

Name: \_\_\_\_\_  
Surname: \_\_\_\_\_  
Date: \_\_\_\_\_

## Multiple Choice (10 points each)

### Instructions

Circle the correct answer for each question, as each question has **only one** correct answer.  
Provide reasoning for your answer in the space next to each question.

A correct answer always awards full points (+10 pts).

An incorrect answer with sensible reasoning awards half points (+5 pts).

An incorrect answer with faulty or no reasoning subtracts points (-3 pts).

No answer gives zero points (neither adds nor subtracts points).

Assume each code cell is **run in isolation**, meaning no other code has been run before it.

### Questions

1. After we run the code below, what will be the value of “x”?

```
1 s = "1.5 to the power of 2 is 2.25"  
2 x = 10 * float(s[-4:])
```

- (a) x = 15.0
- (b) x = 22.5
- (c) The code will raise an error.

2. After we run the code below, what will be the value of “x”?

```
1 s = "1.5 to the power of 2 is 2.25"  
2 count = 0  
3 for x in s:  
4     if x in "123456789":  
5         count += 1  
6 print(x)
```

- (a) 5
- (b) 6
- (c) The code will raise an error.

3. Which statement is **FALSE** regarding keyword arguments in Python?

- (a) Keyword arguments are always required.
- (b) You are not allowed to put keyword arguments before positional arguments.
- (c) Keyword arguments are used to set default parameter values.

4. Consider the following block of code. What will be the console's output?

```
1 def exercise(x: list) -> tuple:
2     # Reverse the list
3     rev = x[::-1]
4     return tuple(rev)
5
6 result = exercise("hola")
```

- (a) Python will print "aloh".
- (b) The code runs fine but the user won't see any output.
- (c) The code will raise an error.

5. Which one of these statements is **FALSE** about tuples?

- (a) Tuples are ordered and can be indexed.
- (b) Tuples are mutable.
- (c) Tuples can contain duplicated values.

6. After we run the code below, what will be the value of "numbers\_new"?

```
1 def duplicate_list_elements(ls: list) -> list:
2     ls_new = []
3     while len(ls) > 0:
4         x = ls[0]
5         ls_new.extend([x, x])
6         ls.remove(x)
7     return ls_new
8
9 numbers = [1, 2, 3]
10 numbers_new = duplicate_list_elements(numbers)
```

- (a) numbers\_new = []
- (b) numbers\_new = [1, 2, 3]
- (c) numbers\_new = [1, 1, 2, 2, 3, 3]

# Pseudo-Code

## Instructions

In the following exercises, you will be asked to write Python functions by hand. Please follow these guidelines:

- Any build-in function you need is in the cheat-sheet provided.
- Clearly indicate each level of indentation using a right arrow ( $\Rightarrow$ ).
- Ensure your code works correctly for any possible example.
- If your code has errors, it will be evaluated based on the provided docstring, comments, and explanations.

Full points will be awarded for code that performs as expected under all conditions.

## 7. Approximating $e$ (20 points)

The mathematical constant  $e$ , approximately equal to 2.71828, is fundamental in many areas of mathematics. It can be defined in various ways, two of which are:

### 1. Limit Definition:

$$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n$$

### 2. Series Expansion (Maclaurin Series for $e^x$ at $x = 1$ ):

$$e = \sum_{n=0}^{\infty} \frac{1}{n!} = 1 + 1 + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots$$

Where  $N \rightarrow \infty$ . Your task is to implement two functions:

- `e_limit(n)`: This computes an approximation of  $e$  using the limit definition. (5 points)
- `e_series_expansion(n)`: This computes an approximation using the series expansion. (15 points)

Both functions will utilize the argument `n` as the upper limit for approximation ( $N$  in the equations above).

## 8. Sports Participation (20 points)

Create a Python function named `sport_participation(football, basketball)` that takes two parameters:

- `football`: A list containing names of students who participated in football.
- `basketball`: A list containing names of students who participated in basketball.

The function must output a dictionary with the following pairs of keys and values:

- `"num_football"`: Number of students participating in football.
- `"num_basketball"`: Number of students participating in basketball.
- `"overlap"`: The list of students participating in both sports.
- `"num_overlap"`: Number of students participating in both sports.
- `"is_subset"`: Boolean. Is `True` when one of the lists is a subset of the other, and `False` otherwise.