# Problem A. Lady and Vus the Cossack's Permutation

Vus the Cossack has a strategically important permutation* $a$ of the length $2 \cdot n - 1$. He encrypts the permutation with the array $b$, where $b_i$ is a median† of the subarray $a_1, a_2, \ldots, a_{2 \cdot i - 1}$. Lady found the cipher and is asking you to restore any suitable permutation.

## Input

The first line contains single integer $n$ ($1 \le n \le 100\,000$) — the cipher length.

The second line contains $n$ integers $b_1, b_2, \ldots, b_n$ ($1 \le b_i \le 2 \cdot n - 1$) — the medians.

It is guaranteed that in the tests, the answer always exists.

## Output

In the single line, print $2 \cdot n - 1$ integers $a_1, a_2, \ldots, a_{2 \cdot i - 1}$.

If there are several correct answers, print any.

## Examples

| standard input | standard output |
|---|---|
| 5<br>1 3 3 4 5 | 1 3 9 2 8 4 7 5 6 |
| 4<br>1 2 3 4 | 1 2 3 4 5 6 7 |

## Note

In the first example:

- $b_1 = 1$ is the median of array $(1)$,
- $b_2 = 3$ is the median of array $(1, 3, 9)$,
- $b_3 = 3$ is the median of array $(1, 3, 9, 2, 8)$,
- $b_4 = 4$ is the median of array $(1, 3, 9, 2, 8, 4, 7)$,
- $b_5 = 5$ is the median of array $(1, 3, 9, 2, 8, 4, 7, 5, 6)$.

*A permutation of the length $k$ is a consequence of integers from 1 to $k$ where every number has single occurrence.
†A median of an array of the odd length is the element which is in the middle position of the sorted array.

# Problem B. Vus the Cossack and Lady's Secret

Everyone has their own secrets. Lady promised to tell her New Top Secret (NTS) to Vus the Cossack for his birthday but didn't keep her word. Instead, she came up with a game for Cossack.

The game is played with a deck of cards, each with a numeric label from 1 to $n$. There are two cards with each possible label, making $2n$ cards in total. Vus the Cossack makes $2n$ steps. At each step, he picks the top card or the second-from-top card and pulls it out of the deck. When he picks two cards with the same face value in a row, he scores a point.

If Vus the Cossack scores the most possible number of points, then Lady may tell him the NTS. Help Cossack find out the NTS and write a program that finds the maximum number of points that can be scored.

## Input

The first line contains one integer $n$ ($1 \leq n \leq 100\,000$) — the maximum card label.

The second line contains $2n$ integers $a_1, a_2, \ldots, a_{2n}$ ($1 \leq a_i \leq n$) — the label of the $i$-th card from the top to the end of the deck. It is guaranteed that every label from 1 to $n$ occurs twice.

## Output

The output should consist of a single integer in a single line — the answer to the task.

## Examples

| standard input | standard output |
|---|---|
| 3<br>1 1 3 2 2 3 | 3 |
| 7<br>1 4 2 5 6 2 7 7 1 3 4 6 3 5 | 3 |

## Note

In the first example, you can pull the cards out in this order: $[1, 1, 2, 2, 3, 3]$.

In the second example, you can pull the cards out in the following order: $[4, 2, 5, 6, 2, 7, 7, 1, 1, 4, 6, 3, 3, 5]$.

# Problem C. XOR Path

Given an integer matrix $a$ of size $n \times n$.

Let's assume that the rows of the matrix are numbered from top to bottom from 1 to $n$, and the columns from left to right from 1 to $n$. Let $a_{i,j}$ be the number at the intersection of the $i$-th row and the $j$-th column.

You start on cell $(1,1)$. You can move only down or right. You must end in the cell $(n,n)$. Let $b_1, b_2, \ldots, b_{2n-1}$ be the integers recorded in the cells you visited. The *value* of a path is equal to $b_1 \oplus b_2 \oplus \ldots \oplus b_{2n-1}$[‡]. Your task is to find a path from $(1,1)$ to $(n,n)$ with the **minimum** *value*.

