🧮 Assignment: Tuples in Python

🎯 **Objective:**
Enhance your proficiency with Python tuples by completing a series of tasks that cover creation, manipulation, and application of tuples in various scenarios.

🟢 Level 1: Beginner – Understanding the Basics

1. **Create a Tuple**
    - Define a tuple containing integers from 1 to 5.
    - Print the tuple and its type.
2. **Access Elements**
    - Retrieve and print the first and last elements of the tuple.
3. **Tuple Length**
    - Determine and print the length of the tuple.
4. **Check Membership**
    - Check if the number 3 exists in the tuple and print the result.
5. **Tuple Concatenation**
    - Create another tuple with integers from 6 to 10.
    - Concatenate both tuples and print the result.

🟡 Level 2: Intermediate – Exploring Tuple Operations

6. **Tuple Unpacking**
    - Given a tuple `person = ("Alice", 30, "Engineer")`, unpack its elements into variables and print them.
7. **Nested Tuples**
    - Create a tuple containing two tuples: one with even numbers and one with odd numbers.
    - Access and print the second element of the first nested tuple.
8. **Slicing Tuples**
    - Slice the concatenated tuple from Level 1 to obtain a tuple of elements from index 2 to 6.
    - Print the sliced tuple.

9. **Tuple Methods**
   - Use the `count()` method to count how many times the number 3 appears in the concatenated tuple.
   - Use the `index()` method to find the index of number 4.
10. **Immutability Test**
    - Attempt to change the first element of the tuple and handle the resulting exception gracefully.

🔴 Level 3: Advanced – Real-world Applications

11. **Student Records**
    - Create a list of student records, where each record is a tuple containing: `(Name, Age, Grade)`.
    - Example: `students = [("Bob", 20, "A"), ("Carol", 22, "B"), ("Dave", 19, "A")]`
    - Print the names of all students with grade 'A'.
12. **Sorting Tuples**
    - Given a list of tuples representing products and their prices:
      `products = [("Laptop", 1200), ("Smartphone", 800), ("Tablet", 400)]`
    - Sort the list in ascending order based on price and print the sorted list.
13. **Tuple Conversion**
    - Convert the `products` list into a dictionary and print it.
14. **Zipping Tuples**
    - Given two tuples:
      ```
      names = ("Eve", "Frank", "Grace")
      scores = (88, 92, 79)
      ```
    - Combine them into a list of tuples using the `zip()` function and print the result.
15. **Function Returning Tuple**
    - Write a function that takes a list of numbers and returns a tuple containing the minimum and maximum values.
    - Example: `min_max([5, 2, 9, 1])` should return `(1, 9)`.

🧮 Assignment: Tuples in Python