Codeforces Round #121 (Div. 2)

A. Funky Numbers

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

As you very well know, this year's funkiest numbers are so called triangular numbers (that is, integers that are representable as $\frac{k(k+1)}{2}$, where k is some positive integer), and the coolest numbers are those that are representable as a sum of two triangular numbers.

A well-known hipster Andrew adores everything funky and cool but unfortunately, he isn't good at maths. Given number n, help him define whether this number can be represented by a sum of two triangular numbers (not necessarily different)!

Input
The first input line contains an integer n $(1 \leq n \leq 10^9)$.

Output
Print "YES" (without the quotes), if n can be represented as a sum of two triangular numbers, otherwise print "NO" (without the quotes).

Sample test(s)

input
256
output
YES

input
512
output
NO

Note
In the first sample number $256 = \frac{2 \cdot 3}{2} + \frac{22 \cdot 23}{2}$.

In the second sample number 512 can not be represented as a sum of two triangular numbers.

B. Walking in the Rain

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

In Berland the opposition is going to arrange mass walking on the boulevard. The boulevard consists of n tiles that are lain in a row and are numbered from 1 to n from right to left. The opposition should start walking on the tile number 1 and the finish on the tile number n. During the walk it is allowed to move from right to left between adjacent tiles in a row, and jump over a tile. More formally, if you are standing on the tile number i ($i < n$ - 1), you can reach the tiles number $i + 1$ or the tile number $i + 2$ from it (if you stand on the tile number n - 1, you can only reach tile number n). We can assume that all the opposition movements occur instantaneously.

In order to thwart an opposition rally, the Berland bloody regime organized the rain. The tiles on the boulevard are of poor quality and they are rapidly destroyed in the rain. We know that the i-th tile is destroyed after a_i days of rain (on day a_i tile isn't destroyed yet, and on day $a_i + 1$ it is already destroyed). Of course, no one is allowed to walk on the destroyed tiles! So the walk of the opposition is considered thwarted, if either the tile number 1 is broken, or the tile number n is broken, or it is impossible to reach the tile number n from the tile number 1 if we can walk on undestroyed tiles.

The opposition wants to gather more supporters for their walk. Therefore, the more time they have to pack, the better. Help the opposition to calculate how much time they still have and tell us for how many days the walk from the tile number 1 to the tile number n will be possible.

Input

The first line contains integer n ($1 \le n \le 10^3$) — the boulevard's length in tiles.

The second line contains n space-separated integers a_i — the number of days after which the i-th tile gets destroyed ($1 \le a_i \le 10^3$).

Output

Print a single number — the sought number of days.

Sample test(s)

input
4 10 3 5 10
output
5

input
5 10 2 8 3 5
output
5

Note

In the first sample the second tile gets destroyed after day three, and the only path left is $1 \to 3 \to 4$. After day five there is a two-tile gap between the first and the last tile, you can't jump over it.

In the second sample path $1 \to 3 \to 5$ is available up to day five, inclusive. On day six the last tile is destroyed and the walk is thwarted.

C. Dynasty Puzzles

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

The ancient Berlanders believed that the longer the name, the more important its bearer is. Thus, Berland kings were famous for their long names. But long names are somewhat inconvenient, so the Berlanders started to abbreviate the names of their kings. They called every king by the first letters of its name. Thus, the king, whose name was Victorious Vasily Pupkin, was always called by the berlanders VVP.

In Berland over its long history many dynasties of kings replaced each other, but they were all united by common traditions. Thus, according to one Berland traditions, to maintain stability in the country, the first name of the heir should be the same as the last name his predecessor (hence, the first letter of the abbreviated name of the heir coincides with the last letter of the abbreviated name of the predecessor). Berlanders appreciate stability, so this tradition has never been broken. Also Berlanders like perfection, so another tradition requires that the first name of the first king in the dynasty coincides with the last name of the last king in this dynasty (hence, the first letter of the abbreviated name of the first king coincides with the last letter of the abbreviated name of the last king). This tradition, of course, has also been always observed.

The name of a dynasty is formed by very simple rules: we take all the short names of the kings in the order in which they ruled, and write them in one line. Thus, a dynasty of kings "ab" and "ba" is called "abba", and the dynasty, which had only the king "abca", is called "abca".

Vasya, a historian, has recently found a list of abbreviated names of all Berland kings and their relatives. Help Vasya to find the maximally long name of the dynasty that could have existed in Berland.

Note that in his list all the names are ordered by the time, that is, if name A is earlier in the list than B, then if A and B were kings, then king A ruled before king B.

Input

The first line contains integer n ($1 \le n \le 5 \cdot 10^5$) — the number of names in Vasya's list. Next n lines contain n abbreviated names, one per line. An abbreviated name is a non-empty sequence of lowercase Latin letters. Its length does not exceed 10 characters.

Output

Print a single number — length of the sought dynasty's name in letters.

If Vasya's list is wrong and no dynasty can be found there, print a single number 0.

Sample test(s)

input
3 abc ca cba
output
6

input
4 vvp vvp dam vvp
output
0

input
3 ab c def
output
1

Note

In the first sample two dynasties can exist: the one called "abcca" (with the first and second kings) and the one called "abccba" (with the first and third kings).

