A. Kitchen Utensils

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

The king's birthday dinner was attended by k guests. The dinner was quite a success: every person has eaten several dishes (though the number of dishes was the same for every person) and every dish was served alongside with a new set of kitchen utensils.

All types of utensils in the kingdom are numbered from 1 to 100. It is known that every set of utensils is the same and consist of different types of utensils, although every particular type may appear in the set at most once. For example, a valid set of utensils can be composed of one fork, one spoon and one knife.

After the dinner was over and the guests were dismissed, the king wondered what minimum possible number of utensils could be stolen. Unfortunately, the king has forgotten how many dishes have been served for every guest but he knows the list of all the utensils left after the dinner. Your task is to find the minimum possible number of stolen utensils.

Input

The first line contains two integer numbers n and k $(1 \le n \le 100, 1 \le k \le 100)$ — the number of kitchen utensils remaining after the dinner and the number of guests correspondingly.

The next line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 100)$ — the types of the utensils remaining. Equal values stand for identical utensils while different values stand for different utensils.

Output

Output a single value — the minimum number of utensils that could be stolen by the guests.

Examples

input
```
5 2
1 2 2 1 3
```
output
```
1
```

input
```
10 3
1 3 3 1 3 5 5 5 5 100
```
output
```
14
```

Note

In the first example it is clear that at least one utensil of type 3 has been stolen, since there are two guests and only one such utensil. But it is also possible that every person received only one dish and there were only six utensils in total, when every person got a set $(1, 2, 3)$ of utensils. Therefore, the answer is 1.

One can show that in the second example at least 2 dishes should have been served for every guest, so the number of utensils should be at least 24: every set contains 4 utensils and every one of the 3 guests gets two such sets. Therefore, at least 14 objects have been stolen. Please note that utensils of some types (for example, of types 2 and 4 in this example) may be not present in the set served for dishes.

B. Personalized Cup

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

At many competitions that have a word «cup» in its official name the winner is presented with an actual cup. This time the organizers of one unusual programming competition have decided to please the winner even more and to add a nameplate to the cup with the handle of the winner.

The nameplate is to be rectangular and the text on it will be printed as a table of several rows and columns. Having some measurements done, the organizers have found out that the number a of rows cannot be greater than 5 while the number b of columns cannot exceed 20. Every cell of the table will contain either an asterisk («*») or a letter of user's handle.

Furthermore, the organizers want the rows of the table to be uniform, which means that the number of asterisks used in different rows should differ by at most one (i.e. you can't have two asterisks in the first row and none in the second). The main goal, however, is to obtain the winner's handle precisely when reading the table from top to bottom and from left to right in every row (skipping asterisks).

The organizers want for the nameplate to have as few rows as possible and among all valid tables with the minimum number of rows they want to choose the one that has the minimum number of columns.

The winner is not yet determined so your task is to write a program that, given a certain handle, generates the necessary table.

Input

The only line contains one string s ($1 \le |s| \le 100$), comprised of uppercase and lowercase Latin letters, — the handle of the winner.

Output

In the first line output the minimum number a of rows in the table and the minimum number b of columns in an optimal table with rows.

The following a lines should contain b characters each — any valid table.

Examples

input
tourist

output
1 7 tourist

input
MyNameIsLifeIAmForeverByYourSideMyNameIsLife

output
3 15 MyNameIsLifeIAm ForeverByYourSi deMyNameIsL*ife

C. Playing Piano

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

Little Paul wants to learn how to play piano. He already has a melody he wants to start with. For simplicity he represented this melody as a sequence $a_1, a_2, \ldots, a_n$ of key numbers: the more a number is, the closer it is to the right end of the piano keyboard.

Paul is very clever and knows that the essential thing is to properly assign fingers to notes he's going to play. If he chooses an inconvenient fingering, he will then waste a lot of time trying to learn how to play the melody by these fingers and he will probably not succeed.

