perplexity

create solutions using python and write the program heading in commit with each

Okay, I will provide Python solutions for each of the programming problems described in the images. I will format each solution with a descriptive commit-style heading.

Problem 1: Tuple Operations

Here's the Python code to handle tuple operations as described:

```
# Commit: feat: Tuple duplicate checker and frequency counter
def tuple_duplicate_check():
    """
    Accepts a tuple of integers, checks for duplicates, and displays them with their freq
    """
    try:
        input_str = input("Enter comma-separated integers: ")
        integer_tuple = tuple(map(int, input_str.split(',')))  # Convert input string to

        duplicates = {}
        for item in integer_tuple:
            if integer_tuple.count(item) > 1 and item not in duplicates:
                duplicates[item] = integer_tuple.count(item)

        if duplicates:
            print("Duplicate elements and their frequencies:")
            for item, frequency in duplicates.items():
                print(f"{item}: {frequency}")
        else:
            print("No duplicate elements found.")

    except ValueError:
        print("Invalid input. Please enter comma-separated integers only.")

# Run the function
tuple_duplicate_check()
```

Problem 2: Nested List Operations

Here's the Python code to handle nested list operations:

```
# Commit: feat: Nested list manipulation: min/max, even replacement
def nested_list_operations():
    """
    Creates a nested list, finds min/max in each sublist, replaces even numbers, and prir
```

```
    """

    nested_list = []
    while True:
        try:
            input_str = input("Enter comma-separated integers for a sublist (or type 'dor
            if input_str.lower() == 'done':
                break
            sublist = list(map(int, input_str.split(',')))
            nested_list.append(sublist)
        except ValueError:
            print("Invalid input. Please enter comma-separated integers only.")

    if not nested_list:
        print("No sublists were entered.")
        return

    for sublist in nested_list:
        if sublist:  # Make sure the sublist is not empty
            print(f"Original sublist: {sublist}")
            print(f"Maximum value: {max(sublist)}")
            print(f"Minimum value: {min(sublist)}")

            for i in range(len(sublist)):
                if sublist[i] % 2 == 0:
                    sublist[i] = "Even"
            print(f"Modified sublist: {sublist}")
        else:
            print("Empty sublist found.")

# Run the function
nested_list_operations()
```

Problem 3: String Operations

Here's the Python code for string operations:

```
# Commit: feat: String processing: lowercase, filter length, and combine
def string_operations():
    """
    Accepts a string, splits it into words, converts to lowercase, filters words by lengt
    """
    input_string = input("Enter a string of comma-separated words: ")
    words = input_string.split(',')
    processed_words = [word.lower() for word in words if len(word.strip()) >= 4]  #strip

    result_string = ','.join(processed_words)
    print(result_string)

# Run the function
string_operations()
```

Problem 4: Sentence Reversal

```
# Commit: feat: Sentence Reversal with Capitalization
def reverse_sentence():
    """
    Accepts a sentence, reverses the order of words, and ensures each word is capitalized
    """
    sentence = input("Enter a sentence: ")
    words = sentence.split()
    reversed_words = words[::-1]
    capitalized_words = [word.capitalize() for word in reversed_words]
    result = " ".join(capitalized_words)
    print(result)

# Run the function
reverse_sentence()
```

Problem 5: List Operations (Duplicates, Second Largest, Sort)

```
# Commit: feat: List operations: remove duplicates, find second largest, sort
def list_operations():
    """
    Accepts a list of integers, removes duplicates, finds the second largest element, and
    """
    try:
        input_str = input("Enter comma-separated integers: ")
        num_list = list(map(int, input_str.split(',')))

        # Remove duplicates
        unique_list = list(dict.fromkeys(num_list))  # preserves order while removing dup

        # Find second largest
        if len(unique_list) >= 2:
            sorted_list = sorted(unique_list, reverse=True)
            second_largest = sorted_list[^1]
            print(f"Second largest element: {second_largest}")
        else:
            print("List does not contain enough elements to find the second largest.")

        # Sort the modified list
        unique_list.sort()
        print(f"Sorted unique list: {unique_list}")

    except ValueError:
        print("Invalid input. Please enter comma-separated integers only.")

# Run the function
list_operations()
```

These are the Python solutions for the programming problems you provided. Each solution is a complete, runnable program with error handling for invalid inputs where appropriate. I have added descriptive commit-style messages for each program as you requested. Let me know if you need any modifications or further assistance!

⁂