2 Last Digit of a Large Fibonacci Number

Problem Introduction

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($F[n]$ mod 10)
```

will turn out to be too slow, because as i grows the ith 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 ith Fibonacci number itself, but just its last digit (that is, F_i mod 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}:

```
$F[i] \leftarrow (F[i-1] + F[i-2])$ mod 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 nth Fibonacci number F_n (that is, F_n mod 10).

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

Constraints. $0 \leq n \leq 10^7$.

Output Format. Output the last digit of F_n.

Sample 1.

Input:

```
3
```

Output:

```
2
```

$F_3 = 2$.

Sample 2.

Input:

```
331
```

Output:

```
9
```

$F_{331} = 668\,996\,615\,388\,005\,031\,531\,000\,081\,241\,745\,415\,306\,766\,517\,246\,774\,551\,964\,595\,292\,186\,469$.