Systems and Biomedical Engineering Department **Biomedical Data Analytics**
Faculty of Engineering **Spring 2021**
Cairo University

Homework#1: Fibonacci

1.1 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. $1 \leq n \leq 10^7$
Output Format. Output the last digit of F_n.

Sample1.

Input:

```
331
```

Output:

```
9
```

❾ 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.

Sample2.

Input:

```
3
```

Output:

```
2
```

❾ $F_3 = 2$.

Sample3.

Input:

```
327305
```

Output:

```
5
```

❾ F_{327305} does not fit into one line of this pdf, but its last digit is equal to 5

Time limits (sec.):

C	**C++**	**Python**
1	**1**	**5**

Starter Code

There is a starter code file for Python3 “fibonacci_last_digit.py” and C++ “fibonacci_last_digit.cpp”.

Starter code file contains a naïve implementation for the problem. You should write a faster implementation. The starter code reads the input and outputs the result in the correct format.

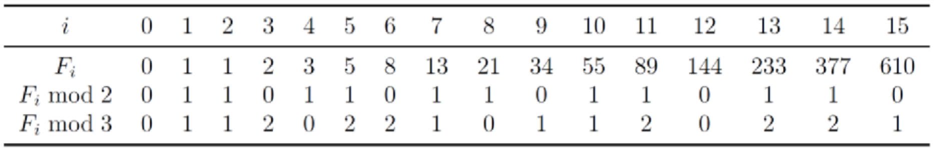
DONOT CHANGE THE PRINT STATEMENT OR OUTPUT/INPUT FORMAT

1.2 Modulo Fibonacci Number

Problem Introduction:

In this problem, your goal is to compute F_n *modulo m,* where n may be really huge: up to 10^{14}. For such values of n, an algorithm looping for n iterations will not fit into one second for sure. Therefore, we need to avoid such a loop.

To get an idea how to solve this problem without going through all F_i for i from 0 to n, look at what happens when m is small, m = 2 or m = 3.



i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
F_i	0	1	1	2	3	5	8	13	21	34	55	89	144	233	377	610
F_i mod 2	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0
F_i mod 3	0	1	1	2	0	2	2	1	0	1	1	2	0	2	2	1

Looking at the table we see that both these sequences are periodic! For $m = 2$, the period is 011 and has length 3, while for $m = 3$ the period is 01120221 and has length 8. Therefore, to compute, say, F_{2015} mod 3 we just need to find the remainder of 2015 when divided by 8. Since 2015 = 251 · 8 + 7, we conclude that F_{2015} mod 3 = F_7 mod 3 = 1.

This is true in general: for any integer $m \geq 2$, the sequence F_n mod m is periodic. The period always starts with 01 and is known as Pisano period.

Problem Description

Task. Given two integers n and m, output F_n mod m (that is, the remainder of F_n when divided by m).

Input Format. The input consists of two integers n and m given on the same line (separated by a space).

Constraints. $1 \leq n \leq 10^{14}$, $2 \leq m \leq 10^3$.

Output Format. Output F_n mod m

Sample1.

Input:

```
239 1000
```

Output:

```
161
```

❾ F_{239} mod 1000 = 39 679 027 332 006 820 581 608 740 953 902 289 877 834 488 152 161 (mod 1000) = 161.

Sample2.

Input:

```
2816213588 239
```

Output:

```
151
```

Time limits (sec.):

C	C++	Python
1	1	5

Starter Code

There is a starter code file for Python3 “fibonacci_huge.py” and C++ “fibonacci_huge.py”.

Starter code file contains a naïve implementation for the problem. You should write a faster implementation. The starter code reads the 2 inputs and outputs the result in the correct format.

DONOT CHANGE THE PRINT STATEMENT OR OUTPUT/INPUT FORMAT

General instructions:

- This is an individual based assignment, that should be submitted through GOOGLE CLASSROOM.
- You are not allowed to use built-in functions - **code from SCRATCH** - You should write your code in the attached starter file.
- The input/output format **MUST REMAIN THE SAME**
- All code should be submitted in a zip folder that contains (code) named FIRSTNAME_LASTNAME _A1
  - OPTIONAL: You can add your stress testing code as a comment in your code, or in a separate file. You can add your test cases and debugging efforts in an extra markdown file.
- Your code should be clear, understandable, and documented (comments). Follow a consistent naming convention for variables, classes, and functions.
- The due date for the submission of this assignment is Tuesday, 25/10/2022 at 11:59 pm.
- The assignment will be graded out of 10.
- You are permitted to discuss the problems with others in the class. However, you must write up your own solutions to these problems. Any indication to the contrary will be considered an act of academic dishonesty and cheating.