Problem A. $2x + 2$

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

bobo has n integers $1, 2, \dots, n$ and uses them to play a game.

He would like to choose a subset S of $\{1, 2, \dots, n\}$ such that for all $x \in S$, $(2x + 2) \notin S$.

Now he is curious about the maximum size of S.

Input

The first line contains an integer n ($1 \le n < 10^{100}$).

Output

A single integer denotes the maximum size.

Examples

standard input	standard output
4	3
10000000000	6666666667

Problem B. Escape Sequences

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

For a string s consisting of only "a" and "b" , let $f(s)$ be the string obtained by replacing all "a" in s with "aa" and "b" with "ab". For example, $f(\text{"aba"}) = \text{"aaabaa"}$.

Given strings s and t, determine the smallest non-negative integer k where t is a consecutive substring of $f^k(s)$.

Note that f^k is defined by:

- $f^0(s) = s$;
- $f^k(s) = f^{k-1}(f(s))$.

Input

The first and second lines contain string s and t respectively ($1 \le |s|, |t| \le 2 \cdot 10^5$).

Strings s and t consist of only characters "a" and "b".

Output

A single integer denotes the minimum k.

If k does not exists, print "-1" instead.

Examples

standard input	standard output
b ab	1
ababa bab	0
a b	-1

Problem C. Balls and Holes

Input file:	*standard input*
Output file:	*standard output*
Time limit:	0.5 seconds
Memory limit:	512 mebibytes

bobo invents a game and keeps playing.

A game $(\{a_1, a_2, \ldots, a_m\}, \{b_1, b_2, \ldots, b_l\})$ is played on the axis. First, bobo places m balls at $a_1, a_2, \ldots, a_m$, respectively. Then bobo digs l holes at $b_1 + 0.5, b_2 + 0.5, \ldots, b_l + 0.5$. Finally bobo pushes all balls forward so that the balls fall into the holes. bobo wins if and only if there are odd number of holes containing at least one ball.

Now bobo has n sets $S_1, S_2, \ldots, S_n$, and he wants to know how many games as (S_i, S_j) $(i < j)$ he can win.

Input

The first line contains an integer n $(2 \le n \le 5000)$.

Each of the following n lines contains an integer k_i, which denotes the size of S_i, followed by k_i distinct integers $S_{i,1}, S_{i,2}, \ldots, S_{i,k_i}$ which denotes the set S_i $(1 \le k_i \le 50, 1 \le S_{i,j} \le 50)$.

Output

A single integer denotes the number games bobo can win.

Examples

standard input	standard output
2 1 1 2 1 2	1
2 2 1 2 2 2 1	0

Problem D. Simple Polygon

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Although bobo is truly smart, he just like you to find a simple polygon for him. The polygon you are going to find should satisfy the following conditions.

1. The polygon is simple. That is to say, any two non-adjacent edges won't intersect or touch and any two adjacent edges have exactly one common point.
2. Edges of the polygon are parallel to either x-axis or y-axis.
3. The perimeter of the polygon equals to l, while the area equals to s.

Input

First line of the input contains two integers l and s ($4 \le l \le 10^9, 1 \le s \le 10^9$).

Output

The first line contains an integer n, which denotes the number of vertices of the polygon you have found ($4 \le n \le 1000$).

Each of the following n lines contains 2 integers x_i, y_i, which denote the coordinates of points (in clockwise or counter-clockwise order) ($0 \le x_i, y_i \le 10^9$).

Any appropriate solution will get accepted.

If no such polygon can be found, simply print "-1".

Examples

standard input	standard output
4 1	4 0 0 1 0 1 1 0 1
4 2	-1

Problem E. GCD vs LCM

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

bobo is good at GCD (greatest common divisor) and LCM (least common multiple).

But today he gets stuck in summing up $\mathrm{lcm}(i, j)$ for all $1 \le i \le n, 1 \le j \le m$ with $\gcd(i, j) \le a$, modulo $(10^9 + 7)$.

Input

The first line contains an integer q, which denotes the number of questions ($1 \le q \le 10^4$).

Each of the following q lines contains 3 integers n, m, a, as described in the statement ($1 \le n, m, a \le 10^5$).

Output

For each question, print a single integer denoting the sum.

Examples

standard input	standard output
2 2 2 1 3 4 2	5 45

Problem F. Saddle Point

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

bobo has a matrix of size $n \times m$, whose elements are integers from $[1, k]$.

Find out the number of matrices with at least one *saddle point*, modulo $(10^9 + 7)$.