## Input

The first line contains one integer $n$ $(2 \le n \le 20)$ — the size of the matrix.

Each of the following $n$ lines contains $n$ integers $a_{i1}, a_{i2}, \ldots, a_{in}$ $(1 \le a_{ij} \le 1\,000\,000\,000)$.

## Output

Output an integer — the **minimum** possible *value* of path.

## Examples

| standard input | standard output |
|---|---|
| 2<br>1 8<br>2 8 | 1 |
| 4<br>99 146 613 1416<br>513 5810 1515 9616<br>1247 5124 6284 5844<br>1139 6135 6427 1561 | 241 |

## Note

In the first example, the following path is optimal: $(1,1)$, $(1,2)$, $(2,2)$. The numbers written in these cells are $[1,8,8]$. $1 \oplus 8 \oplus 8 = 1$, which will be the answer.

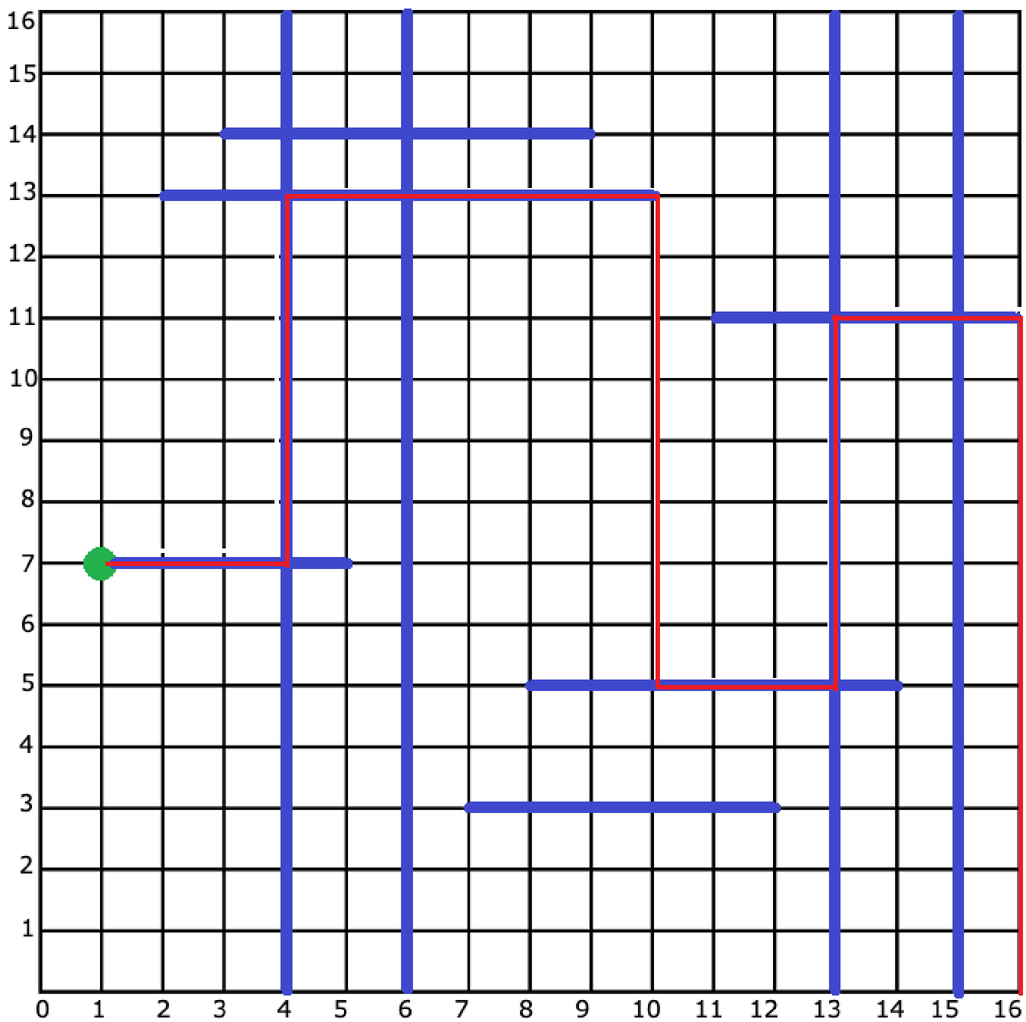
[‡] The expression $x \oplus y$ implies the use of the bitwise operation `XOR` to the numbers $x$ and $y$. This operation exists in all modern programming languages, for example, in `C++` and `Java`, it is marked as "`^`", and in `Pascal` as "`xor`".

