CODEFORCES β
Sponsored by Telegram

15-295 Spring 2017 #6

A. Sereja and Suffixes

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

Sereja has an array a, consisting of n integers $a_1, a_2, ..., a_n$. The boy cannot sit and do nothing, he decided to study an array. Sereja took a piece of paper and wrote out m integers $l_1, l_2, ..., l_m$ $(1 \le l_i \le n)$. For each number l_i he wants to know how many distinct numbers are staying on the positions $l_i, l_i + 1, ..., n$. Formally, he want to find the number of distinct numbers among $a_{l_i}, a_{l_i + 1}, ..., a_n$.?

Sereja wrote out the necessary array elements but the array was so large and the boy was so pressed for time. Help him, find the answer for the described question for each l_i.

Input

The first line contains two integers n and m $(1 \le n, m \le 10^5)$. The second line contains n integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^5)$ — the array elements.

Next m lines contain integers $l_1, l_2, ..., l_m$. The i-th line contains integer l_i $(1 \le l_i \le n)$.

Output

Print m lines — on the i-th line print the answer to the number l_i.

Examples

input

```
10 10
1 2 3 4 1 2 3 4 100000 99999
1
2
3
4
5
6
7
8
9
10
```

output

```
6
6
6
6
6
5
4
3
2
1
```

B. Vacations

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

Vasya has n days of vacations! So he decided to improve his IT skills and do sport. Vasya knows the following information about each of this n days: whether that gym opened and whether a contest was carried out in the Internet on that day. For the i-th day there are four options:

1. on this day the gym is closed and the contest is not carried out;
2. on this day the gym is closed and the contest is carried out;
3. on this day the gym is open and the contest is not carried out;
4. on this day the gym is open and the contest is carried out.

On each of days Vasya can either have a rest or write the contest (if it is carried out on this day), or do sport (if the gym is open on this day).

Find the minimum number of days on which Vasya will have a rest (it means, he will not do sport and write the contest at the same time). The only limitation that Vasya has — *he does not want to do the same activity on two consecutive days: it means, he will not do sport on two consecutive days, and write the contest on two consecutive days*.

Input

The first line contains a positive integer n $(1 \le n \le 100)$ — the number of days of Vasya's vacations.

The second line contains the sequence of integers $a_1, a_2, ..., a_n$ $(0 \le a_i \le 3)$ separated by space, where:

- a_i equals 0, if on the i-th day of vacations the gym is closed and the contest is not carried out;
- a_i equals 1, if on the i-th day of vacations the gym is closed, but the contest is carried out;
- a_i equals 2, if on the i-th day of vacations the gym is open and the contest is not carried out;
- a_i equals 3, if on the i-th day of vacations the gym is open and the contest is carried out.

Output

Print the minimum possible number of days on which Vasya will have a rest. Remember that Vasya refuses:

- to do sport on any two consecutive days,
- to write the contest on any two consecutive days.

Examples

input

```
4
1 3 2 0
```

output

```
2
```

input

```
7
1 3 3 2 1 2 3
```

output

```
0
```

input

```
2
2 2
```

output

```
1
```

Note

In the first test Vasya can write the contest on the day number 1 and do sport on the day number 3. Thus, he will have a rest for only 2 days.

In the second test Vasya should write contests on days number 1, 3, 5 and 7, in other days do sport. Thus, he will not have a rest for a single day.

In the third test Vasya can do sport either on a day number 1 or number 2. He can not do sport in two days, because it will be contrary to the his limitation. Thus, he will have a rest for only one day.

C. Journey

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Recently Irina arrived to one of the most famous cities of Berland — the Berlatov city. There are n showplaces in the city, numbered from 1 to n, and some of them are connected by one-directional roads. The roads in Berlatov are designed in a way such that there **are no** cyclic routes between showplaces.

Initially Irina stands at the showplace 1, and the endpoint of her journey is the showplace n. Naturally, Irina wants to visit as much showplaces as she can during her journey. However, Irina's stay in Berlatov is limited and she can't be there for more than T time units.

Help Irina determine how many showplaces she may visit during her journey from showplace 1 to showplace n within a time not exceeding T. It is guaranteed that there is at least one route from showplace 1 to showplace n such that Irina will spend no more than T time units passing it.

Input

The first line of the input contains three integers n, m and T ($2 \le n \le 5000,\ 1 \le m \le 5000,\ 1 \le T \le 10^9$) — the number of showplaces, the number of roads between them and the time of Irina's stay in Berlatov respectively.

The next m lines describes roads in Berlatov. i-th of them contains 3 integers u_i, v_i, t_i ($1 \le u_i, v_i \le n, u_i \ne v_i, 1 \le t_i \le 10^9$), meaning that there is a road starting from showplace u_i and leading to showplace v_i, and Irina spends t_i time units to pass it. It is guaranteed that the roads do not form cyclic routes.

It is guaranteed, that there is at most one road between each pair of showplaces.

Output

Print the single integer k ($2 \le k \le n$) — the maximum number of showplaces that Irina can visit during her journey from showplace 1 to showplace n within time not exceeding T, in the first line.

Print k distinct integers in the second line — indices of showplaces that Irina will visit on her route, in the order of encountering them.

If there are multiple answers, print any of them.

