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)

On the table below, mark the correct answer for each question. Use an X to mark your choice. If you want to change your answer, fill the entire box and mark a new one. If multiple boxes are marked for the same question, no points will be awarded. See the examples below. Each question has **only one** correct answer. A correct answer awards 5 points. There is no penalty for wrong choices.

Example	A	B	C	D
Chosen A	X	☐	☐	☐
Chosen C, cancelled A	■	☐	X	☐
Chosen B, cancelled A and C	■	X	■	☐
Invalid (multiple choices)	X	X	☐	☐
Invalid (no 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)

These are open-ended tasks. Explain your approach, outline algorithmic steps, and optionally provide compact working code snippets.

Mark your answers for Questions 1–6 below

Question	A	B	C	D
Q1	☐	☐	☐	☐
Q2	☐	☐	☐	☐
Q3	☐	☐	☐	☐
Q4	☐	☐	☐	☐
Q5	☐	☐	☐	☐
Q6	☐	☐	☐	☐

1. Which of these statements is **FALSE** about **for** loops?

   A. A for loop iterates over any iterable.

   B. A for loop can use two loop variables. (e.g. for i, value in iterable)

   C. After a for loop finishes, the loop variable keeps its last assigned value.

   D. A for loop only iterates over immutable iterables.

2. After running the following code, what will be the final value of variable "str1"?

```
str1 = "GOOD LUCK"
for index in range(len(str1)):
  str1[index] = str1[index].lower()
```

   A. `str1 = "GOOD LUCK"`

   B. `str1 = "good luck"`

   C. The code will raise an error.

   D. `str1 = "GOOD LUCK good luck"`

3. Which of these statements is **FALSE** about functions?

   A. You can access global variables inside a function.

   B. We can directly modify a list inside a function without needing to use a return statement.

   C. A function must always have return statement.

   D. The print() statement inside a function only displays a value, but it does not return any output.

4. What will be the output value of the following code?

```
shopping_list = ["Bread", "Eggs", "Banana", "Cheese", "Olive oil"]

fruit_index = shopping_list.index("Banana")
# Outputs the index of "Banana" in the list

shopping_list[fruit_index + 1:fruit_index + 1] = ["Orange", "Apple"]
print(shopping_list)
```

   A. `["Bread", "Eggs", "Banana", "Orange", "Apple", "Cheese", "Olive oil"]`

   B. The code will raise an error.

   C. `["Bread", "Eggs", "Orange", "Apple", "Banana", "Cheese", "Olive oil"]`

   D. `["Bread", "Eggs", "Banana", "Cheese", "Olive oil", "Orange", "Apple"]`

5. After running this code to find the first five even numbers, what will be the final value of the variable "count"?

```
count = 0
number = 0

while count < 5:
  if number%2 == 0:
    count += 1
    print(number)
  else:
    count = 0
  number += 1
```

A. count $= 4$

B. The code will never stop running.

C. count $= 0$.

D. count $= 5$

6. After running this code, what will be the final value of `serie_set`?

```
def difference_add_serie(number_list: list[int]):
    for i in range(1, len(number_list)):
        yield number_list[i] - number_list[i - 1]
        yield number_list[i] + number_list[i - 1]

number_list = [1, 2, 3, 6, 7]
serie_set = set()
for result in difference_add_serie(number_list):
    serie_set.add(result)
```

A. $\{1, 3\}$

B. $\{3, 5, 9, 13\}$

C. $\{1, 3, 5, 9, 13\}$

D. $\{1, 3, 1, 5, 3, 9, 1, 13\}$

7. The function below is designed to remove all vowels from a given string. However, it doesn't remove all the vowels as expected. Identify the problem and fix the code.

```
def remove_vowels(input_str: str) -> str:
        vowels = ("a", "e", "i", "o", "u")
        result = ""
        for char in input_str:
                if char.lower() in vowels:
                        input_str.replace(char, "")
                return input_str

text = "Programming"
print(remove_vowels(text))  # Output: "Programming"
```

8. You have created a function that stacks two 1D-arrays, one of top of another. If the arrays have a different length, the function fills that difference with zeros. Your function raises an error on line 6: "`ValueError: could not broadcast input array from shape (3,) into shape (0,)`". Find the mistake(s) and correct it.

```
1  import numpy as np
2
3  def stack_uneven(x: np.array, y: np.array) -> np.array:
4          if len(x) > len(y):
5                  y = np.zeros(len(x))
6                  y[len(y):] = y
7          else:
8                  x = np.zeros(len(y))
9                  x[len(x):] = x
10         return np.stack((x, y))
11
12 x = np.array([1, 2, 3])
13 y = np.array([4, 5])
14 print(stack_uneven(x, y))
15 # Expected:
16 # [[1, 2, 3],
17 #  [4, 5, 0]]
18 # Output:
19 # [[1., 2., 3.],
20 #  [0., 0., 0.]]
```

9. The following code should join three arrays together to form the first nine letters of the alphabet. When you run it, the last line returns an error: "`AttributeError: "numpy.ndarray" object has no attribute "concatenate"`". Explain the issue(s) and fix it.

```
import numpy as np

nm = np.array([["a", "b", "c"]])
nn = np.array(["d", "e", "f"])
np = np.array(["g", "h", "i"])

nq = np.concatenate([nm, nn, np])
```

10. You are given a list of reports, each report consisting of a string of single digits separated by spaces, for instance `"1 2 3"`. A report is considered valid if it meets the following conditions:

    1. All digits are strictly increasing from start to end.
    2. The absolute difference between any two adjacent digits is at least 1 and at most 3.

    Write a function `count_valid_reports()` that, given **any** list of reports of **any size**, returns an integer: the number of valid reports. For instance:

```
    ls_reports = [
            "1 2 7 8 9",
            "1 3 2 4 5",
            "1 4 4 6 8",
            "1 3 6 7 9"
    ]

    print(count_valid_reports(ls_reports))  # Output: 1
```

Continue exercise 10

11. Your task is to simulate the circular motion of an object by computing its coordinates. For a circle with radius R and an object moving along its circumference, you need to determine the object's x and y positions corresponding to a list of angles α (in degrees).

    The coordinates are determined by the following formulas:

$$x = R \cdot \cos(\alpha)$$
$$y = R \cdot \sin(\alpha)$$

    Part 1: Implement the function `circular_motion(radius, angles)`:

    - `radius`: A scalar value denoting the radius of the circle, R.
    - `angles`: A list of angles in **degrees**, α.

    The function should compute and return two NumPy arrays: `x_coords` and `y_coords`. These arrays represent the x and y coordinates on the circle for each angle in `angles`. Round this values to two decimals.

    Part 2: Implement a function `verify_coordinates(radius, x_coords, y_coords)` to verify the accuracy of your circular motion function. For each coordinate pair (x, y) in `x_coords` and `y_coords`, check if the equation $\sqrt{x^2 + y^2}$ equals the radius R within a reasonable tolerance (e.g., 0.01). This validation ensures that each calculated point lies on the circle. The function should return "True" if all points are on the circle and "False" otherwise.

Continue exercise 11