This corresponds to each person i putting a piece of paper with c_i written on it under chair i.
3. Each person goes back to the original chair they were sitting in at the beginning of the round (person i will go back to chair c_i). Then each person reads the number on the paper under their chair. Let the chair number read by person i be f_i.
4. Each person goes and sits in the chair corresponding to the number they just read. Person i will sit in chair f_i.

After each round, the papers under the chairs are discarded.

These rounds are repeated over and over, and the game ends when every person ends up in a chair with their number (that is, for all i, person i sits in chair i). These n people have been playing this game for a very long time, and they want to know if it will ever end. Can you figure out if the game will end?

Input

The first line of the input contains a single integer n - the number of chairs and the number of people.

The second line contains n integers, $a_1, \ldots, a_n$ ($1 \leq i, a_i \leq n$). This means that the label of the person sitting in chair i is a_i. You are guaranteed that every chair has exactly one person sitting in it.

Output

On a single line, print "YES" if the game will end, and "NO" if it will never end (without the quotes).

Examples

standard input	standard output

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

`4`
`2 3 1 4`
	`NO`

Note

In the first sample, after one round everyone is in the seats corresponding to their number.

In the second sample, after one round, the people sitting in chairs 1,2,3,4 are 3,1,2,4 respectively. After the next round, it goes back to the original configuration. Thus, the game never terminates.

Problem C. Raft Covering

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Moana needs to cover her raft with banana leaves.

Her raft is represented by $2 \times n$ grid, and she has available a bunch of banana leaves of dimension 1×2. She wants to lay all banana leaves on the raft such that no two banana leaves overlap and no part of a banana leaf is hanging off the edge of the raft. She is allowed to place leaves horizontally or vertically. How many ways can she entirely cover her raft with banana leaves in this way?

Moana already knows the answer is the n-th Fibonacci Number $F(n)$, where F is defined recursively as: $F(0) = 1, F(1) = 1, F(i) = F(i-1) + F(i-2)$ for $i > 1$.

However, if it were that easy Moana wouldn't have come to you for help. Moana's raft is no ordinary raft. There are k locations on her raft where she cannot cover with banana leaves (because something else must go there). You are given the coordinates of these locations. So the real task is to count the number of ways she can cover all parts of her raft except the k locations in the manner described above.

To make the problems easier, you only need to output the answer modulo $10^9 + 7$. You can also assume that no two forbidden locations lie in the same column.

Input

The first line of input consists of two integers n ($1 \leq n \leq 10^6$) and k ($1 \leq k \leq 10^5$).

The following k lines each consists of two integers c ($1 \leq c \leq n$) and r (0 or 1), indicating a forbidden location at column c (1 - indexed), row r (0 - indexed). All c's in the input are pairwise distinct.

Output

The number of different ways to cover the raft with banana leaves, mod $10^9 + 7$. Notice that this number can be 0.

Examples

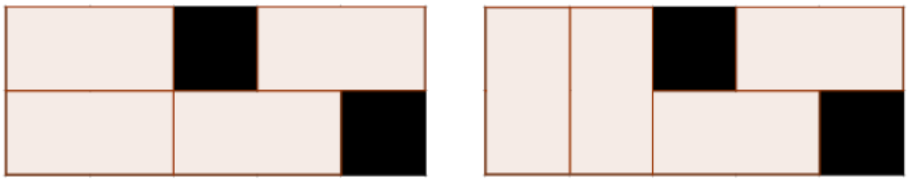
standard input	standard output

5 2
3 0
5 1
	2

2 2
2 0
1 1
	0

Note

In the first sample, there are two ways to cover the raft:



In the second sample, there are no ways to even place a leaf anywhere, and there are two empty spots, so there is no way to cover the raft.

Problem D. Matrix Multiplication

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Moana has math homework. She needs help. Her problem is:

Given two matrices A and B, each of size $n \times n$, $(1 \leq n \leq 1000)$, figure out the matrix product AB, which is another matrix of size $n \times n$.

