National University of Singapore
School of Computing
CS1010S: Programming Methodology
Semester I, 2024/2025

Tutorial 1
Functional Abstraction

Release date: 19th August 2024
Due: 25th August 2024, 23:59

This tutorial consists of **three** questions, which will be discussed during the tutorial session. Please be prepared to answer them.

Note: Please check that your Python installation is configured correctly. See "Mission 0: Setting up Python" for more details.

General Restrictions

- No importing of additional packages unless explicitly allowed to do so.
- Do not use any compound data structures, such as `tuple`, `list`, `dict`, `set`, etc.

Questions

1. Implement a function that takes three numbers as arguments and returns the sum of the squares of the two largest numbers. It is not guaranteed that all numbers are distinct. For example, if the numbers are `1,3,1`, the answer is $3^2 + 1^2 = 10$. If the numbers are `7,7,2`, the answer is $7^2 + 7^2 = 98$.
2. Implement a function `magnitude` that takes in the coordinates of two points on a plane, `(x1, y1)` and `(x2, y2)`, as arguments and returns the magnitude of the vector between them.

   Note: Only for this question, you can consider using the `sqrt` method in the `math` package. To do so, uncomment one of the import statements provided and call the method.

   Sample Execution:

```
def magnitude(x1, y1, x2, y2):
  # Returns the magnitude of the vector
  # between the points (x1, y1) and (x2, y2).

>>> magnitude(2, 2, 5, 6)
5.0
```

3. Implement a function `is_leap_year` that takes one integer parameter and decides whether it corresponds to a leap year, i.e. the function `is_leap_year` returns `True` if the input parameter is true, and `False` otherwise. So which years are leap years? Well, according to Wikipedia:

   > In the Gregorian calendar, the current standard calendar in most of the world, most years that are integer multiples of 4 are leap years. In each leap year, the month of February has 29 days instead of 28. Adding an extra day to the calendar every four years compensates for the fact that a period of 365 days is shorter than a solar year by almost 6 hours. This calendar was first used in 1582.
   >
   > Some exceptions to this rule are required since the duration of a solar year is slightly less than 365.25 days. Over a period of four centuries, the accumulated error of adding a leap day every four years amounts to about three extra days. The Gregorian Calendar therefore omits 3 leap days every 400 years, omitting February 29 in the 3 century years (integer multiples of 100) that are not also integer multiples of 400. For example, 1600 was a leap year, but 1700, 1800 and 1900 were not. Similarly, 2000 was a leap year, but 2100, 2200, and 2300 will not be. By this rule, the average number of days per year is 365 + 1/4 - 1/100 + 1/400 = 365.2425.

   Sample Execution:

```
>>> is_leap_year(2024)
True
>>> is_leap_year(2021)
False
>>> is_leap_year(1900)
False
>>> is_leap_year(2000)
True
```