Codeforces Round #232 (Div. 1)

A. On Number of Decompositions into Multipliers

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

You are given an integer m as a product of integers $a_1, a_2, \ldots a_n$ $m = \prod_{i=1}^{n} a_i$. Your task is to find the number of distinct decompositions of number m into the product of n ordered positive integers.

Decomposition into n products, given in the input, must also be considered in the answer. As the answer can be very large, print it modulo 1000000007 $(10^9 + 7)$.

Input

The first line contains positive integer n ($1 \le n \le 500$). The second line contains space-separated integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$).

Output

In a single line print a single number k — the number of distinct decompositions of number m into n ordered multipliers modulo 1000000007 $(10^9 + 7)$.

Examples

input
1 15
output
1

input
3 1 1 2
output
3

input
2 5 7
output
4

Note

In the second sample, the get a decomposition of number 2, you need any one number out of three to equal 2, and the rest to equal 1.

In the third sample, the possible ways of decomposing into ordered multipliers are [7,5], [5,7], [1,35], [35,1].

A decomposition of positive integer m into n ordered multipliers is a cortege of positive integers $b = \{b_1, b_2, \ldots b_n\}$ such that $m = \prod_{i=1}^{n} b_i$. Two decompositions b and c are considered different, if there exists index i such that $b_i \neq c_i$.

B. On Sum of Fractions

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

Let's assume that

- $v(n)$ is the largest prime number, that does not exceed n;
- $u(n)$ is the smallest prime number strictly greater than n.

Find $\sum_{i=2}^{t} \frac{1}{v(i) \cdot u(i)}$.

Input

The first line contains integer t ($1 \le t \le 500$) — the number of testscases.

Each of the following t lines of the input contains integer n ($2 \le n \le 10^9$).

Output

Print t lines: the i-th of them must contain the answer to the i-th test as an irreducible fraction "p/q", where p, q are integers, $q > 0$.

Examples

input
2 2 3
output
1/6 7/30

C. On Changing Tree

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

You are given a rooted tree consisting of n vertices numbered from 1 to n. The root of the tree is a vertex number 1.

Initially all vertices contain number 0. Then come q queries, each query has one of the two types:

- The format of the query: $1\ v\ x\ k$. In response to the query, you need to add to the number at vertex v number x; to the numbers at the **descendants** of vertex v at distance 1, add $x - k$; and so on, to the numbers written in the descendants of vertex v at distance i, you need to add $x - (i \cdot k)$. The distance between two vertices is the number of edges in the shortest path between these vertices.
- The format of the query: $2\ v$. In reply to the query you should print the number written in vertex v modulo 1000000007 $(10^9 + 7)$.

Process the queries given in the input.

Input

The first line contains integer n ($1 \le n \le 3 \cdot 10^5$) — the number of vertices in the tree. The second line contains $n - 1$ integers $p_2, p_3, \ldots p_n$ ($1 \le p_i < i$), where p_i is the number of the vertex that is the parent of vertex i in the tree.

The third line contains integer q ($1 \le q \le 3 \cdot 10^5$) — the number of queries. Next q lines contain the queries, one per line. The first number in the line is $type$. It represents the type of the query. If $type = 1$, then next follow space-separated integers v, x, k ($1 \le v \le n$; $0 \le x < 10^9 + 7$; $0 \le k < 10^9 + 7$). If $type = 2$, then next follows integer v ($1 \le v \le n$) — the vertex where you need to find the value of the number.

Output

For each query of the second type print on a single line the number written in the vertex from the query. Print the number modulo 1000000007 $(10^9 + 7)$.

Examples

input
3 1 1 3 1 1 2 1 2 1 2 2
output
2 1

Note

You can read about a rooted tree here: `http://en.wikipedia.org/wiki/Tree_(graph_theory)`.

D. On Sum of Number of Inversions in Permutations

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

You are given a permutation p. Calculate the total number of inversions in all permutations that lexicographically do not exceed the given one.

As this number can be very large, print it modulo 1000000007 $(10^9 + 7)$.

Input

The first line contains a single integer n ($1 \le n \le 10^6$) — the length of the permutation. The second line contains n distinct integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le n$).

Output

Print a single number — the answer to the problem modulo 1000000007 $(10^9 + 7)$.

Examples

input
2 2 1
output
1

input
3 2 1 3
output
2

Note

Permutation p of length n is the sequence that consists of n distinct integers, each of them is from 1 to n.

An inversion of permutation $p_1, p_2, \ldots, p_n$ is a pair of indexes (i, j), such that $i < j$ and $p_i > p_j$.

Permutation a do not exceed permutation b lexicographically, if either $a = b$ or there exists such number i, for which the following logical condition fulfills: $(\forall j < i,\ a_j = b_j)$ AND $(a_i < b_i)$.

E. On Iteration of One Well-Known Function

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

Of course, many of you can calculate $\varphi(n)$ — the number of positive integers that are less than or equal to n, that are coprime with n. But what if we need to calculate $\varphi(\varphi(\ldots\varphi(n)))$, where function φ is taken k times and n is given in the canonical decomposition into prime factors?

You are given n and k, calculate the value of $\varphi(\varphi(\ldots\varphi(n)))$. Print the result in the canonical decomposition into prime factors.

Input

The first line contains integer m ($1 \le m \le 10^5$) — the number of distinct prime divisors in the canonical representaion of n.

Each of the next m lines contains a pair of space-separated integers p_i, a_i ($2 \le p_i \le 10^6$; $1 \le a_i \le 10^{17}$) — another prime divisor of number n and its power in the canonical representation. The sum of all a_i doesn't exceed 10^{17}. Prime divisors in the input follow in the strictly increasing order.

The last line contains integer k ($1 \le k \le 10^{18}$).

Output

In the first line, print integer w — the number of distinct prime divisors of number $\varphi(\varphi(\ldots\varphi(n)))$, where function φ is taken k times.

Each of the next w lines must contain two space-separated integers q_i, b_i ($b_i \ge 1$) — another prime divisor and its power in the canonical representaion of the result. Numbers q_i must go in the strictly increasing order.

Examples

input
1 7 1 1
output
2 2 1 3 1

input
1 7 1 2
output
1 2 1

input
1 2 100000000000000000 10000000000000000
output
1 2 900000000000000000

Note

You can read about canonical representation of a positive integer here: `http://en.wikipedia.org/wiki/Fundamental_theorem_of_arithmetic`.

You can read about function $\varphi(n)$ here: `http://en.wikipedia.org/wiki/Euler's_totient_function`.