1. [30 points] Write a function that computes the exact value of the n^{th} Fibonacci number,

$$F_{n+1} = F_n + F_{n-1},$$

where $F_0 = 0$ and $F_1 = 1$, using the recurrence relation.

   (a) The function must be named `fibonacci` verbatim.

   (b) The function must take a single non-negative integer as input and return a single **integer** as output.

   (c) Do not use solution to the difference equation (Binet's formula)—use the recursion formula definition of the Fibonacci sequence.

   (d) Do NOT use a Fibonacci function from any library—you must write your own!

2. [30 points] Write a function that determines if a positive integer is a decimal harshad (10-harshad) number. A positive, N-digit integer x written in base-10 (decimal), represented by

$$x = d_{N-1} \times 10^{N-1} + d_{N-2} \times 10^{N-2} + \ldots + a_2 \times 10^2 + d_1 \times 10^1 + d_0 \times 10^0,$$

is 10-harshad number if it is divisible by the sum of the digits:

$$x \bmod \sum_{i=0}^{N-1} d_i = 0.$$

For example, all positive, single-digit integers less than ten are 10-harshad numbers. Also, 117 is 10-harshad since 117 is divisible by 9:

$$117 \bmod (1 + 1 + 7) = 117 \bmod 9 = 0.$$

   (a) The function must be named `is_harshad` verbatim.

   (b) The function must take a single non-negative integer as input and return a boolean `True`/`False` as output.

3. [30 points] Using your `is_harshad` function (do NOT rewrite or copy/paste your code), write a function that returns the first N 10-harshad numbers starting with 1.

   (a) The function must be named `find_narcissistic` verbatim.

   (b) The function must take a single non-negative integer N as input and return a list of the N integers.

4. [30 points] Write a function that estimates the Omega constant, which is the value of the real root of $f(x; 1)$, where $f(x; a) = e^x x - a$, to sixteen decimal places using the fixed-point iteration,

$$x_{n+1} = x_n + \frac{e^{-x} - x}{1 + e^{-x}},$$

using the initial guess $x_0 = 0$. The root of $f(x; a)$ is called the Lambert W function of a, which is defined as the root of the function implicit function

$$xe^x = a$$

so that $\text{LambertW}(a) = x$. This function is also called the omega function or the product logarithm, and appears frequently in the study of delay differential equations, quantum mechanics, and enzyme kinetics.

   (a) The function must be named `compute_lambertW1_pi` verbatim.

   (b) The function will not take any parameters and will return a floating-point number as output where the return value is $\text{LambertW}(1)$.

   (c) Do NOT use the definition of Lambert W function from any library—you must write your own!