# Problem D. Cossack and Lady Witch

Lady finally decided to tell her New Top Secret (NTS) to Vus the Cossack, but it was not without an adventure. The witch turned Cossack into a little man and invited him to play a game. By the way, many consider Lady to be a witch herself and just buy time.

The witch moved the little man into a two-dimensional space that can be represented as the first quarter of the Cartesian coordinate system. This space holds **horizontals** — segments, parallel to the abscissa axis and **verticals** — lines, parallel to the ordinate axis. Notice that the segments have a beginning and an end, and lines do not.

Vus the Cossack can be set as a point. He can move by the horizontals or verticals if the point to which it is set belongs to one or the other, respectively. By horizontals, a man can only move to the right, and by verticals up and down. Also, if the man is at the right point of one of the horizontals, he may jump down and move until he meets another horizontal or lands on the abscissa axis. In the beginning, Cossack stands at the leftmost point of one of a horizontal. When it lands on the abscissa axis, the game ends. Cossacks' goal is to finish the game at the point with the highest possible abscissa.




The sooner the witch lets the Vus go, the sooner the Lady will tell him NTS. Help Cossack and say the biggest abscissa in which he can finish the game.

## Input

The first line contains one integer $n$ ($1 \le n \le 100\,000$) — the number of horizontals.

The second line contains one integer $m$ ($1 \le m \le 100\,000$) — the number of verticals.

The third line contains one integer $k$ ($1 \le k \le n$) — the number of horizontal at the beginning of which is Cossack Vus stands.

The following $n$ rows contain three integers $x_l, x_r, y$ ($1 \le x_l < x_r \le 10^9, 1 \le y \le 10^9$) — left abscissa, right abscissa and the ordinate of the next horizontal. It is guaranteed that no two horizontal segments have common points.

The following line contains $m$ integers $x$ ($1 \le x \le 10^9$) — the abscissa of the next vertical. It is guaranteed that all lines have different coordinates.

## Output

The output should consist of a single integer in a single line — the answer to the task.

## Examples

| standard input | standard output |
|---|---|
| 6<br>4<br>6<br>8 14 5<br>11 16 11<br>3 9 14<br>2 10 13<br>7 12 3<br>1 5 7<br>15 13 6 4 | 16 |
| 10<br>7<br>5<br>23 69 79<br>73 87 23<br>32 63 29<br>21 70 39<br>19 65 20<br>24 38 54<br>27 52 75<br>51 57 93<br>35 73 40<br>83 87 70<br>51 14 80 75 58 87 37 | 87 |

## Note

The picture illustrates the first example. Blue indicates horizontals and verticals, green — the starting point of Cossack Vus, and red — the most optimal route.

# Problem E. Number Plate

Vus the Cossack and Lady like to play the game while driving. They are looking at the number, written on the car number plate, and they want to make it divisible by 9 with a minimal number of operations. The operation is to add or subtract one from any digit. The digits have to always be from 0 to 9. That is, you can't subtract from 0 or add to 9.

The friends soon got tired of the game. That's why they decided to make it harder adding one more rule — through all possible solutions, players have to find lexicographically smallest.[§]

In the Flowland, the number plates are pretty long, so the game is hard to play. That's why the friends are asking you to write a program, which will play instead of them.

## Input

A single line contains numeric string $s$ ($1 \le |s| \le 1\,000\,000$), where $|s|$ is a length of the line. Note, that the string can start with "0".

## Output

In the single line, print the answer to the problem.

## Examples

| standard input | standard output |
|---|---|
| 23 | 27 |
| 228 | 018 |
| 0234 | 0234 |

## Note

In the first example, it is possible to make 00 from 23 using 5 operations, or 27 using 4 operations, which is more profitable.

