Educational Codeforces Round 14

A. Fashion in Berland

1 second, 256 megabytes

According to rules of the Berland fashion, a jacket should be fastened by `all the buttons except only one`, but not necessarily it should be the last one. Also if the jacket has only one button, it should be fastened, so the jacket will not swinging open.

You are given a jacket with n buttons. Determine if it is fastened in a right way.

Input

The first line contains integer n $(1 \le n \le 1000)$ — the number of buttons on the jacket.

The second line contains n integers a_i $(0 \le a_i \le 1)$. The number $a_i = 0$ if the i-th button is not fastened. Otherwise $a_i = 1$.

Output

In the only line print the word "`YES`" if the jacket is fastened in a right way. Otherwise print the word "`NO`".

input
```
3
1 0 1
```
output
```
YES
```

input
```
3
1 0 0
```
output
```
NO
```

B. s-palindrome

1 second, 256 megabytes

Let's call a string "`s-palindrome`" if it is symmetric about the middle of the string. For example, the string "`oHo`" is "`s-palindrome`", but the string "`aa`" is not. The string "`aa`" is not "`s-palindrome`", because the second half of it is not a mirror reflection of the first half.

A a B b C c D d E e F f G g H h I i J j K k L l M m
N n O o P p Q q R r S s T t U u V v W w X x Y y Z z

English alphabet

You are given a string s. Check if the string is "`s-palindrome`".

Input

The only line contains the string s $(1 \le |s| \le 1000)$ which consists of only English letters.

Output

Print "`TAK`" if the string s is "`s-palindrome`" and "`NIE`" otherwise.

input
```
oXoxoXo
```
output
```
TAK
```

input
```
bod
```
output
```
TAK
```

input
```
ER
```
output
```
NIE
```

C. Exponential notation

2 seconds, 256 megabytes

You are given a positive decimal number x.

Your task is to convert it to the "`simple exponential notation`".

Let $x = a \cdot 10^b$, where $1 \le a < 10$, then in general case the "`simple exponential notation`" looks like "`aEb`". If b equals to zero, the part "`Eb`" should be skipped. If a is an integer, it should be written without decimal point. Also there should not be extra zeroes in a and b.

Input

The only line contains the positive decimal number x. The length of the line will not exceed 10^6. Note that you are given too large number, so you can't use standard built-in data types "`float`", "`double`" and other.

Output

Print the only line — the "`simple exponential notation`" of the given number x.

input
```
16
```
output
```
1.6E1
```

input
```
01.23400
```
output
```
1.234
```

input
```
.100
```
output
```
1E-1
```

input
```
100.
```
output
```
1E2
```

D. Swaps in Permutation

5 seconds, 256 megabytes

You are given a permutation of the numbers $1, 2, ..., n$ and m pairs of positions (a_j, b_j).

At each step you can choose a pair from the given positions and swap the numbers in that positions. What is the lexicographically maximal permutation one can get?

Let p and q be two permutations of the numbers $1, 2, ..., n$. p is lexicographically smaller than the q if a number $1 \le i \le n$ exists, so $p_k = q_k$ for $1 \le k < i$ and $p_i < q_i$.

Input

The first line contains two integers n and m $(1 \le n, m \le 10^6)$ — the length of the permutation p and the number of pairs of positions.

The second line contains n distinct integers p_i $(1 \le p_i \le n)$ — the elements of the permutation p.

Each of the last m lines contains two integers (a_j, b_j) $(1 \le a_j, b_j \le n)$ — the pairs of positions to swap. Note that you are given a `positions`, not the values to swap.

Output

Print the only line with n distinct integers p'_i $(1 \le p'_i \le n)$ — the lexicographically maximal permutation one can get.

input
```
9 6
1 2 3 4 5 6 7 8 9
1 4
4 7
2 5
5 8
3 6
6 9
```
output
```
7 8 9 4 5 6 1 2 3
```

E. Xor-sequences

3 seconds, 256 megabytes

You are given n integers $a_1, a_2, ..., a_n$.

A sequence of integers $x_1, x_2, ..., x_k$ is called a "xor-sequence" if for every $1 \leq i \leq k-1$ the number of ones in the binary representation of the number $x_i \otimes x_{i+1}$'s is a multiple of 3 and $x_i \in \{a_1, a_2, \ldots, a_n\}$ for all $1 \leq i \leq k$. The symbol $\otimes$ is used for the binary exclusive or operation.

How many "xor-sequences" of length k exist? Output the answer modulo $10^9 + 7$.

Note if $a = [1, 1]$ and $k = 1$ then the answer is 2, because you should consider the ones from a as different.

Input

The first line contains two integers n and k ($1 \leq n \leq 100$, $1 \leq k \leq 10^{18}$) — the number of given integers and the length of the "xor-sequences".

The second line contains n integers a_i ($0 \leq a_i \leq 10^{18}$).

Output

Print the only integer c — the number of "xor-sequences" of length k modulo $10^9 + 7$.

input
5 2 15 1 2 4 8
output
13

input
5 1 15 1 2 4 8
output
5

F. Couple Cover

3 seconds, 512 megabytes

Couple Cover, a wildly popular luck-based game, is about to begin! Two players must work together to construct a rectangle. A bag with n balls, each with an integer written on it, is placed on the table. The first player reaches in and grabs a ball randomly (all balls have equal probability of being chosen) — the number written on this ball is the rectangle's width in meters. This ball is not returned to the bag, and the second player reaches into the bag and grabs another ball — the number written on this ball is the rectangle's height in meters. If the area of the rectangle is greater than or equal some threshold p square meters, the players win. Otherwise, they lose.

The organizers of the game are trying to select an appropriate value for p so that the probability of a couple winning is not too high and not too low, but they are slow at counting, so they have hired you to answer some questions for them. You are given a list of the numbers written on the balls, the organizers would like to know how many winning pairs of balls exist for different values of p. Note that two pairs are different if either the first or the second ball is different between the two in pair, and two different balls with the same number are considered different.

Input

The input begins with a single positive integer n in its own line ($1 \leq n \leq 10^6$).

The second line contains n positive integers — the i-th number in this line is equal to a_i ($1 \leq a_i \leq 3 \cdot 10^6$), the number written on the i-th ball.

The next line contains an integer m ($1 \leq m \leq 10^6$), the number of questions you are being asked.

Then, the following line contains m positive integers — the j-th number in this line is equal to the value of p ($1 \leq p \leq 3 \cdot 10^6$) in the j-th question you are being asked.

Output

For each question, print the number of winning pairs of balls that exist for the given value of p in the separate line.

input
5 4 2 6 1 3 4 1 3 5 8

output

```
20
18
14
10
```

input

```
2
5 6
2
30 31
```

output

```
2
0
```