Codecraft-18 and Codeforces Round #458 (Div. 1 + Div. 2, combined)

A. Perfect Squares

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

Given an array $a_1, a_2, \ldots, a_n$ of n integers, find the largest number in the array that is not a perfect square.

A number x is said to be a perfect square if there exists an integer y such that $x = y^2$.

Input

The first line contains a single integer n ($1 \le n \le 1000$) — the number of elements in the array.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($-10^6 \le a_i \le 10^6$) — the elements of the array.

It is guaranteed that at least one element of the array is not a perfect square.

Output

Print the largest number in the array which is not a perfect square. It is guaranteed that an answer always exists.

Examples

input
2 4 2
output
2

input
8 1 2 4 8 16 32 64 576
output
32

Note

In the first sample case, 4 is a perfect square, so the largest number in the array that is not a perfect square is 2.

B. Conan and Agasa play a Card Game

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

Edogawa Conan got tired of solving cases, and invited his friend, Professor Agasa, over. They decided to play a game of cards. Conan has n cards, and the i-th card has a number a_i written on it.

They take turns playing, starting with Conan. In each turn, the player chooses a card and removes it. Also, he removes all cards having a number strictly lesser than the number on the chosen card. Formally, if the player chooses the i-th card, he removes that card and removes the j-th card for all j such that $a_j < a_i$.

A player loses if he cannot make a move on his turn, that is, he loses if there are no cards left. Predict the outcome of the game, assuming both players play optimally.

Input

The first line contains an integer n ($1 \le n \le 10^5$) — the number of cards Conan has.

The next line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^5$), where a_i is the number on the i-th card.

Output

If Conan wins, print "`Conan`" (without quotes), otherwise print "`Agasa`" (without quotes).

Examples

input
3 4 5 7
output
Conan

input
2 1 1
output
Agasa

Note

In the first example, Conan can just choose the card having number 7 on it and hence remove all the cards. After that, there are no cards left on Agasa's turn.

In the second example, no matter which card Conan chooses, there will be one one card left, which Agasa can choose. After that, there are no cards left when it becomes Conan's turn again.

C. Travelling Salesman and Special Numbers

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

The Travelling Salesman spends a lot of time travelling so he tends to get bored. To pass time, he likes to perform operations on numbers. One such operation is to take a positive integer x and reduce it to the number of bits set to 1 in the binary representation of x. For example for number 13 it's true that $13_{10} = 1101_2$, so it has 3 bits set and 13 will be reduced to 3 in one operation.

He calls a number *special* if the minimum number of operations to reduce it to 1 is k.

He wants to find out how many special numbers exist which are not greater than n. Please help the Travelling Salesman, as he is about to reach his destination!

Since the answer can be large, output it modulo $10^9 + 7$.

Input

The first line contains integer n ($1 \leq n < 2^{1000}$).

The second line contains integer k ($0 \leq k \leq 1000$).

Note that n is given in its binary representation without any leading zeros.

Output

Output a single integer — the number of special numbers not greater than n, modulo $10^9 + 7$.

Examples

input
110 2
output
3

input
111111011 2
output
169

Note

In the first sample, the three special numbers are 3, 5 and 6. They get reduced to 2 in one operation (since there are two set bits in each of 3, 5 and 6) and then to 1 in one more operation (since there is only one set bit in 2).

D. Bash and a Tough Math Puzzle

time limit per test: 2.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Bash likes playing with arrays. He has an array $a_1, a_2, \ldots a_n$ of n integers. He likes to guess the greatest common divisor (gcd) of different segments of the array. Of course, sometimes the guess is not correct. However, Bash will be satisfied if his guess is *almost correct*.

Suppose he guesses that the gcd of the elements in the range $[l, r]$ of a is x. He considers the guess to be almost correct if he can change **at most** one element in the segment such that the gcd of the segment is x after making the change. Note that when he guesses, he doesn't actually change the array — he just wonders if the gcd of the segment can be made x. Apart from this, he also sometimes makes changes to the array itself.