In the second example, it is possible to make 279 from 228 using 6 operations, or 018 using 3 operations, which is more profitable.

In the third example, 0234 is already divisible by 9, so it is the answer.

[§]A string $a$ is lexicographically smaller than a string $b$ if and only if one of the following holds:

- $a$ is a prefix of $b$, but $a \neq b$;
- in the first position where $a$ and $b$ differ, the string $a$ has a letter that appears earlier in the alphabet than the corresponding letter in $b$.

# Problem F. Lady's Investments

Lady is going to become an investor. She is planning to invest $l$ hryvnias in each of $k$ companies. Lady is a patriot, so she invests only in companies in her country. There are $n$ cities in the country, connected with $m$ roads. It is possible to get from every city to any other using roads. There are $2^n - 1$ companies. For every non-empty set of cities, there is one company, which has an office in every city from the set.

Currently, the situation in the country is unstable, but Lady is clever, and she invests in the company only if one condition is satisfied: in case of any city blocking, every other office will still be connected with roads. Every city $i$, which contains at least one office of a company, which is invested by Lady, brings $p_i$ hryvnias of income. Lady can invest in less than $k$ companies. If she invests in $x$ companies, she will also get extra $(k - x) \cdot l$ hryvnias of income. Help Lady to find maximal income.

## Input

The first line contains four integers $n$, $m$, $k$, $l$ ($1 \le n \le 100\,000$, $n - 1 \le m \le 300\,000$, $1 \le k \le 40$, $1 \le l \le 1\,000\,000\,000$) — the number of cities, the number of roads, the number of companies which Lady plans to invest and how many hryvnias Lady invests in a single company.

The second line contains $n$ integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le 1\,000\,000\,000$) — the income of $i$-th city.

Each of the next $m$ lines contains two integers $u$, $v$ ($1 \le u \ne v \le n$) — the numbers of cities, which have road between them. It is guaranteed that there is only one road between each pair of cities. It is guaranteed that for every city it is possible to get to any other using these roads.

## Output

In the single line, print the answer to the problem.

## Examples

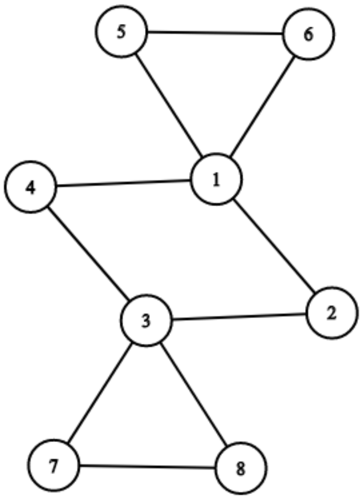
| standard input | standard output |
|---|---|
| 8 10 2 5<br>10 1 10 1 2 2 2 2<br>1 2<br>2 3<br>3 4<br>4 1<br>1 5<br>5 6<br>6 1<br>3 7<br>7 8<br>8 3 | 28 |
| 6 7 3 5<br>10 10 10 10 10 1<br>1 2<br>2 3<br>1 3<br>3 4<br>4 5<br>3 5<br>5 6 | 55 |

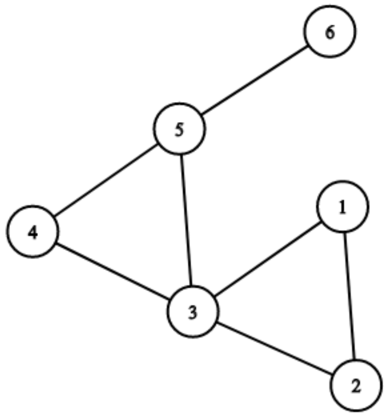
## Note

In the first example, the best option is to invest in the company, which has offices in cities $3, 7, 8$ and company, which has offices in cities $1, 5, 6$.

In the second example, the best option is to invest in the company, which has offices in cities $1, 2, 3$ and company, which has offices in cities $3, 4, 5$. And leave yourself $l$ uninvested in the third company hryvnias.

