Problem A. Birthday Party

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

NOTE: Please note that to solve the problem you would need to implement mergesort algorithm from scratch. Otherwise, your submission will not be graded. Use of any other libraries apart from "stdio.h" is not allowed. Otherwise, your submission will not be graded.

Ishu has plenty of deadlines to complete by the end of the day. But, she also needs to dress up for a friend's birthday party, because of which she can't focus on deadlines. Luckily, she got a close "friend" Rohu who would happily help her. So, Rohu decides to take on the deadlines, and for this, he comes up with the idea of arranging the deadlines first based on preference (easiness in Rohu's words).

However, it is not as simple as it appears as Rohu is quite lazy and would want to make this arrangement using the minimum number of swaps. Help Rohu arrange the deadlines as this might be the only chance he would have of impressing Ishu.

Formally, given N distinct integers $d_1, d_2, ...d_i, ...d_N$. You must report the minimum number of swaps required to order $D_1, D_2, ...D_i, ...D_N$, where $D_1 < D_2 < ... < D_N$.

Input

The first line contains a single integer n ($1 \le n \le 10^5$) — the number of elements in the array.

The second line contains n distinct integers $d_1, d_2, \ldots, d_n$ ($1 \le a_i \le 10^6$).

Output

Print minimum numbers of swaps.

Examples

standard input	standard output
4 1 2 4 3	1
4 2 1 4 3	2
5 5 1 10 7 2	3

Problem B. Sort the Matrix

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

NOTE: Please note that to solve the problem you would need to implement quicksort algorithm from scratch. Otherwise, your submission will not be graded. Use of any other libraries apart from "stdio.h" is not allowed. Otherwise, your submission will not be graded.

Given an m x n 2D matrix A consisting of only $0s$ and $1s$, sort the columns of A such that the sums of columns are in non-decreasing order. In case two columns have equal sums, prioritise the column with a lower index in the original input. Print the final binary matrix after sorting.

Constraints

- $1 \leq m, n \leq 10^4$
- $1 \leq m * n \leq 10^6$

Input

The first line of input consists of two integers m and n respectively. Subsequently, there are m lines with n entries in each line representing the binary 2D matrix.

Output

Print the final binary matrix after sorting according to the criteria specified.

Examples

standard input	standard output
4 5 0 0 1 1 0 1 1 1 0 1 0 1 1 1 0 1 0 1 1 0	0 0 0 1 1 1 1 1 0 1 0 0 1 1 1 0 1 0 1 1
8 3 0 0 0 1 0 1 1 1 0 0 0 1 1 1 1 0 1 0 0 0 0 1 1 1	0 0 0 1 0 1 1 1 0 0 0 1 1 1 1 0 1 0 0 0 0 1 1 1

Note

The cumulative sums of each of the columns are - 2, 2, 4, 3, 1 which on sorting turn out to be 1, 2, 2, 3, 4 (note that column 0 and column 1 have the same sum but column 0 comes first due to smaller index). Based on this sorted sequence, the columns are re-arranged to give the final output as shown.

Problem C. Cupcakes(Easy Version)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

NOTE: the easy and the hard version of the problem vary on the basis of the constraints given. Use of any other libraries apart from "stdio.h" is not allowed. Otherwise, your submission will not be graded.

Lawrence has bought N cupcakes for Kone. But it turns out that Kone doesn't want to eat them. Kone will eat the cupcakes only if Lawrence solves the problem given by him. Kone arranges the cupcakes in a row and assigns a random value A_i to the i^{th} cupcake. The values assigned to the cupcakes will be given to you in the form of an array A. He asks Lawrence to arrange the cupcakes in the ascending order of their values. Now he gives him Q queries to answer. In each query Kone gives three integers L, R, K where L, R are indices of the array. Lawrence needs to print the number of subarrays in the subarray $A[L..R]$ which have K distinct elements.

Constraints:

- $1 \leq A_i \leq 100$
- $1 \leq N \leq 3 * 10^3$
- $1 \leq Q \leq 50$
- $0 \leq L \leq R \leq N - 1$
- $1 \leq K \leq N$

Input

The first line contains two spaced integers N and Q. The next line contains N spaces integers A_i denoting the values of the cupcakes. Each of the next Q lines contain 3 integers L, R, K.

Output

Print the number of subarrays in the subarray $A[L..R]$ which have K distinct elements for every query in separate line.

Examples

standard input	standard output
7 2 60 66 94 56 59 8 67 4 5 7 5 6 6	0 0
5 1 1 2 4 2 5 1 3 1	4

Problem D. Cupcakes(Hard Version)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

NOTE: the easy and the hard version of the problem vary on the basis of the constraints given. Use of any other libraries apart from "stdio.h" is not allowed. Otherwise, your submission will not be graded.

Lawrence has bought N cupcakes for Kone. But it turns out that Kone doesn't want to eat them. Kone will eat the cupcakes only if Lawrence solves the problem given by him. Kone arranges the cupcakes in a row and assigns a random value A_i to the i^{th} cupcake. The values assigned to the cupcakes will be given to you in the form of an array A. He asks Lawrence to arrange the cupcakes in the ascending order of their values. Now he gives him Q queries to answer. In each query Kone gives three integers L, R, K where L, R are indices of the array. Lawrence needs to print the number of subarrays in the subarray $A[L..R]$ which have K distinct elements.

Constraints:

- $1 \leq A_i \leq 100$
- $1 \leq N \leq 5 * 10^4$
- $1 \leq Q \leq 50$
- $0 \leq L \leq R \leq N - 1$
- $1 \leq K \leq N$

Input

The first line contains two spaced integers N and Q. The next line contains N spaces integers A_i denoting the values of the cupcakes. Each of the next Q lines contain 3 integers L, R, K.

Output

Print the number of subarrays in the subarray $A[L..R]$ which have K distinct elements for every query in separate line.

Examples

standard input	standard output
7 2 60 66 94 56 59 8 67 4 5 7 5 6 6	0 0
5 1 1 2 4 2 5 1 3 1	4