Tuple practise

1. Sort Tuple of Tuples by Score (Descending)

Question:
Create a tuple of tuples containing student names and scores. Sort the tuple by score in descending order.

```
# Answer:
students = (("Alice", 85), ("Bob", 90), ("Charlie", 80))
sorted_students = tuple(sorted(students, key=lambda x: x[1],
reverse=True))
print(sorted_students)
# Output: (('Bob', 90), ('Alice', 85), ('Charlie', 80))
```

2. Filter Products Priced Below ₹500

Question:
From a list of tuples representing `(product, price)`, filter out products priced below ₹500.

```
# Answer:
products = [("Laptop", 45000), ("Mouse", 300), ("Keyboard",
700)]
cheap_products = [p for p in products if p[1] < 500]
print(cheap_products)
# Output: [('Mouse', 300)]
```

3. Tuple of Tuples → Dictionary

Question:
Convert a tuple of `(key, value)` pairs into a dictionary.

```
# Answer:
data = (("name", "Sasikala"), ("year", 2025), ("active",
True))
data_dict = dict(data)
print(data_dict)
# Output: {'name': 'Sasikala', 'year': 2025, 'active': True}
```

4. Nested Tuple Access

Question:
Access the value `7` from the following nested tuple:
`nested = (1, (2, 3, (4, 5, (6, 7))))`

```
# Answer:
```

```
nested = (1, (2, 3, (4, 5, (6, 7))))
value = nested[1][2][2][1]
print(value)
# Output: 7
```

5. Tuple Packing with Variable-Length Input

Question:
Write a function that accepts any number of arguments and returns them as a tuple.

```
# Answer:
def pack_into_tuple(*args):
    return args

print(pack_into_tuple("AI", "Python", 2025))
# Output: ('AI', 'Python', 2025)
```

6. Tuple XOR Operation

Question:
Perform element-wise XOR between two tuples of equal length.

```
# Answer:
t1 = (1, 2, 3)
t2 = (3, 2, 1)
xor_result = tuple(a ^ b for a, b in zip(t1, t2))
print(xor_result)
# Output: (2, 0, 2)
```

7. Tuple Frequency Analysis

Question:
Given a tuple of numbers, count how many times each number appears.

```
# Answer:
nums = (1, 2, 2, 3, 3, 3)
freq = {num: nums.count(num) for num in set(nums)}
print(freq)
# Output: {1: 1, 2: 2, 3: 3}
```

If you want, I can make this into a **neatly formatted PDF** where each question has:

- Problem statement
- Example input/output
- Complete solution

That would make it **ready for submission** without extra work.