

Information

Memory limit

The limit is 512 MiB for each problem.

Source code limit

The size of each solution source code can't exceed 256 KiB.

Submissions limit

You can submit at most 50 solutions for each problem.

You can submit a solution to each task at most once per 30 seconds. This restriction does not apply in the last 15 minutes of the contest round.

Scoring

Each problem consists of several subtasks. The subtask score is awarded if all tests in the subtask are passed.

The number of points scored for the problem is the total number of points scored on each of its subtasks. The score for the subtask is the maximum number of points earned for this subtask among all the solutions submitted.

Feedback

To get feedback for your solution, go to "Runs" tab in PCMS2 Web Client and use "View Feedback" link. In each problem of the contest you will see the score for each subtask, or the verdict for the first failed test.

Scoreboard

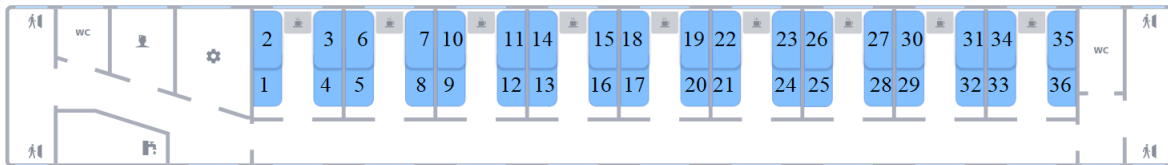
The contestants' scoreboard is available during the contest. Use "Monitor" link in PCMS2 Web Client to access the scoreboard. The standings provided in PCMS2 Web Client are not final.

Problem A. A seat on the train

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

Azat decided to buy a train ticket with seat number n . He knows that there are k cabins in the train car and there are 4 seats in each cabin. Numbering of sleeping places in the cabin:

1. bottom left
2. top left
3. top right
4. bottom right



Help him to understand in which train car and in which cabin Azat will go, and also determine — this is the upper or lower seat.

Input

The first line contains a single integer n ($1 \leq n \leq 10^{18}$) — the seat number on the train.

The second line contains one integer k ($1 \leq k \leq 10^{17}$) — the number of cabins in the train car.

Note that the input data and the answer can be quite large, so you should use a 64-bit data type, for example `long long` in C/C++, `long` in Java, `int64` in Pascal.

Output

In the first line print the number of the train car and cabin where Azat will travel.

In the second line print «Upper» if Azat goes to the top place, otherwise print «Lower».

Scoring

Points for each subtask are awarded only if all tests for this subtask and the required subtasks are successfully passed.

Subtask	Score	Constraints	Required subtasks
0	0	Sample tests	
1	30	$1 \leq n, k \leq 10^6$	
2	30	$1 \leq k \leq 10^6$, $1 \leq n \leq 10^{18}$	1
3	40	No additional constraints	1, 2

Examples

standard input	standard output
34 6	2 3 Upper
36 9	1 9 Lower
26 9	1 7 Upper

Problem B. Complex strings

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

Azat on the train bought a string s from the characters a , b and c . He became interested in finding its longest substring of one of the types:

- ABA
- ACA
- BAB
- BCB
- CAC
- CBC

Character **A** — non-empty sequence of a , **B** — non-empty sequence of b , **C** — non-empty sequence of c .

Input

The first input line contains the string s ($s_i \in \{a, b, c\}$, $|s| \leq 2 \cdot 10^5$).

Output

Print a single integer — the length of the maximum substring satisfying the condition of the problem.

Scoring

Points for each subtask are awarded only if all tests for this subtasks and required subtasks have been successfully completed.

Subtask	Score	Constraints	Required subtasks
0	0	Sample tests	
1	30	$ s \leq 10^2$	
2	30	$ s \leq 10^3$	1
3	40	No additional constraints	1, 2

Examples

standard input	standard output
abcacca	4
abbbaacca	7
abbbaaabb	9

Problem C. Seating Arrangement

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

Boarding is going to start. An aircraft has n rows and m seats in each row. The seat number 1 is located by the window and the seat number m is located by the aisle.

You are given the order of passengers entering the aircraft and the seat number of each passenger. If a passenger wants to take his seat number x , but in any place from the aisle to his seat another passenger has already been seated (let it be seat number y), then the passenger at y should stand up and let other passenger pass.

There are k passenger, and they get in the aircraft one by one starting from 1. Your task is to calculate how many times did passengers stand up in total.

