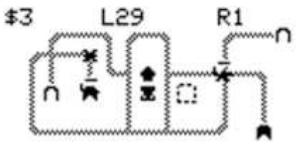
Problem G. Game Design

Tower defence games are the best games to kill time. In this kind of games, the player needs to build towers at certain candidate locations to defend a base building from monsters. The difficulty lies in that the budget for building towers is usually quite tight, and the player often needs to build towers at tricky locations in order to save resources while still being able to kill all the monsters.



Screenshot of a simple 1-bit tower defense game for Arduboy. Public Domain.

Let's consider a simplified version of the game: the map can be viewed as a rooted tree with **at least two** nodes, where the root is the base that the player needs to defend from monsters. Monsters will appear at each leaf node. When a monster appears, it goes up to the root node following the edges of the tree to attack the base. The player can build towers at any node of the tree including the root node. In this simplified game, we assume the towers are powerful enough that any monsters entering a node where a tower is built will be killed. Naturally, building towers at different nodes cost different amounts of resources and you need to defend the base (kill all the monsters) at the lowest possible cost.

Given the map, it's easy to calculate the optimal solution. However, this time you are not playing, but developing a tower defence game as your project for a course at your university. You are required to design a map (you need to give the structure of the tree and the cost of building towers at each node) that has exactly K optimal solutions to defend the base. A solution is optimal if and only if it has the lowest cost among all possible ways to build towers that ensure all monsters are killed. Two solutions are considered different if and only if there is at least one node such that in one solution a tower is built on this node, and in the other solution no tower is built on it.

Input

The input contains only a single case.

The only line of the input contains a single integer K ($1 \le K \le 10^9$), the number of optimal solutions that your map needs to have.

Output

Let's denote the number of nodes of the tree you construct as N, and let's label the nodes from 1 to N where node 1 is the root. Let c_i ($1 \le i \le N$) be the cost of building a tower at node i.

The output should contain 3 lines. The first line should contain a single integer N. The second line should contain $N-1$ space-separated integers, where the i-th ($1 \le i < N$) integer denotes the label of the parent node of node $i+1$. The third line should contain N space-separated integers, where the i-th ($1 \le i \le N$) integer denotes c_i.

Your solution must satisfy $2 \le N \le 10^5$ and $\forall 1 \le i \le N : \ 1 \le c_i \le 10^9$. It is guaranteed that at least one solution exists. If there are multiple valid solutions, any of them will be accepted.

Example

standard input	standard output
2	2 1 1 1

Note

The sample output is only one of the possible solutions for the sample case. There are other valid solutions.