Problem C. Catalan Numbers

Input file:	`catalan.in`
Output file:	`catalan.out`
Time limit:	2 seconds
Memory limit:	512 megabytes

Andrew likes Catalan numbers. Also Andrew likes to joke.

He is an experienced problem setter and prepares lots of contests for training camps. Each contest he prepares a problem that has one integer as input, one integer as output, and answers for 0, 1, 2, 3, 4 and 5 are, respectively, 1, 1, 2, 5, 14 and 42. However, answers for greater inputs don't coincide with corresponding Catalan numbers.

Andrew has already prepared so many contests, that he is short of good problems with such property. So he decided to automate the process of creating such problems. As a good pool of possible problems he considers problems of counting words of specific length in deterministic finite automata. Andrew has chosen k — the desired answer for the input 6, and wants to find deterministic finite automaton that accepts 1, 1, 2, 5, 14, 42, k words of length 0, 1, 2, 3, 4, 5, 6, respectively, that has minimal number of states.

Recall, that the deterministic finite automaton (DFA) is an ordered set $\langle \Sigma, U, S, T, \varphi \rangle$ where Σ is the finite set called *input alphabet*, U is the finite set of *states*, $S \in U$ is the *initial state*, $T \subset U$ is the set of *terminal states* and $\varphi : U \times \Sigma \to U \cup \{\varnothing\}$ is the *transition function*.

The input of the automaton is the string α over Σ. Initially the automaton is in state s. Each step the automaton reads the first character c of the input string and changes its state to $\varphi(u, c)$ where u is the current state. If $\varphi(u, c) = \varnothing$ the automaton immediately rejects α. Then the first character of the input string is removed and the step repeats. If after its input string is empty the automaton is in the terminal state, it accepts the initial string α, in the other case it rejects it.

You are given an integer k. Construct deterministic finite automaton with alphabet of size at most 20, such that it has minimal number of states, and the number of words of lengths from 1 to 6 accepted by the automaton is given by the following table. The number of longer accepted words can be arbitrary.

length	**number of accepted words**
0	1
1	1
2	2
3	5
4	14
5	42
6	k

Input

The input file contains multiple test cases.

Each test case contains a single integer k on a line by itself ($120 \le k \le 140$).

Input is followed by a line with $n = 0$.

Output

For each test case print the requested deterministic automaton in the following format. The first line must contain two integers: n — the number of states and s — the size of the alphabet ($1 \le s \le 20$). Note that you must minimize n, but you don't have to minimize s.

Let the states be numbered from 1 to n, the starting state be 1, let the characters of the alphabet be numbered from 1 to s.

The second line must contain n integers and specify whether corresponding states are terminal: i-th of them must be 1 if the i-th state is terminal, or 0 if the i-th state is not terminal.

The following n lines must contain s integers each: the j-th of the i-th line must be the value of $\varphi(i, j)$ — the state where the transition from the i-th state by the j-th character goes, or 0 if $\varphi(i, j) = \varnothing$.

Examples

catalan.in	catalan.out
131 0	3 4 1 0 0 1 2 0 0 1 2 2 3 2 3 3 0

In the given example if $\Sigma = \{a, b, c, d\}$ the five words of length 3 accepted by the automaton are "aaa", "aba", "baa", "bba" and "bca".