

Name: _____
Surname: _____
Date: _____

Instructions

Always assume each code is run independently, so variables do not persist between exercises.

Questions 1-6: Multiple Choice (5 points each)

Circle the correct answer for each question, as each question has **only one** correct answer. A correct answer awards 5 points. There is no penalty for wrong choices.

Questions 7-9: Debugging (10 points each)

These exercises show code cells with one or more errors. You have to find these errors, explain why they happen, and suggest a fix to them.

Questions 10-11: Open Problems (20 points each)

In these exercises, you will encounter problems that are more open-ended and may not have a single correct answer. Your task is to explain how you would approach each problem by breaking it down into smaller, manageable steps.

Describe how you would structure your code, which logical steps are needed, and what kind of Python tools or constructs (e.g., loops, conditionals, functions, lists, etc.) you would use. You are not required to write full code, but doing so is encouraged if it helps clarify your ideas.

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) / 2 * c`
- (c) `(a + b**2) / (2 * c)`
- (d) `(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 = "abcdefg"`
- (b) `s = "adg"`
- (c) `s = "be"`
- (d) 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.
- (d) To stop an outer loop.

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) x = 2
- (c) The code will raise an error.
- (d) 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)” is a local variable inside the function, while “return x” is a global variable of that function.
- (c) “print(x)” outputs the value of “x” to the console, while “return x” sends the value of “x” back to the calling code.
- (d) “print(x)” will cause an error if we use it inside an user defined function.

6. We run 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
- (d) The code will raise an error.

7. This code aims to count the number of unique words in a string, ignoring case sensitivity. However, it doesn't produce the correct count. Find the error and correct it.

```
1 def count_unique_words(text):  
2     words = text.lower().split(" ")  
3     unique_words = set()  
4     for word in words:  
5         unique_words.add(word)  
6     return len(words)  
7  
8 sentence = "Hello world hello"  
9 print(count_unique_words(sentence)) # Output: 3, Expected: 2
```

8. The following function is supposed to merge two dictionaries, without modifying the originals. In case of key conflicts, the value from the second dictionary should overwrite the one from the first. The code raises an error on line 5: “SyntaxError: cannot assign to function call here”. Identify and fix the issue(s).

```
1 def merge_dictionaries(dict1: dict, dict2: dict) -> dict:
2     merged_dict = dict1
3     for key, value in dict2.items():
4         # Update the key with the value from dict2
5         merged_dict.keys() = value
6     return merged_dict
7
8 dict_a = {"a": 1, "b": 2}
9 dict_b = {"b": 3, "c": 4}
10 print(merge_dictionaries(dict_a, dict_b))
11 # Expected: {"a": 1, "b": 3, "c": 4}
```

9. This code is meant to generate a list of squares of numbers from 1 to 5 using a while loop. However, it results in an infinite loop. Find the mistake and correct it.

```
1 def generate_squares(limit: int=5) -> list[int]:
2     squares = []
3     i = 1
4     while i <= limit:
5         squares.append(i ** 2)
6     return squares
7
8 limit = -1
9 print(generate_squares()) # Expected: [1, 4, 9, 16, 25]
```

10. You are given a string of any length, and you want to determine how many times a specific pattern appears in it, ignoring spaces, uppercase letters, and allowing overlaps. For instance, the pattern “ana” appears four times in “Ana ate an apple and a banana”. Matches include:

- “Ana” at the start, ignoring the uppercase.
- “an-a(pple)” ignoring the space.
- “(b)ana(na)” and “(ban)ana” overlapping.

Design an algorithm to perform the task above. It must work for any possible alphanumeric string and for any target pattern.

Continue exercise 10

11. 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:

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

You have:

- A reference point as a tuple of two coordinates: $A = (x_A, y_A)$
- A list of tuples, each representing a point in the 2D space: $[(x_B, y_B), (x_C, y_C), \dots]$

Design an algorithm that returns a list of Euclidean distances between the reference point A and each of the other points. That is, return: $[AB, AC, \dots]$.

Continue exercise 11