

Problem Statement

Your goal in this problem is to find the last digit of n^{th} Fibonacci number. Recall that Fibonacci numbers grow exponentially fast. For example,

$$F_{200} = 280\ 571\ 172\ 992\ 510\ 140\ 037\ 611\ 932\ 413\ 038\ 677\ 189\ 525 .$$

Therefore, a solution like

$$F_{[0]} \leftarrow 0$$

$$F_{[1]} \leftarrow 1$$

for i from 2 to n :

$$F_{[i]} \leftarrow F_{[i-1]} + F_{[i-2]}$$

print($F_{[n]} \bmod 10$)

will turn out to be *too slow*, because as i grows the i^{th} iteration of the loop computes the sum of longer and longer numbers. Also, for example, F_{1000} does not fit into the standard C++ int type. To overcome this difficulty, you may want to store in $F_{[i]}$ not the i^{th} Fibonacci number itself, but just its last digit (that is, $F_i \bmod 10$).

Computing the last digit of F_i is easy: it is just the last digit of the sum of the last digits of

F_{i-1} and F_{i-2} :

$$F_{[i]} \leftarrow (F_{[i-1]} + F_{[i-2]}) \bmod 10$$

This way, all $F_{[i]}$'s are just digits, so they fit perfectly into any standard integer type, and computing a sum

of $F_{[i-1]}$ and $F_{[i-2]}$ is performed very quickly.

Problem Description

Task. Given an integer n , find the last digit of the n^{th} Fibonacci number F_n (that is, $F_n \bmod 10$).

Input Format. The input consists of a single integer n .

Constraint: $10^5 \leq n \leq 10^8$.

Output Format: Output the last digit of F_n .

Feel free to use any methods or algorithms in achieving the task and also the use any python library is allowed. Happy Coding ☺