Problem A. Arrange and Count!

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 mebibytes

Alice has a sequence $a_1, a_2, \ldots, a_n$. She can rearrange the sequence using the following operation any number of times:

- Select an integer i $(1 \le i \le n)$ and change the sequence to $a_i, a_{i-1}, \ldots, a_1, a_n, a_{n-1}, \ldots, a_{i+1}$.

Alice would like to know the number of different sequences can be obtained modulo $(10^9 + 7)$.

Input

The input consists of several test cases terminated by end-of-file. For each test case:

The first line contains an integer n, the length of the sequence.

The second line contains n integers $a_1, a_2, \ldots, a_n$.

- $1 \le n \le 10^5$
- $1 \le a_i \le n$
- The sum of n does not exceed 2×10^6.

Output

For each test case, print an integer which denotes the result.

Example

standard input	standard output
4 1 1 1 1 4 1 1 2 2 4 1 2 1 2 4 2 1 2 1	1 4 2 2

Problem B. Build More 2020's!

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Byteazar got a string S $(s_1 \dots s_n)$ of length n consisting of only digits '0', '1', and '2', and he wants to pick some **disjoint** subsequences which equal to 2020, as many as possible.

Formally, Byteazar would like to find k quadruples $(a_1, b_1, c_1, d_1), \dots, (a_k, b_k, c_k, d_k)$ such as

- $1 \le a_i < b_i < c_i < d_i \le n$
- $s_{a_i} s_{b_i} s_{c_i} s_{d_i} = 2020$
- $\{a_i, b_i, c_i, d_i\} \cap \{a_j, b_j, c_j, d_j\} = \emptyset$ for $i \ne j$.

Find the maximum value of k.

Input

The input consists of several test cases terminated by end-of-file.

The first line of each test case contains an integer n $(1 \le n \le 10^5)$. Second line contains the string S $(s_1 \dots s_n)$. $(s_i \in \{0, 1, 2\})$. Sum of n in all test cases does not exceed 10^6.

Output

For each test case print an integer which denotes the result.

Examples

standard input	standard output
4 2222 7 2101210 9 122002200	0 1 2

Problem C. Choose Two Subsequences

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Clara has two strings s and t. She would like to choose two subsequences x from s and y from t such that:

- x is lexicographically smaller than or equal to y.
- The sum of $|x|$ and $|y|$ is maximal, where $|s|$ denotes the length of the string s.

Note that:

- Both x and y could be empty string.
- A subsequence is a sequence that can be derived from the given sequence by deleting zero or more elements without changing the order of the remaining elements.
- String x is lexicographically less than string y, if either x is a prefix of y (and $x \neq y$), or there exists such i $(1 \leq i \leq \min(|x|, |y|))$, that $x_i < y_i$, and for any j $(1 \leq j < i)$ $x_j = y_j$.

Input

The input consists of several test cases terminated by end-of-file. For each test case:

The first line contains a string s. The second line contains a string t.

- $1 \leq |s| \leq 2000$
- $1 \leq |t| \leq 2000$
- The sum of $|s|$ does not exceed 20000.
- The sum of $|t|$ does not exceed 20000.
- Both the strings consist only of English lowercase letters.

Output

For each test case, output the sum of $|x|$ and $|y|$.

Example

standard input	standard output
aaaa bbbb abcd abca abcd abcd	8 7 8

Problem D. Determinant Strikes Back

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Dinara has an integer x and two n arrays $a_1, \ldots, a_n$, $b_1, \ldots, b_n$. She makes an $n \times n$ matrix M where

$$M_{i,j} = \begin{cases} x + a_i b_j & \text{when } i = j \\ a_i b_j & \text{otherwise} \end{cases}$$

Find the determinant of the matrix M modulo $(10^9 + 7)$.

Input

The input consists of several test cases terminated by end-of-file.

The first line of each test case contains two integers n and x. The second line contains n integers $a_1, \ldots, a_n$. The third line contains n integers $b_1, \ldots, b_n$.

- $1 \le n \le 10^5$
- $0 \le x, a_i, b_i \le 10^9$
- The sum of n does not exceed 10^6.

Output

For each test case, print an integer which denotes the result.

Examples

standard input	standard output
2 1 0 0 0 0 2 1 1000000000 1000000000 1000000000 1000000000 3 2 2 3 3 2 3 3	1 99 96

Problem E. Efficient Data Structure

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 mebibytes

Elly has two sequences $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$. She would like to perform the following operations:

- `1` x y, change the value of a_x to y.
- `2` x y, change the value of b_x to y.
- `3` x, find the value of c_x, where $c_0 = 0$, $c_i = \max(c_{i-1} + b_i, a_i)$ for $1 \le i \le x$.