The multiplication of two matrices is defined as following: In AB, the entry in row i and column j of the resulting matrix is defined to be the dot product of the row i of A and the column j of B. More formally, if A_{ij} is cell (i, j) in matrix A and similary for B, then $AB_{ij} = \sum_{k=1}^{n} A_{ik}B_{kj}$.

However there is something special about matrix A. First, A is a 0-1 matrix, meaning that each element in A is either 0 or 1. Second, there are only at most 100 appearances of 0 in A, and all the rest of elements are, by definition, 1's.

Each element in B is an integer between 0 and 10^4.

Note that since these input matrices can be very large, so it is recommended to use fast input/output methods: for example, you should use `scanf/printf` instead of `cin/cout` in `C++` and `BufferedReader/PrintWriter` instead of `Scanner/System.out` in `Java`.

Input

The first line of input consists of a single integer, n $(1 \leq n \leq 1000)$, denoting the size of the matrices.

The next n lines, each with n space-separated integers, describe matrix A. $(0 \leq A_{ij} \leq 1)$. It is guaranteed that at most 100 entries in A are 0's. The rest are 1's.

The last n lines, each with n space-separated integers, describe matrix B. $(0 \leq B_{ij} \leq 10^4)$.

Output

Output n lines, each with n space-separated integers, describing the resulting matrix of the product AB.

Example

standard input	standard output

```
4
0 1 1 0
0 1 1 1
0 0 1 1
1 0 0 0
3 5 1 9
4 9 2 6
3 5 9 9
4 1 2 2
```
	```
	7 14 11 15
	11 15 13 17
	7 6 11 11
	3 5 1 9
	```

Problem E. MIV

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 megabytes

Moana has just invented a new text editor! She decided to call it Moana's Island Volcano (MIV) because she likes islands and volcanoes.

MIV operates on a single string. The string can contain only lowercase or uppercase alphabets a-z and A-Z, numbers 0-9, and underscores. Let the length of the text be L, where ($1 \le L \le 2 \cdot 10^3$). There is also a cursor C that represents the current position in the text. The cursor is placed between and around characters in the text, so a cursor position of C means that the cursor is just before the character at index C in the text. The cursor is also allowed to be at the end of the text. You are guaranteed that the cursor must always be at a valid position, so $0 \le C \le L$ at all times.

MIV also knows what words are. Words are the maximal contiguous parts of the text separated by underscores. For example, the words in the text below are marked by asterisks underneath:

```
_Hello__world__Said123the______potato_3_times
 *****  *****  **********      ****** * *****
```

MIV allows you to perform the following operations that move the cursor:

- `h`: Moves the cursor left if it is not already in the beginning. The new cursor position is $\max(0, C-1)$.
- `l`: Moves the cursor right if it is not already at the end. The new cursor position is $\min(L, C+1)$.
- `w`: Moves the cursor to the next beginning of a word on the right of the cursor. The beginning of a word is the cursor position just before the first character of the word. If the cursor is currently at the beginning of a word, it still moves to the next beginning of a word on its right. If there are no more words to the right of the cursor, then the cursor moves to position L, the end of the text.
- `b`: Moves the cursor to the previous beginning of a word on the left of the cursor. If the cursor is currently at the beginning of a word, it still moves to the beginning of the previous word. If there are no more word beginnings to the left of the cursor, then the cursor moves to position 0, the very beginning of the text.

Apart from just moving the cursor around, MIV also has some operations that edit the text:

- `i [string]`: Insert the string (which contains only valid characters) at the current cursor position. After the string is inserted, the cursor also moves to the position right after the inserted string.
- `x`: Delete the character to the right of the cursor. That is, if the cursor is at position C, the character at index C is deleted, and the cursor stays at the same position. If the cursor is at the end of the text at position L, nothing happens.
- `d [movement]`: Delete the characters between the current cursor position and where the cursor would move if the movement is performed. Let $C_1 = C$ be the current cursor position, and let C_2 be the resulting position of the cursor if the movement is performed. Then assume $C_1 \le C_2$, otherwise swap C_1 and C_2. Delete the characters from index C_1 to C_2, including C_1 but not including C_2. If the cursor does not move, then no characters are deleted. After the characters are deleted, the new position of the cursor is $\min(C_1, C_2) = C_1$.