Examples

input

```
4 3 13
1 2 5
2 3 7
2 4 8
```

output

```
3
1 2 4
```

input

```
6 6 7
1 2 2
1 3 3
3 6 3
2 4 2
4 6 2
6 5 1
```

output

```
4
1 2 4 6
```

input

```
5 5 6
1 3 3
3 5 3
1 2 2
2 4 3
4 5 2
```

output

```
3
1 3 5
```

D. Almost Arithmetical Progression

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

Gena loves sequences of numbers. Recently, he has discovered a new type of sequences which he called an almost arithmetical progression. A sequence is an *almost arithmetical progression*, if its elements can be represented as:

- $a_1 = p$, where p is some integer;
- $a_i = a_{i-1} + (-1)^{i+1} \cdot q$ $(i > 1)$, where q is some integer.

Right now Gena has a piece of paper with sequence b, consisting of n integers. Help Gena, find there the longest subsequence of integers that is an almost arithmetical progression.

Sequence $s_1, s_2, \ldots, s_k$ is a subsequence of sequence $b_1, b_2, \ldots, b_n$, if there is such increasing sequence of indexes $i_1, i_2, \ldots, i_k$ $(1 \le i_1 < i_2 < \ldots < i_k \le n)$, that $b_{i_j} = s_j$. In other words, sequence s can be obtained from b by crossing out some elements.

Input

The first line contains integer n $(1 \le n \le 4000)$. The next line contains n integers $b_1, b_2, \ldots, b_n$ $(1 \le b_i \le 10^6)$.

Output

Print a single integer — the length of the required longest subsequence.

Examples

input

```
2
3 5
```

output

```
2
```

input

```
4
10 20 10 30
```

output

```
3
```

Note

In the first test the sequence actually is the suitable subsequence.

In the second test the following subsequence fits: $10, 20, 10$.

E. Choosing Balls

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are n balls. They are arranged in a row. Each ball has a color (for convenience an integer) and an integer value. The color of the i-th ball is c_i and the value of the i-th ball is v_i.

Squirrel Liss chooses some balls and makes a new sequence without changing the relative order of the balls. She wants to maximize the value of this sequence.

The value of the sequence is defined as the sum of following values for each ball (where a and b are given constants):

- If the ball is not in the beginning of the sequence and the color of the ball is same as previous ball's color, add (the value of the ball) $\times$ a.
- Otherwise, add (the value of the ball) $\times$ b.

You are given q queries. Each query contains two integers a_i and b_i. For each query find the maximal value of the sequence she can make when $a = a_i$ and $b = b_i$.

Note that the new sequence can be **empty**, and the value of an empty sequence is defined as zero.

Input

The first line contains two integers n and q ($1 \le n \le 10^5$; $1 \le q \le 500$). The second line contains n integers: $v_1, v_2, ..., v_n$ ($|v_i| \le 10^5$). The third line contains n integers: $c_1, c_2, ..., c_n$ ($1 \le c_i \le n$).

The following q lines contain the values of the constants a and b for queries. The i-th of these lines contains two integers a_i and b_i ($|a_i|, |b_i| \le 10^5$).

In each line integers are separated by single spaces.

Output

For each query, output a line containing an integer — the answer to the query. The i-th line contains the answer to the i-th query in the input order.

Please, do not write the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Examples

input

```
6 3
1 -2 3 4 0 -1
1 2 1 2 1 1
5 1
-2 1
1 0
```

output

```
20
9
4
```

input

```
4 1
-3 6 -1 2
1 2 3 1
1 -1
```

output

```
5
```

Note

In the first example, to achieve the maximal value:

- In the first query, you should select 1st, 3rd, and 4th ball.
- In the second query, you should select 3rd, 4th, 5th and 6th ball.
- In the third query, you should select 2nd and 4th ball.

Note that there may be other ways to achieve the maximal value.

F. Bulbo

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

Bananistan is a beautiful banana republic. Beautiful women in beautiful dresses. Beautiful statues of beautiful warlords. Beautiful stars in beautiful nights.

In Bananistan people play this crazy game – Bulbo. There's an array of bulbs and player at the position, which represents one of the bulbs. The distance between two neighboring bulbs is 1. Before each turn player can change his position with cost $|pos_{new} - pos_{old}|$. After that, a contiguous set of bulbs lights-up and player pays the cost that's equal to the distance to the closest shining bulb. Then, all bulbs go dark again. The goal is to minimize your summed cost. I tell you, Bananistanians are spending their nights playing with bulbs.

Banana day is approaching, and you are hired to play the most beautiful Bulbo game ever. A huge array of bulbs is installed, and you know your initial position and all the light-ups in advance. You need to play the ideal game and impress Bananistanians, and their families.

Input

The first line contains number of turns n and initial position x. Next n lines contain two numbers l_{start} and l_{end}, which represent that all bulbs from interval $[l_{start}, l_{end}]$ are shining this turn.

- $1 \le n \le 5000$
- $1 \le x \le 10^9$
- $1 \le l_{start} \le l_{end} \le 10^9$

Output

Output should contain a single number which represents the best result (minimum cost) that could be obtained by playing this Bulbo game.

Examples

input

```
5 4
2 7
9 16
8 10
9 17
1 6
```

output

```
8
```

Note

Before 1. turn move to position 5

Before 2. turn move to position 9

Before 5. turn move to position 8