Let's denote the fingers of hand by numbers from 1 to 5. We call a *fingering* any sequence $b_1, \ldots, b_n$ of fingers numbers. A fingering is *convenient* if for all $1 \le i \le n-1$ the following holds:

- if $a_i < a_{i+1}$ then $b_i < b_{i+1}$, because otherwise Paul needs to take his hand off the keyboard to play the $(i+1)$-st note;
- if $a_i > a_{i+1}$ then $b_i > b_{i+1}$, because of the same;
- if $a_i = a_{i+1}$ then $b_i \neq b_{i+1}$, because using the same finger twice in a row is dumb. **Please note that there is $\neq$, not $=$ between b_i and b_{i+1}.**

Please provide any convenient fingering or find out that there is none.

Input

The first line contains a single integer n ($1 \le n \le 10^5$) denoting the number of notes.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 2 \cdot 10^5$) denoting the positions of notes on the keyboard.

Output

If there is no convenient fingering, print -1. Otherwise, print n numbers $b_1, b_2, \ldots, b_n$, each from 1 to 5, denoting a convenient fingering, separated by spaces.

Examples

input
5 1 1 4 2 2

output

```
1 4 5 4 5
```

input

```
7
1 5 7 8 10 3 1
```

output

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

input

```
19
3 3 7 9 8 8 8 8 7 7 7 7 5 3 3 3 3 8 8
```

output

```
1 3 4 5 4 5 4 5 4 5 4 5 4 3 5 4 3 5 4
```

Note

The third sample test is kinda "Non stop" song by Reflex.

D. Barcelonian Distance

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

In this problem we consider a very simplified model of Barcelona city.

Barcelona can be represented as a plane with streets of kind $x = c$ and $y = c$ for every integer c (that is, the rectangular grid). However, there is a detail which makes Barcelona different from Manhattan. There is an avenue called Avinguda Diagonal which can be represented as a the set of points (x, y) for which $ax + by + c = 0$.

One can walk along streets, including the avenue. You are given two integer points A and B somewhere in Barcelona. Find the minimal possible distance one needs to travel to get to B from A.

Input

The first line contains three integers a, b and c ($-10^9 \leq a, b, c \leq 10^9$, at least one of a and b is not zero) representing the Diagonal Avenue.

The next line contains four integers x_1, y_1, x_2 and y_2 ($-10^9 \leq x_1, y_1, x_2, y_2 \leq 10^9$) denoting the points $A = (x_1, y_1)$ and $B = (x_2, y_2)$.

Output

Find the minimum possible travel distance between A and B. Your answer is considered correct if its absolute or relative error does not exceed 10^{-6}.

Formally, let your answer be a, and the jury's answer be b. Your answer is accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \leq 10^{-6}$.

Examples

input

```
1 1 -3
0 3 3 0
```

output

```
4.2426406871
```

input

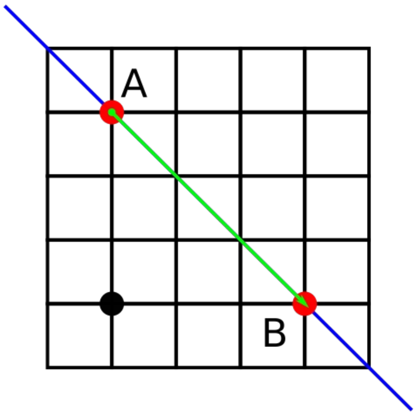
```
3 1 -9
0 3 3 -1
```

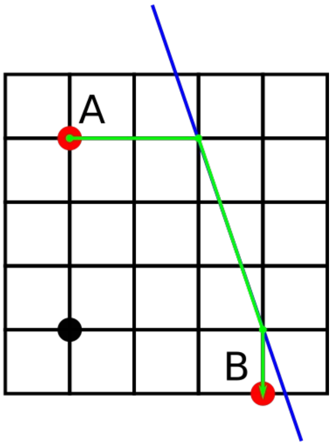
output

```
6.1622776602
```

Note

The first example is shown on the left picture while the second example us shown on the right picture below. The avenue is shown with blue, the origin is shown with the black dot.