Sadly, MIV is just an idea at this point. Help Moana write MIV! Given an initial text S and a sequence of Q ($1 \le Q \le 1000$) operations, print the resulting text after executing the operations. The cursor begins at position 0. You are guaranteed that the length of the text at any time (before and after all the operations) is at most 1000.

Input

The first line contains the initial text S, containing only valid characters. You are given that $0 \leq |S| \leq 1000$. The second line contains a single integer Q, representing the number of operations. The next Q lines contain 1 operation each. The operations are defined as above.

Output

Print a single line containing the resulting string. If the resulting string is empty, print an empty line.

Example

standard input	standard output

`__hello_world_123__`
`18`
`l`
`d w`
`w`
`w`
`b`
`b`
`d b`
`h`
`i __what__`
`x`
`w`
`w`
`d w`
`b`
`b`
`b`
`d b`
`i 123nice456`
	`123nice456what__ello_world_`

Problem F. Chores

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Moana has n ($1 \leq n \leq 5 \cdot 10^5$) chores. Each chore has l_i subtasks. Each chore has a base completion speed v_i (meaning Moana can complete v_i subtasks in that chore in 1 unit of time), but also has a stress factor s_i. Because Moana is stressed by all the chores that she still has yet to do, her completion speed is divided by the sum of all stress factors of chores she hasn't done. Hence, if there are two problems and you complete problem 2 before problem 1, then the total time is $\frac{l_2}{(v_2/(s_1+s_2))} + \frac{l_1}{(v_1/s_1)}$ and so on.

Moana can choose to do the chores in any order she wants. Your task is to find the minimum amount of time for Moana to do all her chores.

Input

The first line of input contains a single integer n - the number of chores ($1 \leq n \leq 5 \cdot 10^5$).

The next n lines contain 3 space-separated integers: l_i, v_i, s_i, ($1 \leq l_i, v_i, s_i \leq 10^6$).

Output

Output a single real number — the minimum amount of time it takes to solve complete all the chores. Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6}.

Namely: let's assume that your answer is a, and the answer of the jury is b. The checker program will consider your answer correct, if $\frac{|a-b|}{\max(1,b)} \leq 10^{-6}$.

Example

standard input	standard output

2
2 1 2
2 2 1
	7.000000

Note

Either order of doing her chores will result in the same amount of time spent, which is 7.

Problem G. Islands

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

The Polynesian archipelago has N islands ($1 \leq N \leq 200$), numbered from 1 to N. Motunui is island 1. Moana has recently discovered that there is actually an underwater system of currents that links the islands. Namely, between any two islands i and j, where $i \neq j$, there can be a current flowing from i to j or not. There can also be a current flowing in the opposite direction that is completely independent of the other current.

Moana would like to know if it is possible to reach all the islands from Motunui just by following the currents. If so, then the configuration of currents is helpful. However, these underwater currents are so fickle and fast-changing that it is impossible to measure them all at once. The best that she can do is to count the number of helpful configurations, i.e. those that would allow her to reach any island from island 1 by following the currents.

Two configurations of the currents are different if there is some pair i, j such that current is flowing from i to j in exactly one of the two configurations. Help Moana count the number of different helpful configurations of currents. Since the number of configurations can be very large, output it modulo a given prime P ($1 \leq P \leq 10^8$).

Input

There is a single line containing the two integers N and P separated by a space, where $1 \leq N \leq 200$ and $1 \leq P \leq 10^8$.

Output

Print one integer, the total number of configurations such that all islands are reachable from island 1, modulo P.

