Problem Introduction

Your goal in this problem is to nd the last digit of n-th Fibonacci number. Recall that Fibonacci numbers grow exponentially fast. For example,
$F200$ = 280571172992510140037611932413038677189525. 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, $F1000$ does not t into the standard C++ int type. To overcome this di culty, you may want to store in $F[i]$ not the ith Fibonacci number itself, but just its last digit (that is, Fi mod 10). Computing the last digit of Fi is easy: it is just the last digit of the sum of the last digits of $Fi-1$ and $Fi-2$:

$F[i] \leftarrow (F[i-1] + F[i-2])$ mod 10

This way, all $F[i]$'s are just digits, so they t 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, Fn mod 10). Input Format. The input consists of a single integer n.
Constraints. $0 \le n \le 10^7$.
Output Format. Output the last digit of F_n.

Sample 1.

Input:
331
Output:
9
Explanation:
F_{331} =
668996615388005031531000081241745415306766517246774551964595292186469.

Sample 2.

Input:
327305
Output:
5
Explanation:
F_{327305} does not fit into one line of this pdf, but its last digit is equal to 5.