Implement an efficient data structure to process those operations.

Input

The input consists of several test cases terminated by end-of-file. For each test case:

The first line contains two integers n and m , which are the length of the two sequences and the number of operations. The second line contains n integers $a_1, a_2, \ldots, a_n$. The third line contains n integers $b_1, b_2, \ldots, b_n$. Each of the last m lines contains a query.

- $1 \le n, m \le 2 \times 10^5$
- $-10^9 \le a_i, b_i, y \le 10^9$
- $1 \le x \le n$
- The sum of n and the sum of m do not exceed 2×10^6.

Output

For each query of type 3, output an integer denoting the value of c_x.

Example

standard input	standard output
4 9 1 2 3 3 -1 2 3 3 3 1 3 2 3 3 3 4 2 2 -4 3 1 3 2 3 3 3 4	1 3 6 9 1 2 5 8

Problem F. Fibonnacci Suffix Array

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 mebibytes

The sequence of Fibonacci words is defined as follows: $fib_0 = b, fib_1 = a, fib_n = fib_{n-1}fib_{n-2}$ for $n \geq 2$. fib_n is the concatenation of fib_{n-1} and fib_{n-2}.

The first few Fibonacci words are: b, a, ab, aba, $abaab$, $abaababa$, $abaababaabaab$, ...

A suffix array for string s of length n is a permutation sa of integers from 1 to n such that $s[sa_1..n], s[sa_2..n], \ldots, s[sa_n..n]$ is the list of non-empty suffixes of s sorted in lexicographical order.

Let sa be the suffix array for fib_n. Your task is to calculate the value of $(sa_{p_1} \bmod m), (sa_{p_2} \bmod m), \ldots, (sa_{p_q} \bmod m)$.

Input

The input consists of several test cases terminated by end-of-file. For each test case:

The first line contains three integers n, q and m.

The second line contains q integers $p_1, p_2, \ldots, p_q$.

- $1 \leq n \leq 10^{18}$
- $1 \leq q \leq 10^5$
- $1 \leq m \leq 2 \times 10^9$
- $1 \leq p_i \leq \min(10^{18}, |fib_n|)$
- The sum of q does not exceed 10^6.

Output

For each test case, output q values $(sa_{p_1} \bmod m), (sa_{p_2} \bmod m), \ldots, (sa_{p_q} \bmod m)$, separated by spaces.

Examples

standard input	standard output
1 1 10 1 2 2 10 1 2 3 3 10 1 2 3 4 5 10 1 2 3 4 5 5 8 10 1 2 3 4 5 6 7 8	1 1 2 3 1 2 3 4 1 5 2 8 3 6 1 4 7 2 5

Problem G. Greatest Square

Input file:	*standard input*
Output file:	*standard output*
Time limit:	8 seconds
Memory limit:	512 mebibytes

Grete has a polygon consisting of n vertices. All sides of the polygon are parallel to the coordinate axes, and each two adjacent sides of the polygon are perpendicular. It is guaranteed that the polygon is simple, that is, it doesn't have self-intersections and self-touches.

Grete has m queries and in each query, a point (u_i, v_i) strictly inside the polygon is given. Grete would like to know the length of the side of the maximal square inside the polygon whose lower left corner is (u_i, v_i).

Input

The input consists of several test cases terminated by end-of-file. For each test case:

The first line contains two integers n and m, which are the number of vertices and the number of queries.

Each of the next n lines contains two integers x_i and y_i, the coordinates of vertices of the polygon in counterclockwise order.

Each of the next m lines contains two integers u_i and v_i, the coordinates of the lower left corner.

- $4 \le n \le 2 \times 10^5$
- $1 \le m \le 2 \times 10^5$
- $-10^8 \le x_i, y_i, u_i, v_i \le 10^8$
- The sum of n and the sum of m do not exceed 2×10^6.

Output

For each query, output an integer denoting the length of the maximal square inside the polygon.

Examples

standard input	standard output
4 3 0 0 4 0 4 4 0 4 1 1 2 2 3 3 12 12 3050 2000 2000 2000 2000 3635 -2000 3635 -2000 2000 -2590 2000 -2590 -2000 -2000 -2000 -2000 -3481 2000 -3481 2000 -2000 3050 -2000 1415 -2882 -1100 498 -827 -3331 -114 -542 -1887 3485 -1606 -1463 -768 880 -1261 1180 330 2648 -1017 -2886 -1213 -585 -2025 -1966	3 2 1 585 3100 2827 2542 150 3606 2755 2455 987 3017 3213 3966

Problem H. Hamming Distance

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

Helena has generated a list of sequences:

$$\begin{aligned} S^1 &= [1] \\ S^2 &= S^1 + [2] + S^1 \\ S^3 &= S^2 + [3] + S^2 \\ &\dots \\ S^m &= S^{m-1} + [m] + S^{m-1} \end{aligned}$$

where $A + B$ means the concatenation of two sequences A and B.

