Input File: `hanoi.in` **100 Points**
Output File: `hanoi.out` **Time Limit:** 3 s
Source Code: `hanoi.pas/.c/.cpp` **Memory Limit:** 16 MB

Towers of Hanoi

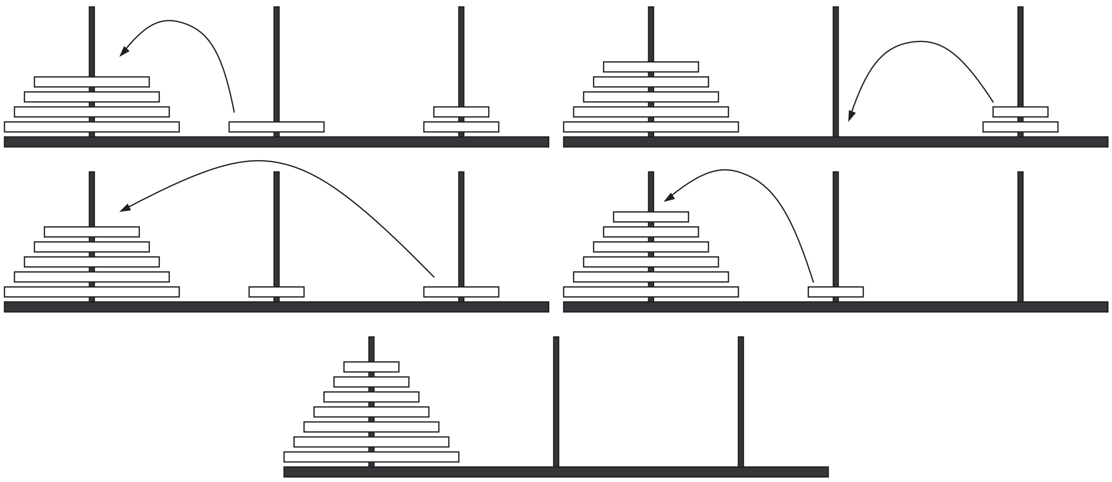
Surely you have already come across the *Towers of Hanoi* problem: Wooden disks of different sizes are stacked on three pegs, and initially, all disks are stacked on the same peg sorted by size, with the largest disk at the bottom. The objective is to transfer the entire tower to one of the other pegs, moving only one disk at a time and never putting a larger disk onto a smaller one.

According to an old myth, the monks at an ancient Tibetian monastery have been trying to solve an especially large instance of this problem with 47 disks for thousands of years. Since this requires at least $2^{47} - 1$ moves and the monks started out without a strategy, they messed it all up while still following the rules. Now they would like to have the disks stacked up neatly on any arbitrary peg using the minimum number of moves. But they all took a vow which forbids them to move the disks contrary to the rules. They want to know on which peg they should best stack the disks, and the minimum number of moves needed.

Write a program that solves this problem for the monks. Your program should also be able to handle any number N ($0 < N \leq 100\,000$) of disks. The numbers involved in the computation can become quite large. Because of that, the monks are only interested in the number of moves modulo $1\,000\,000$.

Example

The following example can be solved in four moves.



Input

The first line of the input file `hanoi.in` consists of the number N of disks. The second line consists of three integers s_1, s_2, s_3 with $0 \leq s_1, s_2, s_3 \leq N$ and $s_1 + s_2 + s_3 = N$, the number of disks on each of the three pegs. Lines three to five each contain the sizes of the disks for one peg. More precisely: The $(i+2)$-th line of the input file consists of integer numbers $m_{i,1} \ldots m_{i,s_i}$ with $1 \leq m_{i,j} \leq N$, the sizes of the disks on peg i. The disks are given from bottom to top, thus $m_{i,1} > m_{i,2} > \ldots > m_{i,s_i}$. Note that an empty stack is given by an empty line. The set of N disks have different sizes. All numbers are separated by a single space.

Output

The first line of the output file `hanoi.out` consists of the number $d \in \{1, 2, 3\}$ of the peg onto which the disks can be stacked using the minimum number of moves. The second line consists of the number M of required moves modulo 1 000 000.

Example

hanoi.in	hanoi.out
7 2 1 4 2 1 3 7 6 5 4	3 4

Testdata

Your program will be tested with 20 different inputs during grading. The following table consists of the first lines of each input file, i.e. the number N of disks for each input.

Test Nr.	1	2	3	4	5	6	7	8	9	10	11	12
N	5	10	15	20	50	100	150	200	1000	2000	3000	4000

Test Nr.	13	14	15	16	17	18	19	20
N	30000	40000	50000	60000	70000	80000	90000	100000