Codeforces Round #285 (Div. 1)

A. Misha and Forest

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

Let's define a forest as a non-directed acyclic graph (also without loops and parallel edges). One day Misha played with the forest consisting of n vertices. For each vertex v from 0 to $n - 1$ he wrote down two integers, $degree_v$ and s_v, were the first integer is the number of vertices adjacent to vertex v, and the second integer is the XOR sum of the numbers of vertices adjacent to v (if there were no adjacent vertices, he wrote down 0).

Next day Misha couldn't remember what graph he initially had. Misha has values $degree_v$ and s_v left, though. Help him find the number of edges and the edges of the initial graph. It is guaranteed that there exists a forest that corresponds to the numbers written by Misha.

Input

The first line contains integer n ($1 \le n \le 2^{16}$), the number of vertices in the graph.

The i-th of the next lines contains numbers $degree_i$ and s_i ($0 \le degree_i \le n - 1$, $0 \le s_i < 2^{16}$), separated by a space.

Output

In the first line print number m, the number of edges of the graph.

Next print m lines, each containing two distinct numbers, a and b ($0 \le a \le n - 1$, $0 \le b \le n - 1$), corresponding to edge (a, b).

Edges can be printed in any order; vertices of the edge can also be printed in any order.

Examples

input
3 2 3 1 0 1 0
output
2 1 0 2 0

input
2 1 1 1 0
output
1 0 1

Note

The XOR sum of numbers is the result of bitwise adding numbers modulo 2. This operation exists in many modern programming languages. For example, in languages C++, Java and Python it is represented as "^", and in Pascal — as "xor".

B. Misha and Permutations Summation

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

Let's define the sum of two permutations p and q of numbers $0, 1, ..., (n - 1)$ as permutation $Perm((Ord(p) + Ord(q)) \bmod n!)$, where $Perm(x)$ is the x-th lexicographically permutation of numbers $0, 1, ..., (n - 1)$ (counting from zero), and $Ord(p)$ is the number of permutation p in the lexicographical order.

For example, $Perm(0) = (0, 1, ..., n - 2, n - 1)$, $Perm(n! - 1) = (n - 1, n - 2, ..., 1, 0)$

Misha has two permutations, p and q. Your task is to find their sum.

Permutation $a = (a_0, a_1, ..., a_{n-1})$ is called to be lexicographically smaller than permutation $b = (b_0, b_1, ..., b_{n-1})$, if for some k following conditions hold: $a_0 = b_0, a_1 = b_1, ..., a_{k-1} = b_{k-1}, a_k < b_k$.

Input

The first line contains an integer n ($1 \le n \le 200\,000$).

The second line contains n distinct integers from 0 to $n - 1$, separated by a space, forming permutation p.

The third line contains n distinct integers from 0 to $n - 1$, separated by spaces, forming permutation q.

Output

Print n distinct integers from 0 to $n - 1$, forming the sum of the given permutations. Separate the numbers by spaces.

Examples

input
2 0 1 0 1
output
0 1

input
2 0 1 1 0
output
1 0

input
3 1 2 0 2 1 0
output
1 0 2

Note

Permutations of numbers from 0 to 1 in the lexicographical order: $(0, 1), (1, 0)$.

In the first sample $Ord(p) = 0$ and $Ord(q) = 0$, so the answer is $Perm((0 + 0) \bmod 2) = Perm(0) = (0, 1)$.

In the second sample $Ord(p) = 0$ and $Ord(q) = 1$, so the answer is $Perm((0 + 1) \bmod 2) = Perm(1) = (1, 0)$.

Permutations of numbers from 0 to 2 in the lexicographical order: $(0, 1, 2), (0, 2, 1), (1, 0, 2), (1, 2, 0), (2, 0, 1), (2, 1, 0)$.

In the third sample $Ord(p) = 3$ and $Ord(q) = 5$, so the answer is $Perm((3 + 5) \bmod 6) = Perm(2) = (1, 0, 2)$.