Examples

standard input	standard output
3 10000079	32
5 10000079	745472
10 10000079	1969918

Note

In the first sample case, there are three islands. We consider three cases. Let $a \to b$ denote that a current from a to b exists and $a \not\to b$ denote that a current from a to b does not exist.

If $1 \to 2$ and $1 \to 3$, then Moana can reach all of the other islands from 1, so we can choose a subset of the remaining 4 currents in $2^4 = 16$ ways.

If $1 \not\to 2$ and $1 \to 3$, then we must have $3 \to 2$ so Moana can reach island 2. The remaining 3 currents can be chosen or not chosen in $2^3 = 8$ ways.

If $1 \not\to 3$ and $1 \to 2$, then we must have $2 \to 3$ so Moana can reach island 3. The remaining 3 currents can be chosen or not chosen in $2^3 = 8$ ways.

Thus, the total number of configurations is $16 + 8 + 8 = 32$.

Problem H. Stuck

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Hei-Hei is such a helpless chicken that Moana needs to put obstacles on her raft to prevent Hei-Hei from falling into the ocean.

Moana notices that Hei-Hei's body looks like a square from above. In addition, Hei-Hei only moves up, down, left, right or diagonally, but never rotates its body. Hence, you can treat Hei-Hei as an axis-aligned square with side length $2s$ ($1 \le s \le 10^5$) on Moana's raft, which is a 2D plane.

Moana has placed n obstacles ($1 \le n \le 1000$) on her raft. Each obstacle is an axis-aligned rectangle, defined by its coordinates x_1, y_2, x_2, y_2. You are given that $-10^5 \le x_1, y_1, x_2, y_2 \le 10^5$ and $x_1 < x_2$, $y_1 < y_2$ for each obstacle. The obstacles might intersect each other. When walking around on the raft, Hei-Hei can come into contact with these obstacles, but not pass through them.

Let $x_{\min}$ and $x_{\max}$ be the minimum and maximum x-coordinate covered by a rectangle, and let $y_{\min}$ and $y_{\max}$ be the same for the y-coordinate. These are the only obstacles on the raft, so Hei-Hei can fall off the raft if and only if it can walk from its starting position to some position outside of the rectangle $x_{\min}, y_{\min}, x_{\max}, y_{\max}$.

Help Moana find out if there is a safe spot for Hei-Hei that would prevent it from ever falling off the raft.

Input

The first line contains two integers, n and s. $1 \le n \le 1000$, $1 \le s \le 10^5$. The next n lines each contain four integers x_1, y_1, x_2, y_2, representing the coordinates of a rectangle, where $-10^5 \le x_1, y_1, x_2, y_2 \le 10^5$ and $x_1 < x_2$, $y_1 < y_2$.

Output

Print a single line containing only the word "YES" if Moana can find a safe spot, or "NO" if there is no such safe spot (without the quotes).

Examples

standard input	standard output

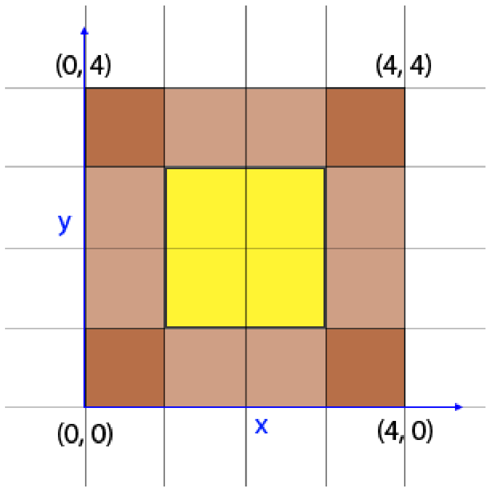
4 1
0 0 4 1
0 3 4 4
0 0 1 4
3 0 4 4
	YES