For a given sequence $[a_1, a_2, \dots, a_n]$, let $f(i)$ be the Hamming distance between $[a_1, a_2, \dots, a_n]$ and $[S^m_i, S^m_{i+1}, \dots, S^m_{i+n-1}]$ $(1 \le i \le |S^m| - n + 1)$.

Helena would like to find the minimum value of $f(i)$ and the sum of $f(i)$ modulo $(10^9 + 7)$.

Note that the Hamming distance between two sequences of equal length is the number of positions at which the corresponding elements are different.

Input

The input consists of several test cases terminated by end-of-file.

The first line contains two integers n and m.

The second line contains n integers $a_1, a_2, \dots, a_n$.

- $1 \le m \le 10^5$
- $1 \le n \le \min(|S^m|, 10^5)$
- $1 \le a_i \le m$
- The sum of n does not exceed 2×10^6.

Output

For each test case, output two integers denoting the minimum value of $f(i)$ and the sum of $f(i)$ modulo $(10^9 + 7)$.

Example

standard input	standard output
3 3 1 2 3 3 3 1 1 1 3 3 1 2 1	1 9 1 7 0 7

Problem I. Integers and Ranges

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

Isaac has a decimal integer $\overline{a_1 a_2 \dots a_n}$, possibly with leading zeroes. He knows that for m ranges $[l_1, r_1], [l_2, r_2], \dots, [l_m, r_m]$, it holds that $a_{l_i} \times a_{l_i+1} \times \cdots \times a_{r_i} \bmod 9 = 0$. Find the number of valid integers $\overline{a_1 a_2 \dots a_n}$, modulo $(10^9 + 7)$.

Input

The input consists of several test cases and is terminated by end-of-file.

The first line of each test case contains two integers n and m.

The ith of the following m lines contains two integers l_i and r_i.

- $1 \le n, m \le 10^3$
- $1 \le l_i \le r_i \le n$
- There are at most 100 test cases.

Output

For each test case, print an integer which denotes the result.

Example

standard input	standard output
2 1 1 2 4 2 1 3 2 4 50 1 1 50	40 4528 100268660

Problem J. Jailing

Input file:	*standard input*
Output file:	*standard output*
Time limit:	10 seconds
Memory limit:	512 mebibytes

Bobo has a matrix of size $n \times m$ filled with integers. It is guaranteed that all cells which contain the same value are 4-side connected.

Let's define a *jailing* J_x of a connected component with value x as minimum-area rectangle (with sides parallel to the matrix sides) that covers all cells of the component.

For each jailing B_x, Jessica would like to find the value of

$$s(B_x) = \sum_{B_y \in A \setminus \{x\}} f(B_x, B_y) \cdot y$$

where A is the set of all integers in the matrix and

$$f(B_x, B_y) = \begin{cases} 0 & \text{the area of intersection of } B_x \text{ and } B_y \text{ is } 0 \\ 0 & B_x \text{ is completely inside } B_y \text{ or vice versa} \\ 1 & \text{Otherwise} \end{cases}$$

Input

The input consists of several test cases terminated by end-of-file. For each test case:

The first line contains two integers n and m – the size of the matrix.

The second line contains $n \cdot m$ integers $a_{1,1}, a_{1,2}, \ldots, a_{1,m}, a_{2,1}, a_{2,2}, \ldots, a_{2,m}, \ldots, a_{n,1}, a_{n,2}, \ldots, a_{n,m}$, where $a_{i,j}$ is the value in the i-th row and the j-th column.

- $1 \le n \cdot m \le 10^6$
- $1 \le a_{i,j} \le nm$
- It is guaranteed that all cells which contain the same value are 4-side connected.
- It is guaranteed that the sum of $n \cdot m$ in all test cases does not exceed 10^7.

Output

For each test case, output an integer denoting the value of $\sum_{x \in A} s(x) \oplus x$, where $\oplus$ denotes the exclusive-or (XOR) operator.

Example

standard input	standard output
4 2 4 8 4 4 4 2 2 2 2 7 12 12 12 13 8 9 14 12 12 7 4 10 11 5 3 5 13 13 3 3 14 2 2 1 1 11 2 2 1 5 7	20 93 56