Lemon tree **P96448_en**

Quinzè Concurs de Programació de la UPC - Semifinal (2017-06-29)

Professor Oak kas a big lemon tree that produces nice lemons. However, every time that a friend visits him, he or she always asks for "a few" of them. As a result, Prof. Oak can barely enjoy his own lemons.

Tired of this situation, Prof. Oak has decided to impose a rule: When someone asks for lemons, he or she can only take a Fibonacci number of them. (Remember the definition: $F_0 = 0$, $F_1 = 1$, $F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$.) This way, perhaps some lemons can be saved...

The lemon tree has currently ℓ lemons, and a group of m mathematicians visits Prof. Oak. They will collaborate to take the maximum total number of lemons. (They will share all them later.) For instance, if $\ell = 46$ and $m = 2$, then each mathematician can ask for $F_8 = 21$ lemons, only leaving 4 lemons to Prof. Oak. It is easy to see that there is no combination for 2 mathematicians with a sum larger than 42 but not larger than 46.

Given ℓ and m, how many lemons will Prof. Oak enjoy?

Input

Input consists of several cases, each with ℓ and m. Assume $0 \leq \ell \leq 10^{18}$ and $0 \leq m \leq 1000$.

Output

For every case, print the number of lemons left by the mathematicians.

Sample input

```
46 2
4 1000
679891637638612257 1
```

Sample output

```
4
0
259695496911122584
```

Problem information

Author : Salvador Roura
Generation : 2024-05-03 09:39:45