Problem A. Boris vs Nursik

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Boris and Nursik play a drunkard card game. In the drunkard card game, all cards are divided equally between two players. Then they reveal one top card, and the one whose card is higher takes both of the revealed cards for himself, which are put under the bottom of his deck. The one who is left without cards loses.

The player who takes the cards for himself first puts the Boris's card under the bottom of his deck, then the Nursik's card (that is, the Nursik's card is at the bottom of the deck).

Write a program that simulates the drunkard card game and determines who wins. The game involves 10 cards with values from 0 to 9, the larger card wins the smaller one. The one special thing is that the card with a value of 0 wins card 9.

Input

The program receives two lines as input: the first line contains 5 numbers separated by spaces - the numbers of the Boris's cards, the second - Nursik's 5 cards. For simplicity, we will assume that all cards are different in value. The cards are listed from top to bottom, that is, each line starts with the card that will be opened first.

Output

The program must determine who wins, and output the word *Boris* or *Nursik*, then output the number of moves made before winning. If the game does not end within 10^6 moves, the program should output only *blin nichya*.

Example

standard input	standard output
1 3 5 7 9 2 4 6 8 0	Nursik 5

Problem B. Hungry dude

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Palisher likes to eat bahandi. There are n packs of bahandi burgers in the shop. All workers have gone and will come back in h hours.

Palisher can decide his burgers-per-hour eating speed of k. Each hour, he chooses some pack of burgers, and eats k burgers from that pack. If the pack has less than k burgers, he eats all of them instead, and won't eat any more burgers during this hour.

Palisher likes to eat slowly, but still wants to finish eating all the burgers before the workers come back.

Input

The first line of the input contains integer n and h ($1 \le n \le 10^5$), ($n \le h \le 10^9$) . The second line of the input contains n integer numbers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — number of burgers in each pack.

Output

Output the minimum integer k such that he can eat all the burgers within h hours.

Examples

standard input	standard output
4 8 3 6 7 11	4
5 5 30 11 23 4 20	30
5 6 30 11 23 4 20	23

Problem C. Gambling is not good!

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Inkar loves to go to the races, hoping to earn a big sum on them. She liked the horse with number k and decided to see if it could beat all the other horses. Inkar got hold of information in which for some pairs of horses it is reported which of these horses is faster. She also got info that all horses have different speeds.

It is required to write a program that will help Inkar determine exactly whether the horse of her choice can win. It is guaranteed that the information is not contradictory. For example, if there are 3 horses, it will not be that horse 1 is faster than horse 2, and horse 2 is faster than horse 3, and horse 3 is faster than horse 1.

Input

The first line contains two integer numbers n and k ($1 \le n \le 100$, $1 \le k \le n$) — number of horses taking part in the races and the number of the horse that Inkar wants to bet on.

The next lines contain two numbers x and y ($1 \le x \le n$, $1 \le y \le n$), indicating that the horse with the number x is faster than the horse with the number y. Pairs x and y are not repeated. The dataset is terminated with a line containing a single zero. You do not need to process this line.

Output

The output line should contain the word Yes if Inkar is sure of his winning and No otherwise.

Examples

standard input	standard output
3 1 1 2 1 3 0	Yes
3 2 2 3 0	No
4 2 3 1 2 3 0	No

Problem D. A and B

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 megabytes

Given n lines consisting of Latin letters A and B. You want to count how many of these lines are beautiful.

A string is beautiful if each letter can be connected to exactly one other identical letter using an arc drawn **from above** in such a way that no two arcs intersect. Display the number of beautiful rows.

Input

The first row contains one number n, the number of rows.

In the following $n$lines are given words consisting of capital Latin letters A and B. The length of each word is from 2 to 10^5, inclusive.

Output

In a single line, display one number - the number of beautiful lines.

Examples

standard input	standard output
3 ABAB AABB ABBA	2
4 AAA AA AB ABBABB	2

Note

$1 \leq n \leq 100$

The sum of the lengths of all words will not exceed 10^6.

Problem E. Rock Game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

We have a collection of rocks, each rock has a positive integer weight.

Each turn, we choose the two **heaviest** rocks and smash them together. Suppose the stones have weights x and y with x <= y. The result of this smash is:

If x==y, both stones are totally destroyed; If x != y, the stone of weight x is totally destroyed, and the stone of weight y has new weight y-x. At the end, there is at most 1 stone left. Output the weight of this stone (or 0 if there are no stones left.)

Input

1 <= stones.length <= 30

1 <= stones[i] <= 1000

Output

In a single line print the answer

Example

standard input	standard output
6 2 7 4 1 8 1	1

Note

We combine 7 and 8 to get 1 so the array converts to [2,4,1,1,1] then, we combine 2 and 4 to get 2 so the array converts to [2,1,1,1] then, we combine 2 and 1 to get 1 so the array converts to [1,1,1] then, we combine 1 and 1 to get 0 so the array converts to [1] then that's the value of last stone.

Problem F. Uragirimono no Requiem

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

You are given string $s(1 \leq |s| \leq 5 * 10^5)$ that consists of english lowercase letters, and a dictionary of size $m(1 \leq m \leq 5 * 10^5)$. Determine if it's possible to split into two non-empty strings s1 and s2, such that s1 + s2 = s and both s1 and s2 exist in dictionary.

Input

First line contains string s. Second line contains integer m - number of words in dictionary. The next m lines contain lowercase english letters - words in dictionary. It's guranteed that sum of sizes of words in dictionary $\leq 5 * 10^5$.

Output

Output 'YES' if answer exists and 'NO' otherwise.

Examples

standard input	standard output
goldenwind 2 golden wind	YES
goldenwind 2 goldenw wind	NO
jojoreference 5 jojo reference lol kek d	YES