Parallel Processing (Hard)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

This is the hard version of the problem. The only difference between two versions is the constraint of N.

You are given a mysterious monoid $(M, \oplus)$ and 4 CPUs to compute it.

Compute the cumulative $\oplus$ of a sequence $A = (A_1, A_2, \ldots, A_N)$ of M in parallel using 4 CPUs, minimizing the number of operations.

Statement

You are given an integer N. Write a program in a custom language to do the following and minimize the number of instructions in your program.

Specification

This program can handle 2004 variables $A[1], A[2], \ldots, A[2000], C_1, C_2, C_3, C_4$. Each variable can hold a sequence of integers, and $A[i]$ $(1 \leq i \leq 2000)$ is initialized to $A[i] = (i)$. (Here (i) denotes an integer sequence consisting of one i.)

At the end of the execution, the following condition must be satisfied:

- For each of $i = 1, 2, ..., N$, $A[i] = (1, 2, ..., i)$ holds.

Format

The first line of the program contains an integer L representing the number of instructions in the program.

The L instructions are written in 4 lines per instruction from the 2nd to the $(4L + 1)$-th lines, and are executed sequentially from top to bottom.

Each instruction is written as 12 integers $c_1, a_1, b_1, c_2, a_2, b_2, c_3, a_3, b_3, c_4, a_4, b_4$, where each integer must be between 1 and 2000 (inclusive).

For each instruction, the following operations are performed in order:

1. Assigns concat$(A[a_1], A[b_1])$ to C_1.
2. Assigns concat$(A[a_2], A[b_2])$ to C_2.
3. Assigns concat$(A[a_3], A[b_3])$ to C_3.
4. Assigns concat$(A[a_4], A[b_4])$ to C_4.
5. Assigns C_1 to $A[c_1]$.
6. Assigns C_2 to $A[c_2]$.
7. Assigns C_3 to $A[c_3]$.
8. Assigns C_4 to $A[c_4]$.

Here, concat(x, y) denotes the sequence obtained by concatenating the sequences x and y in that order.

Input

The input is given in the following format:

N

- All values in the input are integers.
- $17 \leq N \leq 1000$

Output

Let L be the minimum number of instructions. Output in the following format:

L
op_1
op_2
$\vdots$
op_L

op_i $(1 \leq i \leq L)$ represents the i-th operation and should be output in the following format:

c_1 a_1 b_1
c_2 a_2 b_2
c_3 a_3 b_3
c_4 a_4 b_4

Here, each integer must be between 1 and 2000.