C. Misha and Palindrome Degree

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

Misha has an array of n integers indexed by integers from 1 to n. Let's define *palindrome degree* of array a as the number of such index pairs $(l, r)(1 \le l \le r \le n)$, that the elements from the l-th to the r-th one inclusive can be rearranged in such a way that the **whole** array will be a palindrome. In other words, pair (l, r) should meet the condition that after some rearranging of numbers on positions from l to r, inclusive (it is allowed not to rearrange the numbers at all), for any $1 \le i \le n$ following condition holds: $a[i] = a[n - i + 1]$.

Your task is to find the *palindrome degree* of Misha's array.

Input

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

The second line contains n positive integers $a[i]$ ($1 \le a[i] \le n$), separated by spaces — the elements of Misha's array.

Output

In a single line print the answer to the problem.

Examples

input
3 2 2 2
output
6

input
6 3 6 5 3 3 5
output
0

input
5 5 5 2 5 2
output
4

Note

In the first sample test any possible pair (l, r) meets the condition.

In the third sample test following pairs $(1, 3), (1, 4), (1, 5), (2, 5)$ meet the condition.

D. Misha and XOR

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

After Misha's birthday he had many large numbers left, scattered across the room. Now it's time to clean up and Misha needs to put them in a basket. He ordered this task to his pet robot that agreed to complete the task at certain conditions. Before the robot puts a number x to the basket, Misha should answer the question: is it possible to choose one or multiple numbers that already are in the basket, such that their XOR sum equals x?

If the answer is positive, you also need to give the indexes of these numbers. If there are multiple options of choosing numbers, you are allowed to choose any correct option. After Misha's answer the robot puts the number to the basket.

Initially the basket is empty. Each integer you put in the basket takes some number. The first integer you put into the basket take number 0, the second integer takes number 1 and so on.

Misha needs to clean up the place as soon as possible but unfortunately, he isn't that good at mathematics. He asks you to help him.

Input

The first line contains number m ($1 \le m \le 2000$), showing how many numbers are scattered around the room.

The next m lines contain the numbers in the order in which the robot puts them in the basket. Each number is a positive integer strictly less than 10^{600} that doesn't contain leading zeroes.

Output

For each number either print a 0 on the corresponding line, if the number cannot be represented as a XOR sum of numbers that are in the basket, or print integer k showing how many numbers are in the representation and the indexes of these numbers. Separate the numbers by spaces. Each number can occur in the representation at most once.

Examples

input
7 7 6 5 4 3 2 1
output
0 0 0 3 0 1 2 2 1 2 2 0 2 2 0 1

input
2 5 5
output
0 1 0

Note

The XOR sum of numbers is the result of bitwise sum of numbers modulo 2.

E. Misha and LCP on Tree

time limit per test: 8 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Misha has a tree with characters written on the vertices. He can choose two vertices s and t of this tree and write down characters of vertices lying on a path from s to t. We'll say that such string corresponds to pair (s, t).

Misha has m queries of type: you are given 4 vertices a, b, c, d; you need to find the largest common prefix of the strings that correspond to pairs (a, b) and (c, d). Your task is to help him.

Input

The first line contains integer n ($1 \le n \le 300\,000$) — the number of vertices in the tree.

Next follows a line consisting of n small English letters. The i-th character of the string corresponds to the character written on the i-th vertex.

Next $n - 1$ lines contain information about edges. An edge is defined by a pair of integers u, v ($1 \le u, v \le n$, $u \ne v$), separated by spaces.

The next line contains integer m ($1 \le m \le 1\,000\,000$) — the number of queries.

Next m lines contain information about queries. A query is defined by four integers a, b, c, d ($1 \le a, b, c, d \le n$), separated by spaces.

Output

For each query print the length of the largest common prefix on a separate line.

Examples

input
6 bbbabb 2 1 3 2 4 3 5 2 6 5 6 2 5 3 1 1 5 2 3 5 6 5 6 6 3 4 1 6 2 3 4 2 2 4 5

output
2 2 2 0 1 0