A. Sereja and Bottles

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

Sereja and his friends went to a picnic. The guys had n soda bottles just for it. Sereja forgot the bottle opener as usual, so the guys had to come up with another way to open bottles.

Sereja knows that the i-th bottle is from brand a_i, besides, you can use it to open **other** bottles of brand b_i. You can use one bottle to open multiple other bottles. Sereja can open bottle with opened bottle or closed bottle.

Knowing this, Sereja wants to find out the number of bottles they've got that they won't be able to open in any way. Help him and find this number.

Input

The first line contains integer n $(1 \le n \le 100)$ — the number of bottles. The next n lines contain the bottles' description. The i-th line contains two integers a_i, b_i $(1 \le a_i, b_i \le 1000)$ — the description of the i-th bottle.

Output

In a single line print a single integer — the answer to the problem.

Examples

input

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

output

```
4
```

input

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

output

```
0
```

B. Sereja and Array

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Sereja has got an array, consisting of n integers, $a_1, a_2, ..., a_n$. Sereja is an active boy, so he is now going to complete m operations. Each operation will have one of the three forms:

1. Make v_i-th array element equal to x_i. In other words, perform the assignment $a_{v_i} = x_i$.
2. Increase each array element by y_i. In other words, perform n assignments $a_i = a_i + y_i$ $(1 \le i \le n)$.
3. Take a piece of paper and write out the q_i-th array element. That is, the element a_{q_i}.

Help Sereja, complete all his operations.

Input

The first line contains integers n, m $(1 \le n, m \le 10^5)$. The second line contains n space-separated integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^9)$ — the original array.

Next m lines describe operations, the i-th line describes the i-th operation. The first number in the i-th line is integer t_i $(1 \le t_i \le 3)$ that represents the operation type. If $t_i = 1$, then it is followed by two integers v_i and x_i, $(1 \le v_i \le n, 1 \le x_i \le 10^9)$. If $t_i = 2$, then it is followed by integer y_i $(1 \le y_i \le 10^4)$. And if $t_i = 3$, then it is followed by integer q_i $(1 \le q_i \le n)$.

Output

For each third type operation print value a_{q_i}. Print the values in the order, in which the corresponding queries follow in the input.

Examples

input

```
10 11
1 2 3 4 5 6 7 8 9 10
3 2
3 9
2 10
3 1
3 10
1 1 10
2 10
2 10
3 1
3 10
3 9
```

output

```
2
9
11
20
30
40
39
```

C. Sereja and Contest

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

During the last Sereja's Codesecrof round the server crashed many times, so the round was decided to be made unrated for some participants.

Let's assume that n people took part in the contest. Let's assume that the participant who got the first place has rating a_1, the second place participant has rating a_2, ..., the n-th place participant has rating a_n. Then changing the rating on the Codesecrof site is calculated by the formula $d_i = \sum_{j=1}^{i-1}(a_j \cdot (j-1) - (n-i) \cdot a_i)$.

After the round was over, the Codesecrof management published the participants' results table. They decided that if for a participant $d_i < k$, then the round can be considered unrated for him. But imagine the management's surprise when they found out that the participants' rating table is dynamic. In other words, when some participant is removed from the rating, he is removed from the results' table and the rating is recalculated according to the new table. And of course, all applications for exclusion from the rating are considered in view of the current table.

We know that among all the applications for exclusion from the rating the first application to consider is from the participant with the best rank (the rank with the minimum number), for who $d_i < k$. We also know that the applications for exclusion from rating were submitted by all participants.

Now Sereja wonders, what is the number of participants to be excluded from the contest rating, and the numbers of the participants in the original table in the order of their exclusion from the rating. Pay attention to the analysis of the first test case for a better understanding of the statement.

Input

The first line contains two integers n, k $(1 \le n \le 2 \cdot 10^5, -10^9 \le k \le 0)$. The second line contains n space-separated integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^9)$ — ratings of the participants in the initial table.

Output

Print the numbers of participants in the order in which they were removed from the table. Print the initial numbers of the participants, that is, the numbers that the participants had in the initial table.

Examples

input

```
5 0
5 3 4 1 2
```

output

```
2
3
4
```

input

```
10 -10
5 5 1 7 5 1 2 4 9 2
```

output

```
2
4
5
7
8
9
```

Note

Consider the first test sample.

1. Initially the sequence of the contest participants' ratings equals [5, 3, 4, 1, 2]. You can use this sequence to calculate the sequence of rating changes: [0, -9, -13, 8, 14]. According to the problem statement, the application of the participant who won the second place will be considered first.
2. As soon as the second place winner is out from the ratings, the participants' rating sequence will equal [5, 4, 1, 2]. By this sequence you can count the new sequence of rating changes: [0, -8, 2, 6]. According to the problem statement, the application of the participant who won the second place will be considered. Initially this participant won third place.
3. The new rating sequence equals [5, 1, 2], the new sequence of rating changes equals [0, -1, 1]. The second place participant's application is taken into consideration, initially this participant won the fourth place.
4. The new rating sequence equals [5, 2], the new sequence of rating changes equals [0, 0]. No more applications will be considered.

Thus, you should print 2, 3, 4.

D. Sereja and Periods

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

Let's introduce the designation $[x, n] = x + x + \cdots + x = \sum_{i=1}^{n} x$, where x is a string, n is a positive integer and operation " $+$ " is the string concatenation operation. For example, $[abc, 2] = abcabc$.

We'll say that string s *can be obtained* from string t, if we can remove some characters from string t and obtain string s. For example, strings ab and $acba$ can be obtained from string $xacbac$, and strings bx and aaa cannot be obtained from it.

Sereja has two strings, $w = [a, b]$ and $q = [c, d]$. He wants to find such maximum integer p $(p > 0)$, that $[q, p]$ can be obtained from string w.

Input

The first line contains two integers b, d $(1 \le b, d \le 10^7)$. The second line contains string a. The third line contains string c. The given strings are not empty and consist of lowercase English letters. Their lengths do not exceed 100.

Output

In a single line print an integer — the largest number p. If the required value of p doesn't exist, print 0.

Examples

input

```
10 3
abab
bab
```

output

```
3
```

E. Sereja and Subsequences

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Sereja has a sequence that consists of n positive integers, $a_1, a_2, ..., a_n$.

First Sereja took a piece of squared paper and wrote all **distinct** non-empty non-decreasing subsequences of sequence a. Then for each sequence written on the squared paper, Sereja wrote on a piece of lines paper all sequences that *do not exceed* it.

A sequence of positive integers $x = x_1, x_2, ..., x_r$ doesn't exceed a sequence of positive integers $y = y_1, y_2, ..., y_r$, if the following inequation holds: $x_1 \le y_1, x_2 \le y_2, ..., x_r \le y_r$.

Now Sereja wonders, how many sequences are written on the lines piece of paper. Help Sereja, find the required quantity modulo 1000000007 $(10^9 + 7)$.

Input

The first line contains integer n $(1 \le n \le 10^5)$. The second line contains n integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^6)$.

Output

In the single line print the answer to the problem modulo 1000000007 $(10^9 + 7)$.

Examples

input

```
1
42
```

output

```
42
```

input

```
3
1 2 2
```

output

```
13
```

input

```
5
1 2 3 4 5
```

output

```
719
```