

Name: _____
Surname: _____
Date: _____

Debugging (15 points each)

Instructions

The functions below contain one or more errors. You have to find these errors, explain why they happen, and suggest a fix to them.

Questions

1. The following code asks the user for numbers and shows their factorials. However, it fails on line 5: “TypeError: “>” not supported between instances of “str” and “int””. Explain and fix the issue(s).

```
1 input = input("Give me a number: ")
2
3 while input != 0:
4     result = 1
5     while input > 0:
6         result = result * input
7         input -= 1
8
9     print(f"The factorial of your number is: result")
10    input = input("Give me another number: ")
```

2. This code is intended to remove all occurrences of a specific value from a list. However, it doesn't remove all instances as expected. Identify and fix the issue(s).

```
1 def remove_value(lst: list, value: int) -> list:
2     for i in range(len(lst)):
3         if lst[i] == value:
4             lst.pop(i)
5     return lst
6
7 numbers = [1, 2, 3, 2, 4, 2]
8 print(remove_value(numbers, 2))
9 # Expected Output: [1, 3, 4]
10 # Actual Output: [1, 3, 2, 4, 2]
```

3. The function below takes 1D-array and reshapes then to be squared. It works well for the first array, but the second one raises an error. Identify and fix the issue(s).

```
1 import numpy as np
2
3 def square_shape(arr: np.array) -> np.array:
4     if arr.ndim > 1:
5         print("Only valid with 1D-arrays")
6         return
7     length = len(arr)
8     size = length // 2
9     return arr.reshape(size, size)
10
11 arr = np.array([1, 2, 3, 4])
12 print(square_shape(arr))
13 # Output: [[1, 2], [3, 4]]
14
15 arr = np.array([1, 2, 3, 4, 5, 6])
16 print(square_shape(arr))
17 # ValueError: cannot reshape array of size 6 into shape (3,3)
```

4. The following code aims to take a subslice of the array “x”. See what the user expects to get, and what is the actual output. Explain what is happening and fix the code so that we get the expected output.

```
1 import numpy as np
2
3 x = np.array(
4     [[10, 20, 30],
5      [40, 50, 60],
6      [70, 80, 90]])
7 idx_rows = [0, 2]
8 idx_cols = [1, 2]
9 print(x[idx_rows, idx_col])
10 # Expected output: [[20, 30], [80, 90]]
11 # Actual output: [20, 90]
```

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.

5. Removing Characters (20 points)

Create the `remove_first_char()` function to take a string and return a new version. This version should remove all occurrences of the first character, except the first one itself. For example, “sun shines” becomes “sun hine” (all “s” except the first were removed). Note that the function treats lowercase and uppercase as distinct characters.

6. Average Value in Merged Dictionary (20 points)

Create a Python function called `dict_merge()` that merges a list of dictionaries into one unified dictionary. This function should assume that all input dictionaries exclusively contain numeric values. (10 points)

In cases where the same key is present in multiple dictionaries, compute the final value for this key in the resultant merged dictionary as the average of its values. (5 points)

Ensure that the final output values are integers if they do not have a fractional component. (5 points)

```
1 # For instance
2 dict_1 = {"a": 1, "b": 2}
3 dict_2 = {"a": 2, "c": 3}
4 dict_out = dict_merge(dict_1, dict_2)
5 print(dict_out)
6 # Output: {a: 1.5, b: 2, c: 3}
```