

Problem G

Cut the Silver Bar

Source file name: `silver.c`, `silver.cpp` or `silver.java`

A creditor wants a daily payment during n days from a poor miner in debt. Since the miner can not pay his daily obligation, he has negotiated with the creditor an alternative way, convenient for both parties, to pay his debt: the miner will give an equivalent of a 1μ ($= 0.001mm$) long piece of a silver bar as a guarantee towards the debt. The silver bar owned by the poor miner is initially $n\mu$ units long.

By the end of n days the miner would not have any more silver to give and the creditor would have received an amount of silver equivalent to that of the silver bar initially owned by the miner. By then, the miner expected to have enough money to pay the debt at the next day so that he would have back all his silver.

With this negotiation in mind, the miner has realized that it was not necessary to cut exactly 1μ silver piece from the bar everyday. For instance, at the third day he could give the creditor a 3μ silver piece, taking back the equivalent of a 2μ silver piece which the creditor should already have.

Since cutting the bar was rather laborious and time consuming, the miner wanted to minimize the number of cuts he needed to perform on his silver bar in order to make the daily silver deposits during the n days. Could you help him?

Input

Input consists of several cases, each one defined by a line containing a positive integer number n (representing the length in micras of the silver bar and the number of days of the amortization period). You may assume that $0 < n < 20000$.

The end of the input is recognized by a line with 0.

The input must be read from the file `silver.in`.

Output

For each given case, output one line with a single number: the minimum number of cuts in which to cut a silver bar of length $n\mu$ to guarantee the debt during n days.

The output must be written to standard output.

Sample input	Output for the sample input
1	0
5	2
3	1
0	