In the second sample there aren't acceptable dynasties.

The only dynasty in the third sample consists of one king, his name is "c".

D. Demonstration

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

In the capital city of Berland, Bertown, demonstrations are against the recent election of the King of Berland. Berland opposition, led by Mr. Ovalny, believes that the elections were not fair enough and wants to organize a demonstration at one of the squares.

Bertown has n squares, numbered from 1 to n, they are numbered in the order of increasing distance between them and the city center. That is, square number 1 is central, and square number n is the farthest from the center. Naturally, the opposition wants to hold a meeting as close to the city center as possible (that is, they want an square with the minimum number).

There are exactly k $(k < n)$ days left before the demonstration. Now all squares are free. But the Bertown city administration never sleeps, and the approval of an application for the demonstration threatens to become a very complex process. The process of approval lasts several days, but every day the following procedure takes place:

- The opposition shall apply to hold a demonstration at a free square (the one which isn't used by the administration).
- The administration tries to move the demonstration to the worst free square left. To do this, the administration organizes some long-term activities on the square, which is specified in the application of opposition. In other words, the administration starts using the square and it is no longer free. Then the administration proposes to move the opposition demonstration to the worst free square. If the opposition has applied for the worst free square **then request is accepted and administration doesn't spend money**. If the administration does not have enough money to organize an event on the square in question, the opposition's application is accepted. If administration doesn't have enough money to organize activity, then rest of administration's money spends and application is accepted
- If the application is not accepted, then the opposition can agree to the administration's proposal (that is, take the worst free square), or withdraw the current application and submit another one the next day. If there are no more days left before the meeting, the opposition has no choice but to agree to the proposal of City Hall. If application is accepted opposition can reject it. It means than opposition still can submit more applications later, **but square remains free**.

In order to organize an event on the square i, the administration needs to spend a_i bourles. Because of the crisis the administration has only b bourles to confront the opposition. What is the best square that the opposition can take, if the administration will keep trying to occupy the square in question each time? Note that the administration's actions always depend only on the actions of the opposition.

Input

The first line contains two integers n and k — the number of squares and days left before the meeting, correspondingly ($1 \le k < n \le 10^5$).

The second line contains a single integer b — the number of bourles the administration has ($1 \le b \le 10^{18}$).

The third line contains n space-separated integers a_i — the sum of money, needed to organise an event on square i ($1 \le a_i \le 10^9$).

Please, do not use 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.

Output

Print a single number — the minimum number of the square where the opposition can organize the demonstration.

Sample test(s)

input
5 2 8 2 4 5 3 1
output
2

input
5 2 8 3 2 4 1 5
output
5

input
5 4 1000000000000000 5 4 3 2 1
output
5

Note

In the first sample the opposition can act like this. On day one it applies for square 3. The administration has to organize an event there and end up with 3 bourles. If on the second day the opposition applies for square 2, the administration won't have the money to intervene.

In the second sample the opposition has only the chance for the last square. If its first move occupies one of the first four squares, the administration is left with at least 4 bourles, which means that next day it can use its next move to move the opposition from any square to the last one.

In the third sample administration has a lot of money, so opposition can occupy only last square.

E. Fools and Roads

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

They say that Berland has exactly two problems, fools and roads. Besides, Berland has n cities, populated by the fools and connected by the roads. All Berland roads are bidirectional. As there are many fools in Berland, between each pair of cities there is a path (or else the fools would get upset). Also, between each pair of cities there is no more than one simple path (or else the fools would get lost).

But that is not the end of Berland's special features. In this country fools sometimes visit each other and thus spoil the roads. The fools aren't very smart, so they always use only the simple paths.

A simple path is the path which goes through every Berland city not more than once.

The Berland government knows the paths which the fools use. Help the government count for each road, how many distinct fools can go on it.

Note how the fools' paths are given in the input.

Input

The first line contains a single integer n ($2 \le n \le 10^5$) — the number of cities.

Each of the next n - 1 lines contains two space-separated integers u_i, v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$), that means that there is a road connecting cities u_i and v_i.

The next line contains integer k ($0 \le k \le 10^5$) — the number of pairs of fools who visit each other.

Next k lines contain two space-separated numbers. The i-th line ($i > 0$) contains numbers a_i, b_i ($1 \le a_i, b_i \le n$). That means that the fool number $2i$ - 1 lives in city a_i and visits the fool number $2i$, who lives in city b_i. The given pairs describe simple paths, because between every pair of cities there is only one simple path.

Output

Print n - 1 integer. The integers should be separated by spaces. The i-th number should equal the number of fools who can go on the i-th road. The roads are numbered starting from one in the order, in which they occur in the input.

Sample test(s)

input
5 1 2 1 3 2 4 2 5 2 1 4 3 5
output
2 1 1 1

input
5 3 4 4 5 1 4 2 4 3 2 3 1 3 3 5
output
3 1 1 1

Note

In the first sample the fool number one goes on the first and third road and the fool number 3 goes on the second, first and fourth ones.

In the second sample, the fools number 1, 3 and 5 go on the first road, the fool number 5 will go on the second road, on the third road goes the fool number 3, and on the fourth one goes fool number 1.