The pictures for the examples:



# Problem G. Team

Vus the Cossack wants to handle a special operation. His army has a strong hierarchy. Vus himself is a Commander-in-chief, he subordinates generals, which subordinate colonels, etc.

If a soldier has a direct leader, then the direct leader of the direct leader is also the soldier's leader, and his direct leader too, etc. Every soldier $p$ has a fixed salary $c_p$ hryvnias (strange, but sometimes leaders get less than subordinates).

The whole squad budget has $b$ hryvnias. The squad consists of the chief and performers. Vus the Cossack can appoint as a chief himself or any other soldier. Performers can include only chief's subordinates (including chief himself), though chief doesn't necessarily have to be a performer. The squad level is a product of the chief's leadership level $l$ and the number of performers.

Help Vus the Cossack to find the biggest possible squad level, with paying every **performer**, so the total expenses won't exceed $b$.

## Input

The first line contains two integers $n$ and $b$ ($1 \le n \le 100\,000$, $1 \le b \le 1\,000\,000\,000$) — number of soldiers and operation budget.

Each of the next $n$ lines contains three integers $p_i, c_i, l_i$ ($1 \le p_i \le i-1, p_i = 0$, if $i = 1$, $1 \le c_i, l_i \le 1\,000\,000\,000$) — the direct leader of the soldier, or 0, if it is Vus the Cossack, the salary and the leadership level.

## Output

In the single, line print the answer to the problem.

## Example

| standard input | standard output |
|---|---|
| 5 4<br>0 3 3<br>1 3 5<br>2 2 2<br>1 2 4<br>2 3 1 | 6 |

## Note

The best option is to choose a soldier with the number 1 (Vus the Cossack) as a chief and choose soldiers with numbers 3 and 4 as performers.

# Problem H. Vus the Cossack, Lady, and a Disk

Vus the Cossack has already known the Lady's New Top Secret, but he is still interested in playing the games she offers. This time, Lady wants to teach Vus the Cossack to do magic disk operations.

A disk is a circle divided into $k$ equal segments, each of them contains an integer, from 1 to $k$ in increasing order, is inscribed clockwise on each segment on the front side. On the backside, each segment contains an integer, from $-1$ to $-k$ in decreasing order, is inscribed counterclockwise. In addition, in the same segment on the different sides are written two opposite numbers. The upper sector records the number 1.

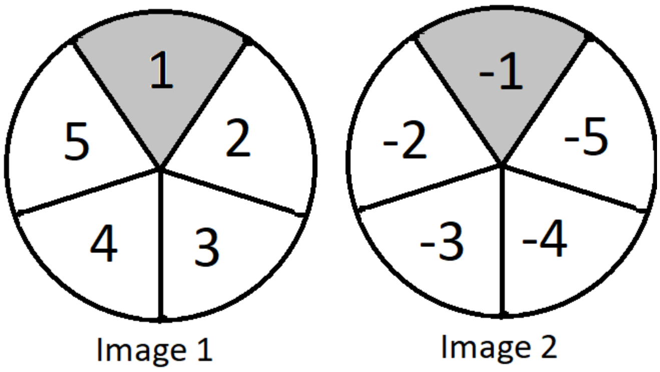



Image 1 shows the front of the disk and image 2 — the back. The upper sector is highlighted in gray.

There is also a deck of $n$ cards. Each card has a label that does not exceed $10^9$ in absolute value. In one turn, Lady scans **all** the cards from start to end. Let the number written on the next card is $a$. If $a > 0$, then the disk rotates clockwise into $a$ segments. If $a < 0$, then the disk rotates counterclockwise into $-a$ segments. If $a = 0$, the disk flips from front to back or from back to the front side, keeping the top sector in place.

Initial information is given about the numbers, written on $n$ cards and $q$ operations, each of which requires two cards to be swapped. After each operation, the Lady makes one turn with the deck of cards received, and Vus the Cossack must say a number that will be written on the upper segment on the side, which will be visible after that turn.

Help Cossack find the number in the upper segment after each turn. Note that before each new turn, the disk returns to its original position, that is, the upper sector records the number 1.

