

Problem D. Output Limit Exceeded

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 256 mebibytes

We all know that $\binom{n}{k} = \frac{n \cdot (n-1) \cdot \dots \cdot (n-k+2) \cdot (n-k+1)}{1 \cdot 2 \cdot \dots \cdot (k-1) \cdot k}$ is an integer number for any $0 \leq k \leq n$. But it would be nice if we could prove it by providing a matching between factors in numerator and denominator, wouldn't it?

Let's build a bipartite graph with k vertices in each part. The i -th vertex in the left part corresponds to factor $(n+1-i)$ from numerator and j -th vertex in the right part corresponds to factor j from denominator. There is an edge $i-j$ if and only if j divides $(n+1-i)$. The number k is *provable* for n if there is a perfect matching in this bipartite graph.

Given n , check if k is provable for each k satisfying $0 \leq k \leq n$.

Input

The only line contains one integer n ($0 \leq n \leq 10^{18}$).

Output

Print string of length $(n+1)$ consisting of '0' and '1', $(k+1)$ -th character should be '1' if and only if k is provable for n .

What, you think this will get Output Limit Exceeded? Hmmmm... Okay. Let's compress the string.

Let $s_0 = "0"$ and $s_1 = "1"$. You can define s_2, s_3, \dots, s_t . String s_i should be a concatenation of several earlier defined strings. Formally, $\forall i (2 \leq i \leq t) : s_i = s_{j_1} + s_{j_2} + \dots + s_{j_{k_i}}$, here $1 \leq k_i, \forall r (1 \leq r \leq k_i) : j_r < i$. String s_t should be the answer to the problem.

In the first line print one integer t ($2 \leq t \leq 500$).

In the next $t-1$ lines print the descriptions of s_i . Each description should have a form $k_i j_1 j_2 \dots j_{k_i}$, with $1 \leq k_i$ and $0 \leq j_r < i$.

Total length of all descriptions should not exceed 10 000: $\sum_{i=2}^t k_i \leq 10\,000$.

We can show that for all valid tests there exists a way to construct the answer string abiding all the limitations. If there are several possible ways to do so, print any one of them. Note that you don't have to minimize t or total length of all descriptions.

Examples

standard input	standard output
1	2 2 1 1
0	2 1 1
7	4 2 1 1 4 1 2 0 0 3 3 1 2

Note

In the third sample: $s_2 = s_1 + s_1 = "1" + "1" = "11"$, $s_3 = s_1 + s_2 + s_0 + s_0 = "1" + "11" + "0" + "0" = "11100"$, $s_4 = s_3 + s_1 + s_2 = "11100" + "1" + "11" = "11100111"$.