Codeforces Round #344 (Div. 2)

A. Interview

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

Blake is a CEO of a large company called "Blake Technologies". He loves his company very much and he thinks that his company should be the best. That is why every candidate needs to pass through the interview that consists of the following problem.

We define function $f(x, l, r)$ as a bitwise OR of integers $x_l, x_{l+1}, \ldots, x_r$, where x_i is the i-th element of the array x. You are given two arrays a and b of length n. You need to determine the maximum value of sum $f(a, l, r) + f(b, l, r)$ among all possible $1 \le l \le r \le n$.

Input

The first line of the input contains a single integer n ($1 \le n \le 1000$) — the length of the arrays.

The second line contains n integers a_i ($0 \le a_i \le 10^9$).

The third line contains n integers b_i ($0 \le b_i \le 10^9$).

Output

Print a single integer — the maximum value of sum $f(a, l, r) + f(b, l, r)$ among all possible $1 \le l \le r \le n$.

Examples

input
5 1 2 4 3 2 2 3 3 12 1
output
22

input
10 13 2 7 11 8 4 9 8 5 1 5 7 18 9 2 3 0 11 8 6
output
46

Note

Bitwise OR of two non-negative integers a and b is the number $c = a\ OR\ b$, such that each of its digits in binary notation is 1 if and only if at least one of a or b have 1 in the corresponding position in binary notation.

In the first sample, one of the optimal answers is $l = 2$ and $r = 4$, because $f(a, 2, 4) + f(b, 2, 4) = (2\ OR\ 4\ OR\ 3) + (3\ OR\ 3\ OR\ 12) = 7 + 15 = 22$. Other ways to get maximum value is to choose $l = 1$ and $r = 4$, $l = 1$ and $r = 5$, $l = 2$ and $r = 4$, $l = 2$ and $r = 5$, $l = 3$ and $r = 4$, or $l = 3$ and $r = 5$.

In the second sample, the maximum value is obtained for $l = 1$ and $r = 9$.

B. Print Check

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

Kris works in a large company "Blake Technologies". As a best engineer of the company he was assigned a task to develop a printer that will be able to print horizontal and vertical strips. First prototype is already built and Kris wants to tests it. He wants you to implement the program that checks the result of the printing.

Printer works with a rectangular sheet of paper of size $n \times m$. Consider the list as a table consisting of n rows and m columns. Rows are numbered from top to bottom with integers from 1 to n, while columns are numbered from left to right with integers from 1 to m. Initially, all cells are painted in color 0.

Your program has to support two operations:

1. Paint all cells in row r_i in color a_i;
2. Paint all cells in column c_i in color a_i.

If during some operation i there is a cell that have already been painted, the color of this cell also changes to a_i.

Your program has to print the resulting table after k operation.

Input

The first line of the input contains three integers n, m and k ($1 \le n, m \le 5000$, $n \cdot m \le 100\,000$, $1 \le k \le 100\,000$) — the dimensions of the sheet and the number of operations, respectively.

Each of the next k lines contains the description of exactly one query:

- $1\ r_i\ a_i$ ($1 \le r_i \le n$, $1 \le a_i \le 10^9$), means that row r_i is painted in color a_i;
- $2\ c_i\ a_i$ ($1 \le c_i \le m$, $1 \le a_i \le 10^9$), means that column c_i is painted in color a_i.

Output

Print n lines containing m integers each — the resulting table after all operations are applied.

Examples

input
3 3 3 1 1 3 2 2 1 1 2 2
output
3 1 3 2 2 2 0 1 0

input
5 3 5 1 1 1 1 3 1 1 5 1 2 1 1 2 3 1
output
1 1 1 1 0 1 1 1 1 1 0 1 1 1 1

Note

The figure below shows all three operations for the first sample step by step. The cells that were painted on the corresponding step are marked gray.

C. Report

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

Each month Blake gets the report containing main economic indicators of the company "Blake Technologies". There are n commodities produced by the company. For each of them there is exactly one integer in the final report, that denotes corresponding revenue. Before the report gets to Blake, it passes through the hands of m managers. Each of them may reorder the elements in some order. Namely, the i-th manager either sorts first r_i numbers in non-descending or non-ascending order and then passes the report to the manager $i + 1$, or directly to Blake (if this manager has number $i = m$).

Employees of the "Blake Technologies" are preparing the report right now. You know the initial sequence a_i of length n and the description of each manager, that is value r_i and his favourite order. You are asked to speed up the process and determine how the final report will look like.

Input

The first line of the input contains two integers n and m ($1 \le n, m \le 200\,000$) — the number of commodities in the report and the number of managers, respectively.

The second line contains n integers a_i ($|a_i| \le 10^9$) — the initial report before it gets to the first manager.