## Input

The first line contains three integers $k$, $n$, and $q$ ($1 \le k \le 10^9, 2 \le n \le 10^5, 1 \le q \le 10^5$) — the number of segments, the cards in the deck and the operations respectively.

The second line contains $n$ integers $a_1, a_2, \ldots, a_n$ ($-10^9 \le a_i \le 10^9$) — the label of the $i$-th card from the top of the deck.

Each of the following $q$ rows contains two integers $x$ and $y$ ($1 \le x < y \le n$) — the positions of the cards that need to be changed during the operation.

## Output

For each operation, output a single number on a separate line — the answer to the task after the next operation.

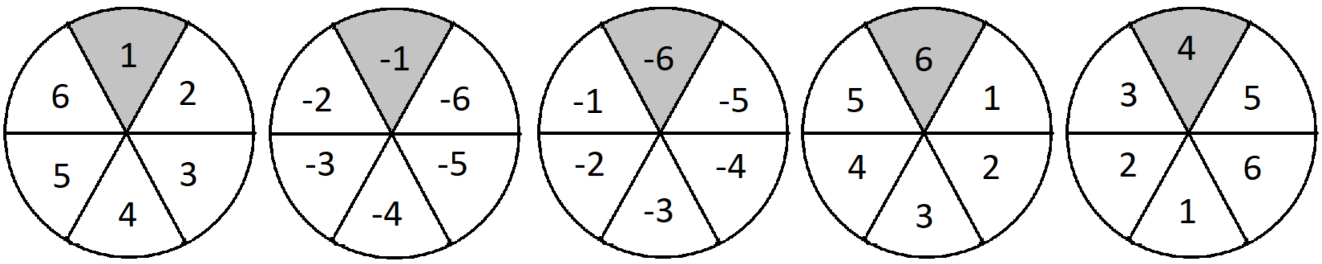
## Examples

| standard input | standard output |
|---|---|
| 6 4 1<br>0 2 0 -7<br>2 4 | 4 |
| 6 7 2<br>3 0 1 -2 0 2 -7<br>2 5<br>3 5 | 2<br>4 |

## Note

The picture below illustrates the moves of the disk in the first example.

After the swapping operation, the deck has the following order 0, −7, 0, 2. The highest segment contains 1. Lady takes the card 0 and now −1 is in the highest segment. After the card −7, there is −6 in the highest segment. After the next zero, 6 become the number written in the highest segment. Finally, after the card 2, there will be 4. It is shown below.



# Problem I. The Duck Doesn't Like Statements

The Big Duck loves problems, but it doesn't like long statements, so it asked me to make the statement as short as possible.

You are given an array $a$ of the length $n$ and one number $k$. Each element of the array is equal to either $1$, $-1$, or $0$. Your task is to replace all zeros with $-1$ or $1$ so that the sum of the elements of the array is **positive** and divisible by $k$.

## Input

The first line contains two integers $n$ and $k$ ($1 \le n, k \le 2 \cdot 10^5$) — the length of the array and the number by which the sum of the elements of the array should be divisible.

The second line contains $n$ integers $a_1, a_2, \cdots, a_n$ ($-1 \le a_i \le 1$) — the elements of the array.

## Output

If it is possible to replace the zeros with either $-1$ or $1$ so that the sum of the elements of the array is positive and divisible by $k$, then print the string "`Yes`" in the first line (without quotes) and in the second line print $n$ integers — the elements of the new array. If there are several possible solutions, print any.

Otherwise, print the string "`No`" (without quotes) in one line.

## Examples

| standard input | standard output |
|---|---|
| 4 2<br>-1 -1 0 0 | No |
| 4 4<br>0 0 -1 0 | No |
| 4 2<br>-1 1 0 0 | Yes<br>-1 1 1 1 |
| 6 2<br>-1 -1 1 1 1 1 | Yes<br>-1 -1 1 1 1 1 |

# Problem J. Subsequences of Subsequences

You have been given a string $s$ of lower-case Latin letters. Your task is to find the number of *good* non-empty subsequences¶ of this line.

A subsequence is called *good* if all its non-empty subsequences are palindromes‖.

