Python Assignments

Sum of List Elements

Problem: Write a Python program that calculates the sum of all elements in a list using a loop.

Test Cases:

- **Test Case 1**:
  Input: lst = [1, 2, 3, 4, 5]
  Output: 15
- **Test Case 2**:
  Input: lst = [10, -2, 30, 5, 0]
  Output: 43

Approach:
You can use a for loop to iterate through the list, adding each element to a variable that holds the total sum.

Largest Element in a List

Problem: Write a Python program to find the largest element in a given list of integers using a loop.

Test Cases:

- **Test Case 1**:
  Input: lst = [10, 25, 47, 3, 99, 56]
  Output: 99
- **Test Case 2**:
  Input: lst = [-10, -50, -30, -1]
  Output: -1

Approach:
You can initialize a variable to hold the largest number and update it while looping through the list by comparing each element.

Count Occurrences of an Element

Problem: Write a function that takes a list and an element as input and returns the number of times the element occurs in the list.

Test Cases:

- **Test Case 1**:
  Input: lst = [1, 2, 3, 2, 4, 2], element = 2
  Output: 3

Explanation: The number 2 appears three times in the list.

- **Test Case 2**:
  Input: lst = [10, 10, 20, 30, 10, 20], element = 20
  Output: 2

Explanation: The element 20 appears twice in the list.

Remove Duplicates from List

Problem: Write a Python program to remove all duplicates from a list.

Test Cases:

- **Test Case 1**:
  Input: lst = [1, 2, 2, 3, 4, 4, 5]
  Output: [1, 2, 3, 4, 5]
- **Test Case 2**:
  Input: lst = [10, 20, 20, 10, 30]
  Output: [10, 20, 30]

Approach:
You can create an empty list and loop through the original list. If an element is not already in the new list, add it.

Reverse a List

Problem: Write a Python program to reverse the order of elements in a list without using the built-in reverse() function.

Test Cases:

- **Test Case 1**:
  Input: lst = [1, 2, 3, 4, 5]
  Output: [5, 4, 3, 2, 1]
- **Test Case 2**:
  Input: lst = [10, 20, 30, 40]
  Output: [40, 30, 20, 10]

Approach:
You can loop through the list backward using negative indexing and store the reversed elements in a new list.

List Comprehension for Even Number

Problem: Write a Python program that generates a list of all even numbers between 1 and 100 using a loop.

Test Cases:

- **Test Case 1**:
  Output: [2, 4, 6, 8, ..., 100]
- **Test Case 2**:
  Output: [2, 4, 6, ..., 50] (if the range is changed from 1 to 50)

Approach:
You can use a for loop with an if condition that checks if a number is even (num % 2 == 0) and then append it to a list.

Find Second Largest Number

Problem: Write a Python program to find the second largest number in a list using a loop.

Test Cases:

- **Test Case 1**:
  Input: lst = [10, 20, 30, 40, 50]
  Output: 40
- **Test Case 2**:
  Input: lst = [5, 3, 9, 7, 1]
  Output: 7

Approach:
Find the largest number using a loop, then loop again to find the largest number that is smaller than the largest one.

Cumulative Sum of a List

Problem: Write a Python program that creates a new list where each element is the cumulative sum of elements from the original list.

Test Cases:

- **Test Case 1**:
  Input: lst = [1, 2, 3, 4]
  Output: [1, 3, 6, 10]
- **Test Case 2**:
  Input: lst = [5, 10, 15]
  Output: [5, 15, 30]

Approach:
You can use a loop to iterate through the list and maintain a running total, appending it to a new list.

Intersection of Two List

Problem: Write a Python program that finds the common elements between two lists.

Test Cases:

- **Test Case 1**:
  Input: lst1 = [1, 2, 3, 4], lst2 = [3, 4, 5, 6]
  Output: [3, 4]
- **Test Case 2**:
  Input: lst1 = [10, 20, 30], lst2 = [40, 50, 60]
  Output: []

Approach:
Check if each element from one list appears in the other list, then add it to a new list of common elements.

Split List into Two Parts

Problem: Write a Python program to split a given list into two equal halves. If the list has an odd number of elements, the first half should contain one more element than the second half.

Test Cases:

- **Test Case 1**:
  Input: lst = [1, 2, 3, 4, 5]
  Output: [1, 2, 3], [4, 5]
- **Test Case 2**:
  Input: lst = [10, 20, 30, 40, 50, 60]
  Output: [10, 20, 30], [40, 50, 60]

Approach:
You can calculate the midpoint of the list. Use slicing to divide the list into two parts. For an odd-length list, ensure the first half has one more element than the second.