Note that a *saddle point* is a position (i, j) which is both strict maximum of the i-th row and j-th column.

Input

3 integers n, m, k $(1 \le n, m \le 500, 1 \le k \le 10)$.

Output

A single integer denotes the number of matrices.

Examples

standard input	standard output
2 2 2	6
500 500 2	48326276

Problem G. Or Max

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

bobo has a sequence $a_1, a_2, \ldots, a_n$. He would like to choose k consecutive elements and maximize the value S that is defined as their maximum plus their bitwise or.

For all $1 \le k \le n$, find the maximal value bobo can achieve.

Input

The first line contains an integer n ($1 \le n \le 10^5$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i < 2^{16}$).

Output

n integers, where the i-th integer is maximal S for $k = i$.

Examples

standard input	standard output
3 1 0 2	4 4 5

Problem H. Subspace

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

bobo is a big fan of linear algebra! He plans to count the number of k-dimension subspaces in $\mathbb{F}_q^n$ modulo p.

For those who are not familiar with linear algebra:

- $\mathbb{F}_q$ is the set $\{0, 1, \dots, q-1\}$, with addition and multiplication modulo q defined on;
- $\mathbb{F}_q^n$ is the n-dimension vector space $\{(x_1, x_2, \dots, x_n) : x_1, x_2, \dots, x_n \in \mathbb{F}_q\}$;
- A subset $K \subseteq \mathbb{F}_q^n$ is a subspace, if and only if for all $\mathbf{p}, \mathbf{q} \in K$, $\mathbf{p} + \mathbf{q} \in K$;
- The dimension of subspace K is the cardinality of the maximal independent subset;
- A subset $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k\} \subseteq K$ is called independent if and only if equation $c_1 \cdot \mathbf{p}_1 + c_2 \cdot \mathbf{p}_2 + \cdots + c_k \cdot \mathbf{p}_k = 0$ has only solution $c_1 = c_2 = \cdots = c_k = 0$.

Input

4 integers q, n, k, p $(2 \le q \le 10^9, 1 \le k \le n \le 10^9, 2 \le p \le 2 \cdot 10^5)$.

It is guaranteed that p and q are prime numbers.

Output

A single integer denotes the number of subspaces.

Examples

standard input	standard output
2 3 2 100003	7

Problem I. Intervals

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

bobo draws n intervals on the axis, which are conveniently numbered by $1, 2, \ldots, n$. As an excellent mathematician, he managed to set all n intervals of length 10^6.

Then bobo carefully computes $I_{i,j}$, the length of the intersection of intervals i and j, and discards all intervals. However, bobo wants to check his calculations and he is eager to know whether the result **can be** correct.

In another word, determine if there exists n intervals of length 10^6 providing the same result.

Input

The first line contains an integer n ($1 \le n \le 1000$).

Each of the following n lines contains n integers $I_{i,1}, I_{i,2}, \ldots, I_{i,n}$ ($0 \le I_{i,j} \le 10^6$).

Since bobo knows math well, it is guaranteed that $I_{i,j} = I_{j,i}$ and $I_{i,i} = 10^6$.

Output

If for given $I_{i,j}$ it is possible to find at least one appropriate set of intervals, print "`Yes`". Otherwise, print "`No`".

Examples

standard input	standard output
3 1000000 500000 0 500000 1000000 500000 0 500000 1000000	Yes
3 1000000 500000 500000 500000 1000000 500000 500000 500000 1000000	No

Problem J. Power of XOR

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4.5 seconds
Memory limit:	512 mebibytes

bobo has a set of n integers $\{a_1, a_2, \dots, a_n\}$. He randomly picks a subset $\{x_1, x_2, \dots, x_m\}$ (each subset has equal probability to be picked), and would like to know the expectation of $[\text{popcount}(x_1 \oplus x_2 \oplus \cdots \oplus x_m)]^k$.

Note that $\text{popcount}(x)$ is the number of ones in the binary notation of x, and $\oplus$ denotes bitwise exclusive-or.

Input

The first line contains 2 integers n, k ($1 \le n \le 44, 1 \le k \le 10^9$).

The second line contains n integers $a_1, a_2, \dots, a_n$ ($0 \le a_i < 2^{44}$).

Output

If the expectation is E, print a single integer denotes $E \cdot 2^n \bmod (10^9 + 7)$.

Examples

standard input	standard output
3 2 1 2 3	12
2 1000000000 1 2	140625003