Two sequences are considered different if they can be obtained by deleting characters with different indexes. That is, even if the sequence strings are the same, but the character indices to be deleted are different, then these sequences are considered different.

## Input

The first line contains one string $s$ ($1 \le |s| \le 10^5$). All its characters lower-case Latin letters.

## Output

Print an integer — the number of *good* non-empty subsequences of $s$ modulo $10^9 + 7$ (1000000007).

## Examples

| standard input | standard output |
|---|---|
| abcd | 4 |
| abacaba | 19 |

## Note

In the first example, there are 4 *good* subsequences: "a", "b", "c", "d".

¶A sequence $a$ is a subsequence of a sequence $b$ if $a$ can be obtained from $b$ by deletion of several (possibly, zero or all) elements.

‖Palindrome — a string that reads the same backward as forward.

# Problem K. Interesting

Lady has just created a problem.

There is an array $a$ with $n$ non-negative integers.

You are allowed to swap a pair of elements, but you are not allowed to change position of one element twice. As a result of these operations, you will get an array $b$.

Let's define an array $c$ of the size $n$ where $c_i = \min(a_i, b_i)$.

Answer to this problem is the minimum possible value of $c_1 \oplus c_2 \oplus \ldots \oplus c_n$**.

Lady has $m$ subarrays, and for each of them, she wants to know the answer. Please, help her to get it.

Consider, that all these $m$ tasks are independent.

## Input

The first line contains two integers $n$ and $m$ ($1 \le n, m \le 150\,000$) — the size of the array and the number of subarrays.

The second line contains $n$ integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i < 2^{30}$) — the numbers.

Each of next $m$ lines contains two integers $l$ and $r$ ($1 \le l \le r \le n$), which means that you want to get answer for the subarray $[a_l, a_{l+1}, \ldots, a_r]$.

## Output

Output $m$ integers — answers for tasks.

## Example

| standard input | standard output |
|---|---|
| 6 4<br>1 2 3 4 5 6<br>1 1<br>3 5<br>4 6<br>1 6 | 1<br>2<br>4<br>0 |

## Note

In the first task, it is impossible to change the positions of the elements of the array.

In the second task, to get the optimal result, we don't need to swap any elements of the array.

In the third task, to get the optimal result, we need to swap the elements of the array with the values 5 and 6.

In the fourth task, one of the possible solutions is to swap the following pairs $(1, 3)$, $(2, 6)$, and $(4, 5)$.

**The expression $x \oplus y$ implies the use of the bitwise operation `XOR` to the numbers $x$ and $y$. This operation exists in all modern programming languages, for example, in `C++` and `Java`, it is marked as "`^`", and in `Pascal` as "`xor`".

# Problem L. Cossack's the First Game

Vus the Cossack finally came up with his first game, but he has never shown it to anyone. He is asking you to test this game.

You have been given an array $a$ of the length $n$. The number $m$ is also given. All numbers in the array are positive integers that do not exceed $m$. Each number from 1 to $m$ has its own price. To win the game, you need to find a subsegment of the array with the maximum price. The subsegment price is the sum of the prices of the numbers that occur on the subsegment exactly 1 time.

Help Cossack find the subsegment with the maximum price for the given array and number prices.

## Input

The first line contains two integers $n$ and $m$ ($1 \le m \le n \le 200\,000$) — the size of the array and the maximum number in the array.

The second line contains $n$ integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le m$) — the numbers of array. It is guaranteed that every number from 1 to $m$ occurs at least once.

The third line contains $m$ integers $c_1, c_2, \ldots, c_m$ ($1 \le c_i \le 1\,000\,000$) — the prices of the numbers.

## Output

The output should consist of a single integer in a single line — the answer to the task.

## Examples

| standard input | standard output |
|---|---|
| 7 3<br>3 1 3 3 2 2 1<br>12 10 1 | 22 |
| 7 5<br>1 3 5 1 2 4 2<br>18 18 14 10 1 | 61 |

## Note

In the first example, the segment with the maximum price is $[6 \ldots 7]$.

In the second example, the segment with the maximum price is $[2 \ldots 6]$.