Fibonacci Matrix

The Fibonacci series appears in lots of unexpected places, such as Pascal's triangle, as shown in the next figure:

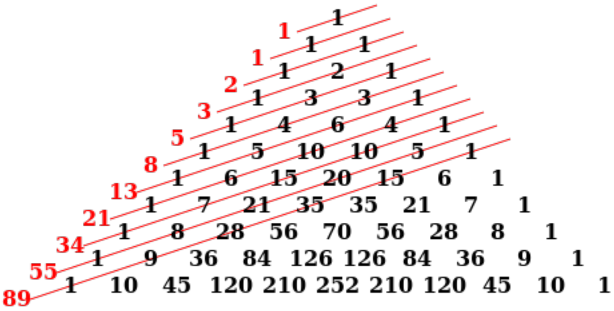

Figure 1: The Fibonacci series in Pascal's triangle[1]

Another way to obtain the numbers of the Fibonacci sequence is the equation:

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n = \begin{pmatrix} fib_{n-1} & fib_n \\ fib_n & fib_{n+1} \end{pmatrix} \quad \forall n \geq 1$$

Code a function that calculates the n^{th} term of the Fibonacci series using this idea.
Coding suggestions:

- Use a `struct Matrix` to store a $2x2$ matrix.
- `Matrix empty()` returns a matrix filled with $0's$
- `Matrix identity()` returns a matrix filled with $1's$ in the main diagonal
- `Matrix prod(Matrix m1, Matrix m2)` multiplies two matrices and returns a matrix with the result %46337

In addition to coding the solution, you have to specify the algorithm and calculate its complexity.

Input

Each test case is a single line with a non-negative value of $n \leq 2^{31} - 1$. The input ends when the input value is 0.

Output

The output of each test case is $fib_n \% 46337$.

Sample input

```
1
2
4
22
23
24
0
```

[1] https://en.wikipedia.org/wiki/Fibonacci_polynomials

Sample output

```
1
1
3
17711
28657
31
```

Notes

This exercise has been designed by Marco Antonio Gómez Martín. It must be understood in the context of the *Data Structures and Algorithms* course, FDI-UCM 2016/2017 (prof. Gonzalo Méndez). Therefore, the only valid solutions are those that use the concepts studied in this course. Additional remarks may be provided in class.