SWE 30009 Software Testing & Reliability TP 2 2024
Title: Assignment 2

Name (Student ID): Joseph Linao (104556329)

Lab class: Wednesday / 10:30 AM to 11:30 AM / BA709

Due Date: Monday 9th September 2024 at 11:00 PM

Task 1:

- Test case 1:
  - Input: [15, 26, 30, 16, 7, 45]
  - Output: [7, 15, 16, 26, 45] and []
  - Purpose: to test the scenario where all inputs are positive. This should ensure that the negative list remains empty.
- Test case 2:
  - Input: [-1, -11, -9, -20, -50]
  - Output: [] and [-50, -20, -11, -9, -1]
  - Purpose: to test the scenario where all inputs are negative. This should ensure that the positive list remains empty.
- Test case 3:
  - Input: [10, -5, -10, 10, 20, -5]
  - Output: [10, 20] and [-10, -5]

    - Purpose: to test the handling of both positive and negative integers, including duplicates, and if check if duplicates are removed and sorted correctly.
- Test case 4:
    - Input: [-100, 100, 50, -100, -50, 50]
    - Output: [100, 50] and [-100, -50]
    - Purpose: to test the upper and lower bounds of the allowed integer values and how they are processed.
- Test case 5:
    - Input: [6, -9]
    - Output: [6] and [-9]
    - Purpose: to test the smallest non-trivial input where one positive and one negative number exist, ensuring correct sorting and separation.
- Test case 6:
    - Input: [-1, 0, -7, -11]
    - Output: [] and [-11, -7, -1]
    - Purpose: to test the handling of the number 0 (which is considered negative in this assignment) and its exclusion from both lists.

Task 2:

- Test case 3: [10, -5, -10, 10, 20, -5]
    - It covers both positive and negative numbers
    - It tests the handling of duplicate numbers, ensuring that duplicates are correctly removed in both lists.
    - It verifies if the lists are sorted properly in ascending order after removing duplicates.
    - This test case is comprehensive and can expose potential bugs in the program's core logic, such as how the program handles separation, sorting, and deduplication of integers.

Task 3:

- Test case 1:
    - Input: [15, 26, 30, 16, 7, 45]
    - Expected Output: [7, 15, 16, 26, 30, 45] and []
    - Actual Result: [7, 15, 16, 26, 30, 45] and []

    - Explanation: the program correctly handles an all-positive input, sorting the positive integers in ascending order and leaving the negative list empty as expected
- Test case 2:
    - Input: [-1, -11, -9, -20, -50]
    - Expected Output: [] and [-50, -20, -11, -9, -1]
    - Actual Result: [] and [-50, -20, -11, -9, -1]
    - Explanation: the program handles an all-negative input, similarity to test case one, which allow the program to sort all negative integers in ascending order while leaving the positive integer list empty.
- Test case 3:
    - Input: [10, -5, -10, 10, 20, -5]
    - Expected Output: [10, 20] and [-10, -5]
    - Actual Result: [10, 10, 20] and [-10, -5, -5]
    - Explanation: the program fails to remove duplicates; this indicates that the program does not handling duplicates before sorting the list.
- Test case 4:
    - Input: [-100, 100, 50, -100, -50, 50]
    - Expected Output: [100, 50] and [-100, -50]
    - Actual Result: [50, 50, 100] and [-100, -100, -50]
    - Explanation: Similarity to test case 3, the program does not allow to remove duplicates.
- Test case 5:
    - Input: [6, -9]
    - Expected Output: [6] and [-9]
    - Actual Result: [6] and [-9]
    - Explanation: this program can even handle the small input with one positive and negative inputs
- Test case 6:
    - Input: [-1, 0, -7, -11]
    - Expected Output: [] and [-11, -7, -1]
    - Actual Result:
    - Explanation: since the actual result is blank, meaning that 0 is excluded from the positive and negative lists, thus the program will not function.