Since he can't figure it out himself, Bash wants you to tell him which of his guesses are almost correct. Formally, you have to process q queries of one of the following forms:

- $1\ l\ r\ x$ — Bash guesses that the gcd of the range $[l, r]$ is x. Report if this guess is almost correct.
- $2\ i\ y$ — Bash sets a_i to y.

Note: The array is 1-indexed.

Input

The first line contains an integer n ($1 \le n \le 5 \cdot 10^5$) — the size of the array.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the elements of the array.

The third line contains an integer q ($1 \le q \le 4 \cdot 10^5$) — the number of queries.

The next q lines describe the queries and may have one of the following forms:

- $1\ l\ r\ x$ ($1 \le l \le r \le n$, $1 \le x \le 10^9$).
- $2\ i\ y$ ($1 \le i \le n$, $1 \le y \le 10^9$).

Guaranteed, that there is at least one query of first type.

Output

For each query of first type, output "YES" (without quotes) if Bash's guess is almost correct and "NO" (without quotes) otherwise.

Examples

input
3 2 6 3 4 1 1 2 2 1 1 3 3 2 1 9 1 1 3 2

output
YES YES NO

input
5 1 2 3 4 5 6 1 1 4 2 2 3 6 1 1 4 2 1 1 5 2 2 5 10 1 1 5 2

output
NO YES NO YES

Note

In the first sample, the array initially is $\{2, 6, 3\}$.

For query 1, the first two numbers already have their gcd as 2.

For query 2, we can achieve a gcd of 3 by changing the first element of the array to 3. Note that the changes made during queries of type 1 are temporary and do not get reflected in the array.

After query 3, the array is now $\{9, 6, 3\}$.

For query 4, no matter which element you change, you cannot get the gcd of the range to be 2.

E. Palindromes in a Tree

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

You are given a tree (a connected acyclic undirected graph) of n vertices. Vertices are numbered from 1 to n and each vertex is assigned a character from `a` to `t`.

A path in the tree is said to be palindromic if at least one permutation of the labels in the path is a palindrome.

For each vertex, output the number of palindromic paths passing through it.

Note: The path from vertex u to vertex v is considered to be the same as the path from vertex v to vertex u, and this path will be counted only once for each of the vertices it passes through.

Input

The first line contains an integer n ($2 \le n \le 2 \cdot 10^5$) — the number of vertices in the tree.

The next n - 1 lines each contain two integers u and v ($1 \le u, v \le n, u \ne v$) denoting an edge connecting vertex u and vertex v. It is guaranteed that the given graph is a tree.

The next line contains a string consisting of n lowercase characters from `a` to `t` where the i-th ($1 \le i \le n$) character is the label of vertex i in the tree.

Output

Print n integers in a single line, the i-th of which is the number of palindromic paths passing through vertex i in the tree.

Examples

input
5 1 2 2 3 3 4 3 5 abcbb
output
1 3 4 3 3

input
7 6 2 4 3 3 7 5 2 7 2 1 4 afefdfs
output
1 4 1 1 2 4 2

Note

In the first sample case, the following paths are palindromic:

2 - 3 - 4

2 - 3 - 5

4 - 3 - 5

Additionally, all paths containing only one vertex are palindromic. Listed below are a few paths in the first sample that are not palindromic:

1 - 2 - 3

1 - 2 - 3 - 4

1 - 2 - 3 - 5

F. Substrings in a String

time limit per test: 6 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Given a string s, process q queries, each having one of the following forms:

- $1\ i\ c$ — Change the i-th character in the string to c.
- $2\ l\ r\ y$ — Consider the substring of s starting at position l and ending at position r. Output the number of times y occurs as a substring in it.

Input

The first line of the input contains the string s ($1 \le |s| \le 10^5$) of lowercase English letters.

The second line contains an integer q ($1 \le q \le 10^5$) — the number of queries to process.

The next q lines describe the queries and may have one of the following forms:

- $1\ i\ c$ ($1 \le i \le |s|$)
- $2\ l\ r\ y$ ($1 \le l \le r \le |s|$)

c is a lowercase English letter and y is a non-empty string consisting of only lowercase English letters.

