# A. GCD Table

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

The GCD table $G$ of size $n \times n$ for an array of positive integers $a$ of length $n$ is defined by formula

$$g_{ij} = \gcd(a_i, a_j).$$

Let us remind you that the greatest common divisor (GCD) of two positive integers $x$ and $y$ is the greatest integer that is divisor of both $x$ and $y$, it is denoted as $\gcd(x, y)$. For example, for array $a = \{4, 3, 6, 2\}$ of length 4 the GCD table will look as follows:

Given all the numbers of the GCD table $G$, restore array $a$.

### Input

The first line contains number $n$ ($1 \le n \le 500$) — the length of array $a$. The second line contains $n^2$ space-separated numbers — the elements of the GCD table of $G$ for array $a$.

All the numbers in the table are positive integers, not exceeding $10^9$. Note that the elements are given in an arbitrary order. It is guaranteed that the set of the input data corresponds to some array $a$.

### Output

In the single line print $n$ positive integers — the elements of array $a$. If there are multiple possible solutions, you are allowed to print any of them.

#### Examples

**input**

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

**output**

```
4 3 6 2
```

**input**

```
1
42
```

**output**

```
42
```

**input**

```
2
1 1 1 1
```

**output**

```
1 1
```

# B. Once Again...

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

You are given an array of positive integers $a_1, a_2, ..., a_{n \times T}$ of length $n \times T$. We know that for any $i > n$ it is true that $a_i = a_{i-n}$. Find the length of the longest non-decreasing sequence of the given array.

## Input

The first line contains two space-separated integers: $n$, $T$ ($1 \le n \le 100$, $1 \le T \le 10^7$). The second line contains $n$ space-separated integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 300$).

## Output

Print a single number — the length of a sought sequence.

### Examples

**input**

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

**output**

```
5
```

## Note

The array given in the sample looks like that: 3, **1**, 4, **2**, **3**, 1, **4**, 2, 3, 1, **4**, 2. The elements in bold form the largest non-decreasing subsequence.

# C. Superior Periodic Subarrays

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

You are given an infinite periodic array $a_0, a_1, \ldots, a_{n-1}, \ldots$ with the period of length $n$. Formally, $a_i = a_{i \bmod n}$. A periodic subarray $(l, s)$ ($0 \le l < n$, $1 \le s < n$) of array $a$ is an infinite periodic array with a period of length $s$ that is a subsegment of array $a$, starting with position $l$.

A periodic subarray $(l, s)$ is *superior*, if when attaching it to the array $a$, starting from index $l$, any element of the subarray is larger than or equal to the corresponding element of array $a$. An example of attaching is given on the figure (top — infinite array $a$, bottom — its periodic subarray $(l, s)$):

Find the number of distinct pairs $(l, s)$, corresponding to the superior periodic arrays.

### Input

The first line contains number $n$ ($1 \le n \le 2 \cdot 10^5$). The second line contains $n$ numbers $a_0, a_1, \ldots, a_{n-1}$ ($1 \le a_i \le 10^6$), separated by a space.

### Output

Print a single integer — the sought number of pairs.

#### Examples

**input**

```
4
7 1 2 3
```

**output**

```
2
```

**input**

```
2
2 1
```

**output**

```
1
```

**input**

```
3
1 1 1
```

**output**

```
6
```

### Note

In the first sample the superior subarrays are (0, 1) and (3, 2).

Subarray (0, 1) is superior, as $a_0 \ge a_0, a_0 \ge a_1, a_0 \ge a_2, a_0 \ge a_3, a_0 \ge a_0, \ldots$

Subarray (3, 2) is superior $a_3 \ge a_3, a_0 \ge a_0, a_3 \ge a_1, a_0 \ge a_2, a_3 \ge a_3, \ldots$

In the third sample any pair of $(l, s)$ corresponds to a superior subarray as all the elements of an array are distinct.

# D. Number of Binominal Coefficients

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

For a given prime integer $p$ and integers $\alpha, A$ calculate the number of pairs of integers $(n, k)$, such that $0 \le k \le n \le A$ and $\binom{n}{k}$ is divisible by $p^{\alpha}$.

As the answer can be rather large, print the remainder of the answer moduly $10^9 + 7$.

Let us remind you that $\binom{n}{k}$ is the number of ways $k$ objects can be chosen from the set of $n$ objects.

## Input

The first line contains two integers, $p$ and $\alpha$ ($1 \le p, \alpha \le 10^9$, $p$ is prime).

The second line contains the decimal record of integer $A$ ($0 \le A < 10^{1000}$) without leading zeroes.

## Output

In the single line print the answer to the problem.

### Examples

**input**

```
2 2
7
```

**output**

```
3
```

**input**

```
3 1
9
```

**output**

```
17
```

**input**

```
3 3
9
```

**output**

```
0
```

**input**

```
2 4
5000
```

**output**

```
8576851
```

## Note

In the first sample three binominal coefficients divisible by 4 are $\binom{4}{1} = 4$, $\binom{4}{3} = 4$ and $\binom{6}{3} = 20$.

# E. Boolean Function

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In this problem we consider Boolean functions of four variables $A, B, C, D$. Variables $A, B, C$ and $D$ are logical and can take values 0 or 1. We will define a function using the following grammar:

```
<expression> ::= <variable> | (<expression>) <operator> (<expression>)
```

```
<variable> ::= 'A' | 'B' | 'C' | 'D' | 'a' | 'b' | 'c' | 'd'
```

```
<operator> ::= '&' | '|'
```

Here large letters $A, B, C, D$ represent variables, and small letters represent their negations. For example, if $A = 1$, then character `'A'` corresponds to value 1, and value character `'a'` corresponds to value 0. Here character '&' corresponds to the operation of logical AND, character '|' corresponds to the operation of logical OR.

You are given expression $s$, defining function $f$, where some operations and variables are missing. Also you know the values of the function $f(A, B, C, D)$ for some $n$ distinct sets of variable values. Count the number of ways to restore the elements that are missing in the expression so that the resulting expression corresponded to the given information about function $f$ in the given variable sets. As the value of the result can be rather large, print its remainder modulo $10^9 + 7$.

### Input

The first line contains expression $s$ ($1 \le |s| \le 500$), where some characters of the operators and/or variables are replaced by character '?'.

The second line contains number $n$ ($0 \le n \le 2^4$) — the number of integers sets for which we know the value of function $f(A, B, C, D)$. Next $n$ lines contain the descriptions of the sets: the $i$-th of them contains five integers $a_i, b_i, c_i, d_i, e_i$ ($0 \le a_i, b_i, c_i, d_i, e_i \le 1$), separated by spaces and meaning that $f(a_i, b_i, c_i, d_i) = e_i$.

It is guaranteed that all the tuples $(a_i, b_i, c_i, d_i)$ are distinct.

### Output

In a single line print the answer to the problem.

#### Examples

**input**

```
?
2
1 0 1 0 1
0 1 1 0 1
```

**output**

```
2
```

**input**

```
(A)?(?)
1
1 1 0 0 0
```

**output**

```
4
```

**input**

```
((?)&(?))|((?)&(?))
0
```

**output**

```
4096
```

**input**

```
b
1
1 0 1 1 1
```

**output**

```
1
```

### Note

In the first sample the two valid expressions are 'C' and 'd'.

In the second sample the expressions look as follows: '(A)&(a)', '(A)&(b)', '(A)&(C)', '(A)&(D)'.