Then follow m lines with the descriptions of the operations managers are going to perform. The i-th of these lines contains two integers t_i and r_i ($t_i \in \{1, 2\}$, $1 \le r_i \le n$), meaning that the i-th manager sorts the first r_i numbers either in the non-descending (if $t_i = 1$) or non-ascending (if $t_i = 2$) order.

Output

Print n integers — the final report, which will be passed to Blake by manager number m.

Examples

input
3 1 1 2 3 2 2
output
2 1 3

input
4 2 1 2 4 3 2 3 1 2
output
2 4 1 3

Note

In the first sample, the initial report looked like: 1 2 3. After the first manager the first two numbers were transposed: **2 1** 3. The report got to Blake in this form.

In the second sample the original report was like this: 1 2 4 3. After the first manager the report changed to: **4 2 1** 3. After the second manager the report changed to: **2 4** 1 3. This report was handed over to Blake.

D. Messenger

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Each employee of the "Blake Techologies" company uses a special messaging app "Blake Messenger". All the stuff likes this app and uses it constantly. However, some important futures are missing. For example, many users want to be able to search through the message history. It was already announced that the new feature will appear in the nearest update, when developers faced some troubles that only you may help them to solve.

All the messages are represented as a strings consisting of only lowercase English letters. In order to reduce the network load strings are represented in the special compressed form. Compression algorithm works as follows: string is represented as a concatenation of n blocks, each block containing only equal characters. One block may be described as a pair (l_i, c_i), where l_i is the length of the i-th block and c_i is the corresponding letter. Thus, the string s may be written as the sequence of pairs $\langle (l_1, c_1), (l_2, c_2), \ldots, (l_n, c_n) \rangle$.

Your task is to write the program, that given two compressed string t and s finds all occurrences of s in t. Developers know that there may be many such occurrences, so they only ask you to find the **number** of them. Note that p is the starting position of some occurrence of s in t if and only if $t_p t_{p+1} \ldots t_{p+|s|-1} = s$, where t_i is the i-th character of string t.

Note that the way to represent the string in compressed form may not be unique. For example string "aaaa" may be given as $\langle (4, a) \rangle$, $\langle (3, a), (1, a) \rangle$, $\langle (2, a), (2, a) \rangle$...

Input

The first line of the input contains two integers n and m ($1 \le n, m \le 200\,000$) — the number of blocks in the strings t and s, respectively.

The second line contains the descriptions of n parts of string t in the format "l_i-c_i" ($1 \le l_i \le 1\,000\,000$) — the length of the i-th part and the corresponding lowercase English letter.

The second line contains the descriptions of m parts of string s in the format "l_i-c_i" ($1 \le l_i \le 1\,000\,000$) — the length of the i-th part and the corresponding lowercase English letter.

Output

Print a single integer — the number of occurrences of s in t.

Examples

input
5 3 3-a 2-b 4-c 3-a 2-c 2-a 2-b 1-c
output
1

input
6 1 3-a 6-b 7-a 4-c 8-e 2-a 3-a
output
6

input
5 5 1-h 1-e 1-l 1-l 1-o 1-w 1-o 1-r 1-l 1-d
output
0

Note

In the first sample, t = "aaabbccccaaacc", and string s = "aabbc". The only occurrence of string s in string t starts at position $p = 2$.

In the second sample, t = "aaabbbbbbaaaaaaacccceeeeeeeeaa", and s = "aaa". The occurrences of s in t start at positions $p = 1$, $p = 10$, $p = 11$, $p = 12$, $p = 13$ and $p = 14$.

E. Product Sum

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

Blake is the boss of Kris, however, this doesn't spoil their friendship. They often gather at the bar to talk about intriguing problems about maximising some values. This time the problem is really special.

You are given an array a of length n. The *characteristic* of this array is the value $c = \sum_{i=1}^{n} a_i \cdot i$ — the sum of the products of the values a_i by i. One may perform the following operation **exactly once**: pick some element of the array and move to any position. In particular, it's allowed to move the element to the beginning or to the end of the array. Also, it's allowed to put it back to the initial position. The goal is to get the array with the maximum possible value of characteristic.

Input

The first line of the input contains a single integer n ($2 \le n \le 200\,000$) — the size of the array a.

The second line contains n integers a_i ($1 \le i \le n$, $|a_i| \le 1\,000\,000$) — the elements of the array a.

Output

Print a single integer — the maximum possible value of characteristic of a that can be obtained by performing no more than one move.

Examples

input
4 4 3 2 5
output
39

input
5 1 1 2 7 1
output
49

input
3 1 1 2
output
9

Note

In the first sample, one may pick the first element and place it before the third (before 5). Thus, the answer will be $3 \cdot 1 + 2 \cdot 2 + 4 \cdot 3 + 5 \cdot 4 = 39$.

In the second sample, one may pick the fifth element of the array and place it before the third. The answer will be $1 \cdot 1 + 1 \cdot 2 + 1 \cdot 3 + 2 \cdot 4 + 7 \cdot 5 = 49$.