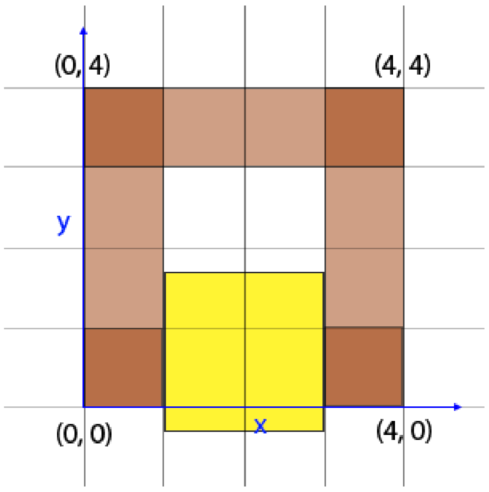
5 1
0 0 1 1
3 0 4 1
0 3 4 4
0 0 1 4
3 0 4 4
	NO

4 1
0 1 1 6
1 5 4 6
1 1 7 2
6 2 7 3
	NO

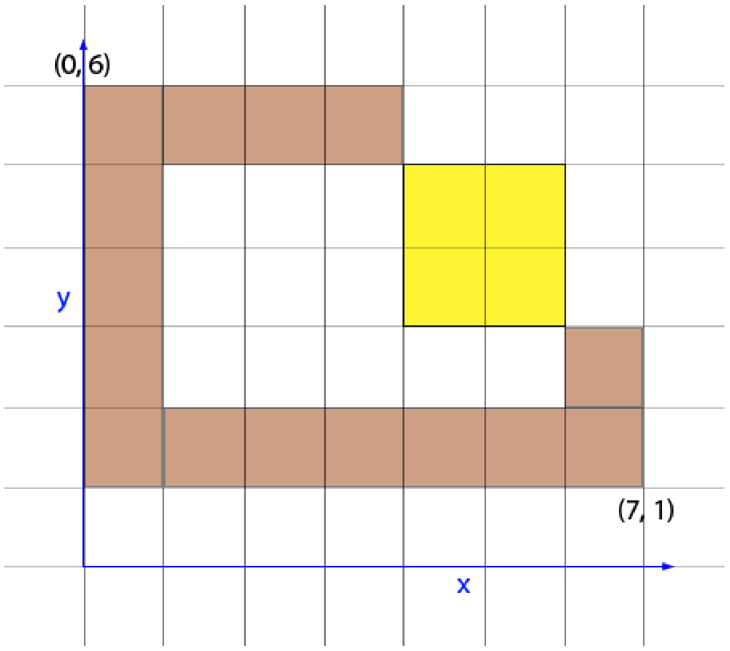
Note

The pictures below depict the first two sample cases:





In the first test case, Hei-Hei (the yellow square) can be placed in the middle of the four boundary walls, and it will be safe. In the second test case, even if Hei-Hei is initially placed in the center, it is able to escape through the hole in the bottom and escape. The third test case looks like the following:



In this case, Hei-Hei can still escape through the hole in the corner. Note that Hei-Hei fits exactly through the gap, but if it were any smaller, Hei-Hei would not be able to escape.

Problem I. Coconut trees

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

On the island of Motunui, there are n coconut trees located at various positions on a line. Tree i is located at position x_i. All the given positions of the trees are distinct.

The trees are equal, i.e. each tree has height h. Due to the wind, when a tree is cut down, it either falls left with probability p, or falls right with probability $1-p$. If a tree hits another tree while falling, that tree will fall in the same direction as the tree that hit it. A tree can hit another tree only if the distance between them is strictly less than h.

For example, imagine there are 4 trees located at positions 1, 3, 5 and 8, while $h = 3$ and the tree at position 1 falls right. It hits the tree at position 3 and it starts to fall too. In it's turn it hits the tree at position 5 and it also starts to fall. The distance between 8 and 5 is exactly 3, so the tree at position 8 will not fall.

As long as there are still trees standing, Moana will select one of the remaining standing trees **uniformly at random**. Selected tree is then cut down. If there is only one tree remaining, Moana always selects it. As the ground is covered with grass, Moana wants to know the expected total length of the ground covered with fallen trees after she cuts them all down.

