

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. You want to compute the following expression in Python: $(a + b^2)/(2c)$. Which one of the following options does it right?

- (a) `(a + b^2) / (2 * c)`
- (b) `(a + b**2) / (c * 2)`
- (c) `(a + b**2) / 2 * c`

2. After running the following code, what will be the final value of variable “s”?

```
1 s = "aBcDeFg"  
2 s = s[1::3].lower()
```

- (a) `s = "adg"`
- (b) `s = "be"`
- (c) The code will raise an error.

3. What is the purpose of the “else” statement in an “if-else” block?

- (a) To execute code only if all the above “if” conditions are False.
- (b) To specify an alternative condition.
- (c) To indicate the end of the code block.

4. After running the following code, what will be the final value of variable “x”?

```
1 st_a = {1, 2, 3}
2 st_b = {2, 2, 3}
3 st_c = list(st_a.union(st_b))
4 x = sorted(st_c)[0]
```

- (a) x = 1
- (b) The code will raise an error.
- (c) We cannot predict the outcome.

5. What is the difference between “print(x)” and “return x” inside an user defined function?

- (a) “print(x)” and “return x” are interchangeable; they both output the value of “x”.
- (b) “print(x)” outputs the value of “x” to the console, while “return x” sends the value of “x” back to the calling code.
- (c) “print(x)” will cause an error if we use it inside an user defined function.

6. Consider the following code. Which will be the value of “x” in the **global scope**?

```
1 x = 20
2
3 def exercise(y: int) -> int:
4     x = 4
5     return y / x
6
7 x = exercise(x)
```

- (a) x = 4
- (b) x = 5
- (c) x = 20

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. String Patterns (20 points)

Create a function named `count_pattern(main_str, pattern)` that takes two parameters:

- `main_str`: A string in which you have to find the pattern.
- `pattern`: A string pattern which you have to find in the `main_str`.

The function should return the number of times the `pattern` occurs in the `main_str`, including overlapping instances. For instance, the pattern "121" appears three times in "1212121".

8. Pair Distance Calculator (20 points)

In 2D space, a point can be represented as a pair of coordinates (x, y) . The Euclidean distance between two points (x_1, y_1) and (x_2, y_2) is given by the formula:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Your task is to implement a function `compute_distances(reference_point, point_list)` where:

- `reference_point` is a tuple of two coordinates (x, y) .
- `point_list` is a list of such tuples, each representing a point in the 2D space.

The function should return a list of distances where each distance corresponds to the Euclidean distance between the `reference_point` and each point in `point_list`.