Sponsored by Telegram

Codeforces Round #256 (Div. 2)

A. Rewards

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

Bizon the Champion is called the Champion for a reason.

Bizon the Champion has recently got a present — a new glass cupboard with n shelves and he decided to put all his presents there. All the presents can be divided into two types: medals and cups. Bizon the Champion has a_1 first prize cups, a_2 second prize cups and a_3 third prize cups. Besides, he has b_1 first prize medals, b_2 second prize medals and b_3 third prize medals.

Naturally, the rewards in the cupboard must look good, that's why Bizon the Champion decided to follow the rules:

- any shelf cannot contain both cups and medals at the same time;
- no shelf can contain more than five cups;
- no shelf can have more than ten medals.

Help Bizon the Champion find out if we can put all the rewards so that all the conditions are fulfilled.

Input

The first line contains integers a_1, a_2 and a_3 $(0 \le a_1, a_2, a_3 \le 100)$. The second line contains integers b_1, b_2 and b_3 $(0 \le b_1, b_2, b_3 \le 100)$. The third line contains integer n $(1 \le n \le 100)$.

The numbers in the lines are separated by single spaces.

Output

Print "`YES`" (without the quotes) if all the rewards can be put on the shelves in the described manner. Otherwise, print " `NO`" (without the quotes).

Sample test(s)

input

```
1 1 1
1 1 1
4
```

output

```
YES
```

input

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

output

```
YES
```

input

```
1 0 0
1 0 0
1
```

output

```
NO
```

B. Suffix Structures

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

Bizon the Champion isn't just a bison. He also is a favorite of the "Bizons" team.

At a competition the "Bizons" got the following problem: "You are given two distinct words (strings of English letters), s and t. You need to transform word s into word t". The task looked simple to the guys because they know the suffix data structures well. Bizon Senior loves suffix automaton. By applying it once to a string, he can remove from this string any single character. Bizon Middle knows suffix array well. By applying it once to a string, he can swap any two characters of this string. The guys do not know anything about the suffix tree, but it can help them do much more.

Bizon the Champion wonders whether the "Bizons" can solve the problem. Perhaps, the solution do not require both data structures. Find out whether the guys can solve the problem and if they can, how do they do it? Can they solve it either only with use of suffix automaton or only with use of suffix array or they need both structures? Note that any structure may be used an unlimited number of times, the structures may be used in any order.

Input

The first line contains a non-empty word s. The second line contains a non-empty word t. Words s and t are different. Each word consists only of lowercase English letters. Each word contains at most 100 letters.

Output

In the single line print the answer to the problem. Print "`need tree`" (without the quotes) if word s cannot be transformed into word t even with use of both suffix array and suffix automaton. Print "`automaton`" (without the quotes) if you need only the suffix automaton to solve the problem. Print "`array`" (without the quotes) if you need only the suffix array to solve the problem. Print "`both`" (without the quotes), if you need both data structures to solve the problem.

It's guaranteed that if you can solve the problem only with use of suffix array, then it is impossible to solve it only with use of suffix automaton. This is also true for suffix automaton.

Sample test(s)

input

```
automaton
tomat
```

output

```
automaton
```

input

```
array
arary
```

output

```
array
```

input

```
both
hot
```

output

```
both
```

input

```
need
tree
```

output

```
need tree
```

Note

In the third sample you can act like that: first transform "`both`" into "`oth`" by removing the first character using the suffix automaton and then make two swaps of the string using the suffix array and get "`hot`".

C. Painting Fence

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

Bizon the Champion isn't just attentive, he also is very hardworking.

Bizon the Champion decided to paint his old fence his favorite color, orange. The fence is represented as n vertical planks, put in a row. Adjacent planks have no gap between them. The planks are numbered from the left to the right starting from one, the i-th plank has the width of 1 meter and the height of a_i meters.

Bizon the Champion bought a brush in the shop, the brush's width is 1 meter. He can make vertical and horizontal strokes with the brush. During a stroke the brush's full surface must touch the fence at all the time (see the samples for the better understanding). What minimum number of strokes should Bizon the Champion do to fully paint the fence? Note that you are allowed to paint the same area of the fence multiple times.

Input

The first line contains integer n $(1 \le n \le 5000)$ — the number of fence planks. The second line contains n space-separated integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^9)$.

Output

Print a single integer — the minimum number of strokes needed to paint the whole fence.

Sample test(s)

input

```
5
2 2 1 2 1
```

output

```
3
```

input

```
2
2 2
```

output

```
2
```

input

```
1
5
```

output

```
1
```

Note

In the first sample you need to paint the fence in three strokes with the brush: the first stroke goes on height 1 horizontally along all the planks. The second stroke goes on height 2 horizontally and paints the first and second planks and the third stroke (it can be horizontal and vertical) finishes painting the fourth plank.

In the second sample you can paint the fence with two strokes, either two horizontal or two vertical strokes.

In the third sample there is only one plank that can be painted using a single vertical stroke.

D. Multiplication Table

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

Bizon the Champion isn't just charming, he also is very smart.

While some of us were learning the multiplication table, Bizon the Champion had fun in his own manner. Bizon the Champion painted an $n \times m$ multiplication table, where the element on the intersection of the i-th row and j-th column equals $i \cdot j$ (the rows and columns of the table are numbered starting from 1). Then he was asked: what number in the table is the k-th largest number? Bizon the Champion always answered correctly and immediately. Can you repeat his success?

Consider the given multiplication table. If you write out all $n \cdot m$ numbers from the table in the non-decreasing order, then the k-th number you write out is called the k-th largest number.

Input

The single line contains integers n, m and k $(1 \le n, m \le 5 \cdot 10^5;\ 1 \le k \le n \cdot m)$.

Output

Print the k-th largest number in a $n \times m$ multiplication table.

Sample test(s)

input
```
2 2 2
```
output
```
2
```

input
```
2 3 4
```
output
```
3
```

input
```
1 10 5
```
output
```
5
```

Note

A 2×3 multiplication table looks like this:

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

E. Divisors

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

Bizon the Champion isn't just friendly, he also is a rigorous coder.

Let's define function $f(a)$, where a is a sequence of integers. Function $f(a)$ returns the following sequence: first all divisors of a_1 go in the increasing order, then all divisors of a_2 go in the increasing order, and so on till the last element of sequence a. For example, $f([2, 9, 1]) = [1, 2, 1, 3, 9, 1]$.

Let's determine the sequence X_i, for integer i $(i \ge 0)$: $X_0 = [X]$ ($[X]$ is a sequence consisting of a single number X), $X_i = f(X_{i-1})$ $(i > 0)$. For example, at $X = 6$ we get $X_0 = [6]$, $X_1 = [1, 2, 3, 6]$, $X_2 = [1, 1, 2, 1, 3, 1, 2, 3, 6]$.

Given the numbers X and k, find the sequence X_k. As the answer can be rather large, find only the first 10^5 elements of this sequence.

Input

A single line contains two space-separated integers — X $(1 \le X \le 10^{12})$ and k $(0 \le k \le 10^{18})$.

Output

Print the elements of the sequence X_k in a single line, separated by a space. If the number of elements exceeds 10^5, then print only the first 10^5 elements.

Sample test(s)

input

```
6 1
```

output

```
1 2 3 6
```

input

```
4 2
```

output

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

input

```
10 3
```

output

```
1 1 1 2 1 1 5 1 1 2 1 5 1 2 5 10
```