

Problem J. Horses

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

Big Horse is the God of Horses. He has n different kinds of horses. Since his eyes are not so good, he cannot distinguish between the horses of the same kind.

Now he wants to arrange m horses in a queue. But the horses are so active that they may change their positions at will. However, Big Horse noticed that only two adjacent horses may swap, and this may happen only if the two **kinds** of horses are friends. Since the horses can swap their positions at any time, Big Horse considers two queues equivalent if and only if one can be reached from the other one by a finite number of swaps.

Now Big Horse has a queue $a = (a_1, \dots, a_m)$ of horses. He wants to add some other horses to the left of the queue. However, Big Horse cannot tell left from right. So he wants to add a queue $b = (b_1, \dots, b_k)$ such that b commutes with a : in other words, $b_1, \dots, b_k, a_1, \dots, a_m$ is equivalent to $a_1, \dots, a_m, b_1, \dots, b_k$.

However, the number of such b may be too large. Big Horse only cares about the “minimal” such queues b . Specifically, he is interested in b such that:

- b commutes with a ,
- b is not equivalent to $c_1, \dots, c_{k'}, d_1, \dots, d_{k''}$ such that c commutes with a and d commutes with a ,
- b is lexicographically the least among all queues equivalent to it.

He found out that there are at most n minimal queues. He asks you to help him find them.

Input

In the first line, there is an integer n ($1 \leq n \leq 200$).

Then follow $n - 1$ lines. In the i -th of these lines, there are $n - i$ integers. The j -th integer in the i -th of these lines is 1 if a horse of kind i can swap with a horse of kind $i + j$, and 0 otherwise.

The next line contains an integer m ($1 \leq m \leq 300\,000$).

The last line contains m integers a_1, \dots, a_m : the kinds of horses in the queue ($1 \leq a_i \leq n$).

Output

Output the minimal queues, one per line. Since a queue may be too long, when the minimal queue is b , you only need to print the hash value $b_1 + b_2 \cdot (n + 1) + \dots + b_k \cdot (n + 1)^{(k-1)}$ modulo 998 244 353.

You should output the minimal queues in lexicographical order (order them before hashing).

Example

standard input	standard output
3	1
1 1	14
0	
5	
2 1 3 2 3	

Note

The two minimal queues in the example are (1) and (2, 3).