Input

The first line contains three integers n , m and k ($1 \leq n \leq 10^9$, $1 \leq m \leq 10^9$, $1 \leq k \leq 10^5$) — the number of rows, the number of seats in each row and the number of passengers, respectively. It is guaranteed that the number of passengers is not greater than $n \cdot m$.

Each of the next k lines contains two integers x and y ($1 \leq x \leq n$, $1 \leq y \leq m$) — the row number and the seat number, where current passenger should sit down.

It is guaranteed that each passenger has his own unique seat.

Output

Print a single integer — how many times did passengers stand up in total.

Scoring

Subtask	Score	Constraints	Required subtasks
0	0	Sample tests	—
1	15	$1 \leq n \leq 100$ $m = 3$	—
2	15	$1 \leq n \leq 100$ $1 \leq m \leq 100$	1
3	20	$1 \leq n \leq 10^9$ $1 \leq m \leq 100$	1, 2
4	25	$1 \leq n \leq 10^9$ $1 \leq m \leq 10^5$	1, 2, 3
5	25	No additional constraints	1, 2, 3, 4

Examples

standard input	standard output
1 2 2 1 2 1 1	1
1 2 2 1 1 1 2	0
2 2 2 1 2 2 1	0
2 3 5 1 3 2 2 2 3 1 1 2 1	3
3 3 9 1 3 1 2 1 1 2 2 2 1 2 3 3 1 3 3 3 2	5

Problem D. Max cover

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

You have n integers a_1, \dots, a_n arranged in a circle. You choose a single integer s from one to n . After that circle is distributed on a few non-intersecting parts: the first one includes elements $a_s, \dots, a_{(s+k-1)\%n+1}$. The second one includes the next k elements: $a_{(s+k)\%n+1}, \dots, a_{(s+2k-1)\%n+1}$ and so on while the whole circle is not covered. Pay attention what length of the last part can be less than k .

Lets find the maximum number in each part, after that find the minimum of those and call found number as result.

What is the maximum value of result if you can choose s ?

Input

There are two natural numbers in the first line: n ($1 \leq n \leq 2 \cdot 10^5$) and k ($1 \leq k \leq n$).

There are n integer numbers in the second line: $a_1 \dots, a_n$ ($-10^9 \leq a_i \leq 10^9$).

Output

Print a single number — the maximum result you can get after the optimal s choosen.

Scoring

Score for every subtask will be credited only after all tests from this subtask and the dependencies are succeeded.

Subtask	Score	Constraints	Required subtasks
0	0	Sample tests	
1	10	$1 \leq n \leq 500$	
2	20	$1 \leq n \leq 2000$	1
3	30	$1 \leq n \leq 2 \cdot 10^5,$ $n\%k = 0$	—
4	40	No additional constraints	1, 2, 3

Examples

standard input	standard output
3 2 1 2 3	2
6 2 -1 -1 1 1 -1 1	1

Problem E. Building Houses

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

Azat started his own construction company. The company has n bricks storages, there are a_i bricks in the storage number i . The company has got t orders for building houses, the order number j requires q_j bricks to build a house.

Azat selects some orders and calculates the total number of bricks required to process the orders. He also selects some bricks storages and uses them to process the orders. Azat selects brick storages in such a way that the total number of bricks in selected storages is equal to the total number of bricks required to process selected orders. In other words, no one brick should remain unused after selecting orders and bricks storages.

Your task is to determine the maximum number of orders Azat can process.

Input

The first line contains a single integer n ($1 \leq n \leq 100$) — the number of bricks storages.

The second line contains n integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^4$) — the number of bricks in each storage. It is guaranteed that the total number of bricks (the sum of all a_i) does not exceed 10^4 .

The third line contains a single integer t ($1 \leq t \leq 100$) — the number of orders.

The fourth line contains t integers q_1, q_2, \dots, q_t ($1 \leq q_i \leq 10^4$) — the number of bricks required to build each house. It is guaranteed that the total number of bricks required to build houses (the sum of all q_i) does not exceed 10^4 .

Output

Output a single integer — the maximum number of orders Azat can process.

Scoring

Points for each subtask are awarded only if all tests for this subtask and the required subtasks are successfully passed.

Subtask	Score	Constraints	Required subtasks
0	0	Sample tests	—
1	10	Array a_i has 2 distinct elements $n \leq 20$ $t = 1$	—
2	15	$n \cdot t \leq 20$	1
3	15	$n \leq 20$ $t \leq 20$	1, 2
4	15	$t = 1$	—
5	20	$n = 1$	—
6	25	No additional constraints	1, 2, 3, 4, 5

Examples

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