Input

The first line of the input contains two integers, n ($1 \le n \le 5 \cdot 10^5$) and h ($1 \le h \le 10^8$) and a real number p ($0 \le p \le 1$), given with no more than six decimal places.

The second line of the input contains n integers, $x_1, x_2, \cdots, x_n$ ($-10^8 \le x_i \le 10^8$) in no particular order.

Output

Print a single real number — the expected total length of the ground covered by coconut trees when they have all fallen down. Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6}.

Namely: let's assume that your answer is a, and the answer of the jury is b. The checker program will consider your answer correct, if $\frac{|a-b|}{\max(1,b)} \le 10^{-6}$.

Examples

standard input	standard output

`2 2 0.500000`
`1 2`
	`3.25`

`4 3 0.4`
`4 3 1 2`
	`6.8704`

Note

In the first sample, there are 3 scenarios:

1. Both trees falls left. This can either happen with the right tree falling left first, which has $\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$ probability (also knocking down the left tree), or the left tree can fall left and then the right tree can fall left, which has $\frac{1}{4} \cdot \frac{1}{2} = \frac{1}{8}$ probability. Total probability is $\frac{1}{4} + \frac{1}{8} = \frac{3}{8}$.

2. Both trees fall right. This is analogous to (1), so the probability of this happening is $\frac{3}{8}$.

3. The left tree fall left and the right tree falls right. This is the only remaining scenario so it must have $1 - \frac{3}{8} - \frac{3}{8} = \frac{1}{4}$ probability.

Cases 1 and 2 lead to a total of 3 units of ground covered, while case 3 leads to a total of 4 units of ground covered. Thus, the expected value is $3 \cdot (\frac{3}{8}) + 3 \cdot (\frac{3}{8}) + 4 \cdot (\frac{1}{4}) = 3.25$

Problem J. Queries

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

It's tough being the village chief. Moana needs to answer her villagers' questions every day. Help Moana answer her villagers' questions quickly.

There are n seashells on the coastline. The type of each seashell is given by a number from 1 to n. Fortunately, Moana knows everything about her island, including the exact location of all the seashells along the beach. She gives you a sequence of n numbers, where the i-th number is the type of the i-th seashell along the coast, where $1 \leq i \leq n$.

The villagers have m questions. Each question is represented by two integers l, r where $1 \leq l \leq r \leq n$. Consider the types of the seashells starting from the l-th seashell to the r-th seashell. The answer to the question is the maximum number of times that any given type of seashell appears. For example, if the seashell types are $[3, 3, 5, 1, 1, 2, 1, 3, 2]$, then type 1 and 3 appear most frequently at 3 times each, so the answer to the question is 3.

Input

The first line of input contains two space-separated integers n and m ($1 \leq n, m \leq 10^5$), representing the number of seashells and the number of questions.

The next line of input contains n integers $a_1, a_2, \cdots, a_n$, where $1 \leq a_i \leq n$ for all $1 \leq i \leq n$, where a_i is the type of the i-th seashell.

The next m lines contain two space-separated integers l_i and r_i representing question i, where $1 \leq l_i \leq r_i \leq n$ for all $1 \leq i \leq m$.

Output

For each question in the same order as the input, print the answer to the question (a single integer) on a single line by itself.

Examples

`standard input`	`standard output`

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

`40 10`
`16 28 15 31 38 20 39 10 30 18 30 11 1`
`30 20 34 23 11 5 34 16 1 30 31 2 18`
`31 8 31 18 20 31 20 33 18 18 13 8 21`
`18`
`5 22`
`27 40`
`7 14`
`14 30`
`26 29`
`11 24`
`7 27`
`11 31`
`2 29`
`22 35`
	`3`
	`4`
	`3`
	`3`
	`2`
	`3`
	`4`
	`3`
	`4`
	`4`