Huge Pascal's Triangle

During programming courses, either high-school or university level, it is often taught that one of the key elements of an algorithm is the "input", among other things. However, an algorithm that takes no input still deserves to be called an "algorithm". You are now to demonstrate to yourself what such a program might look like.

The task is to generate Pascal's triangle. It is a well-known concept in combinatorics. In addition to that, the rules rooted deep within this triangle come up often seemingly out of nowhere in many other areas. The truth is, that once they do appear, it feels like magic. See for yourself. A basic formula for generating the entry at n-th row and k-th column of Pascal's triangle is

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

The special case is the topmost row, for which we have

$$\binom{0}{0} = 1.$$

We can thus establish the relationship to produce each entry as

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k},$$

for any non-negative integer n and any integer $0 \leq k \leq n$.

In order to sufficiently experience the beauty, the task is to keep generating the triangle as long as the current row does not contain any value either equal or exceeding the limit $L = 10^{60}$. The sample output could give you an idea of what the output could look like for $L = 10^2$.

Output sample

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
.
.
.
```