E. The Unbearable Lightness of Weights

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

You have a set of n weights. You know that their masses are a_1, a_2, ..., a_n grams, but you don't know which of them has which mass. You can't distinguish the weights.

However, your friend does know the mass of each weight. You can ask your friend to give you exactly k weights with the total mass m (both parameters k and m are chosen by you), and your friend will point to any valid subset of weights, if it is possible.

You are allowed to make this query only once. Find the maximum possible number of weights you can reveal after this query.

Input

The first line contains a single integer n $(1 \le n \le 100)$ — the number of weights.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 100)$ — the masses of the weights.

Output

Print the maximum number of weights you can learn the masses for after making a single query.

Examples

input

```
4
1 4 2 2
```

output

```
2
```

input

```
6
1 2 4 4 4 9
```

output

```
2
```

Note

In the first example we can ask for a subset of two weights with total mass being equal to 4, and the only option is to get $\{2, 2\}$.

Another way to obtain the same result is to ask for a subset of two weights with the total mass of 5 and get $\{1, 4\}$. It is easy to see that the two remaining weights have mass of 2 grams each.

In the second example we can ask for a subset of two weights with total mass being 8, and the only answer is $\{4, 4\}$. We can prove it is not possible to learn masses for three weights in one query, but we won't put the proof here.

F. Vasya and Maximum Matching

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

Vasya has got a tree consisting of n vertices. He wants to delete some (possibly zero) edges in this tree such that the maximum matching in the resulting graph is unique. He asks you to calculate the number of ways to choose a set of edges to remove.

A matching in the graph is a subset of its edges such that there is no vertex incident to two (or more) edges from the subset. A maximum matching is a matching such that the number of edges in the subset is maximum possible among all matchings in this graph.

Since the answer may be large, output it modulo 998244353.

Input

The first line contains one integer n ($1 \le n \le 3 \cdot 10^5$) — the number of vertices in the tree.

Each of the next $n-1$ lines contains two integers u and v ($1 \le u, v \le n, u \ne v$) denoting an edge between vertex u and vertex v. It is guaranteed that these edges form a tree.

Output

Print one integer — the number of ways to delete some (possibly empty) subset of edges so that the maximum matching in the resulting graph is unique. Print the answer modulo 998244353.

Examples

input

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

output

```
4
```

input

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

output

```
6
```

input

```
1
```

output

```
1
```

Note

Possible ways to delete edges in the first example:

- delete $(1,2)$ and $(1,3)$.
- delete $(1,2)$ and $(1,4)$.
- delete $(1,3)$ and $(1,4)$.
- delete all edges.

Possible ways to delete edges in the second example:

- delete no edges.
- delete $(1,2)$ and $(2,3)$.
- delete $(1,2)$ and $(3,4)$.
- delete $(2,3)$ and $(3,4)$.
- delete $(2,3)$.
- delete all edges.

G. Chattering

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

There are n parrots standing in a circle. Each parrot has a certain level of respect among other parrots, namely r_i. When a parrot with respect level x starts chattering, x neighbours to the right and to the left of it start repeating the same words in 1 second. Their neighbours then start repeating as well, and so on, until all the birds begin to chatter.

You are given the respect levels of all parrots. For each parrot answer a question: if this certain parrot starts chattering, how many seconds will pass until all other birds will start repeating it?

Input

In the first line of input there is a single integer n, the number of parrots ($1 \le n \le 10^5$).

In the next line of input there are n integers r_1, ..., r_n, the respect levels of parrots in order they stand in the circle ($1 \le r_i \le n$).

Output

Print n integers. i-th of them should equal the number of seconds that is needed for all parrots to start chattering if the i-th parrot is the first to start.

Examples

input

```
4
1 1 4 1
```

output

```
2 2 1 2
```

input

```
8
1 2 2 1 5 1 3 1
```

output

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