The sum of $|y|$ over all queries of second type is at most 10^5.

It is guaranteed that there is at least one query of second type.

All strings are 1-indexed.

$|s|$ is the length of the string s.

Output

For each query of type 2, output the required answer in a separate line.

Examples

input
ababababa 3 2 1 7 aba 1 5 c 2 1 7 aba

output
3 1

input
abcdcbc 5 2 1 7 bc 1 4 b 2 4 7 bc 1 2 a 2 1 4 aa

output
2 2 1

Note

Consider the first sample case. Initially, the string `aba` occurs 3 times in the range $[1, 7]$. Note that two occurrences may overlap.

After the update, the string becomes `ababcbaba` and now `aba` occurs only once in the range $[1, 7]$.

G. Sum the Fibonacci

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

You are given an array s of n non-negative integers.

A 5-tuple of integers (a, b, c, d, e) is said to be valid if it satisfies the following conditions:

- $1 \le a, b, c, d, e \le n$
- $(s_a \mid s_b)$ & s_c & $(s_d \wedge s_e) = 2^i$ for some integer i
- s_a & $s_b = 0$

Here, '|' is the bitwise OR, '&' is the bitwise AND and '^' is the bitwise XOR operation.

Find the sum of $f(s_a|s_b) * f(s_c) * f(s_d \wedge s_e)$ over all valid 5-tuples (a, b, c, d, e), where $f(i)$ is the i-th Fibonnaci number ($f(0) = 0, f(1) = 1, f(i) = f(i - 1) + f(i - 2)$).

Since answer can be is huge output it modulo $10^9 + 7$.

Input

The first line of input contains an integer n ($1 \le n \le 10^6$).

The second line of input contains n integers s_i ($0 \le s_i < 2^{17}$).

Output

Output the sum as described above, modulo $10^9 + 7$

Examples

input
2 1 2
output
32

input
3 7 4 1
output
3520

input
10 1 3 0 7 3 7 6 5 7 5
output
1235424

input
10 50 9 11 44 39 40 5 39 23 7
output
113860062

H. Ember and Storm's Tree Game

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Ember and Storm play a game. First, Ember picks a labelled tree T of n vertices, such that the degree of every vertex is at most d. Then, Storm picks two distinct vertices u and v in this tree and writes down the labels of the vertices in the path from u to v in a sequence $a_1, a_2 \ldots a_k$. Finally, Ember picks any index i ($1 \le i < k$) in the array. Now he performs one of the following two operations exactly once:

- flip the subrange $[i + 1, k]$ and add a_i to it. After this, the sequence becomes $a_1, \ldots a_i, a_k + a_i, a_{k-1} + a_i, \ldots a_{i+1} + a_i$
- negate the subrange $[i + 1, k]$ and add a_i to it. i.e., the array becomes $a_1, \ldots a_i, -a_{i+1} + a_i, -a_{i+2} + a_i, \ldots -a_k + a_i$

Ember wins if the array is monotonically increasing or decreasing after this. Otherwise Storm wins.

The game can be described by the tuple (T, u, v, i, op) where op is «flip» or «negate» depending on the action Ember chose in the last turn. Find the number of tuples that can occur if Ember and Storm play optimally. When they play optimally, if there are multiple moves by which they are guaranteed to win, then they may play any of the winning moves. Otherwise, if someone loses no matter what they play, then they may play any of the possible moves.

Report the answer modulo m.

Input

The input consists of a single line containing three integers n, d and m ($2 \le n \le 200$, $1 \le d < n$, $1 \le m \le 2 \cdot 10^9$).

Output

Print a single number — the number of possible tuples if Ember and Storm play as described, modulo m.

Examples

input
2 1 1000000007
output
4

input
3 1 250
output
0

input
3 2 100
output
36

Note

In the first sample case, there is only one possible tree. There are two possible paths, 1 to 2 and 2 to 1. For both paths, i can only be 1, and op can take both possibilities. Therefore, the answer is 4.

In the second sample, there are no possible trees.

In the third sample, there are three possible trees.