Name: Vu Ha Phuong
Student ID: 104177306

Software Testing and Reliability
SWE30009

ASSIGNMENT 2

TASK 1

	Input	Output	Objective
Test case 1	[0, 1, 2, 3]	Negative numbers: [0] Positive numbers: [1, 2, 3]	This test case contains only positive numbers and number 0. It serves to test whether the program correctly considers 0 as a negative number.
Test case 2	[3, -1, 2, -5, 1, -2]	Negative numbers: [-5, -2, -1] Positive numbers: [1, 2, 3]	This test case checks if the program correctly splits positive and negative integers into two separate lists.
Test case 3	[-10, -20, -5, -6, -2]	Negative numbers: [-20, -10, -6, -5, -2] Positive numbers: []	This test case is to test the scenario where all integers are negative hence the output of positive integers is empty.
Test case 4	[-3, -2, -1, 3, 2, 1]	Negative numbers: [-3, -2, -1] Positive numbers: [1, 2, 3]	This test case contains both negative and positive numbers, where all numbers are sorted in descending order. It serves to test if the program correctly sorts both the positive and negative integers in ascending order.
Test case 5	[-5, -5, -2, -2, -1, 3, 5, 7]	Negative numbers: [-5, -2, -1] Positive numbers: [3, 5, 7]	This test case contains duplicated negative integers. It serves to test the proper removal of duplicated integers in negative list.
Test case 6	[-15, -14, -13, -12, -11, -10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]	Negative numbers: [-15, -14, -13, -12, -11, -10, -9, -8, -7, -6, -5, -4, -3, -2, -1] Positive numbers: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]	This test case contains 30 unique integers, both negative and positive, without any duplicates and without the number 0. This test case is used to test whether the program can handle input with the maximum number of characters. The output will include all 30 numbers, without any missing.

Name: Vu Ha Phuong
Student ID: 104177306

TASK 2

Selected test cases: *Test case 4*.

- **Description:** This test case contains both negative and positive numbers, where all numbers are sorted in descending order. It serves to test if the program correctly sorts both the positive and negative integers in ascending order.
- **Justification:**
    1. This test case includes both positive and negative integers to ensure that the program is capable of handling mixed input values. It checks the program's ability to correctly sort these numbers into their respective lists and tests whether the sorting function is working correctly, which are all fundamental functions of the program.
    2. This test ensures that the program performs correctly with all possible inputs. Additionally, with the inclusion of the number 0 in this range of inputs allows us to verify whether program handles it correctly. Handling 0 is another key requirement of this program.
    3. With the wide range of input values (but only in the range of -100 to 100), this test case can accept input list include duplicate integers, which helps to test whether the program can accurately identify and remove duplicates before sorting. Handling duplicates is also a key requirement for this program.
    4. The test case closely simulates real-world scenarios where inputs might contain a mix of positive and negative numbers, as well as duplicates.

- **Conclusion:** This test case is the most comprehensive of the six because it covers a wide range of scenarios in a single test. By choosing this test, we can maximize the chances of detecting any bugs or issues in the program, making this test case the best option for a single test with limited resources.

TASK 3

	Input	**Expected output**	**Real output**	**Justification**	**Program improve suggestion**
Test case 1	[0, 1, 2, 3]	+ Negative numbers: [0] + Positive numbers: [1, 2, 3]	Nothing	The program incorrectly handles the number 0. It should treat 0 as a negative number, placing it in the negative list. However, the program currently throws an error and treats 0 as an invalid input.	+ Remove "if" function for number 0. + Change the result of "neg_nums" variable. This variable not only accept number smaller 0 but also accept 0 as a valid result.

Name: Vu Ha Phuong
Student ID: 104177306

Test case 2	[3, -1, 2, -5, 1, -2]	+ Negative numbers: [-5, -2, -1] + Positive numbers: [1, 2, 3]	+ Positive numbers: [1, 2, 3] + Negative numbers: [-5, -2, -1]	This program correctly splits positive and negative integers into two separate lists. Therefore, the splitting function in this program is correct.	
Test case 3	[-10, -20, -5, -6, -2]	+ Negative numbers: [-20, -10, -6, -5, -2] + Positive numbers: []	+ Positive numbers: [] + Negative numbers: [-20, -10, -6, -5, -2]	This program correctly handles the positive and negative inputs. When there are no positive inputs, it will return an empty list.	
Test case 4	[-3, -2, -1, 3, 2, 1]	+ Negative numbers: [-3, -2, -1] + Positive numbers: [1, 2, 3]	+ Positive numbers: [1, 2, 3] + Negative numbers: [-3, -2, -1]	The program correctly sorts both lists in ascending order. Therefore, the sorting function in this program is correct.	
Test case 5	[-5, -5, -2, -2, -1, 3, 5, 7]	+ Negative numbers: [-5, -2, -1] + Positive numbers: [3, 5, 7]	+ Positive numbers: [3, 5, 7] + Negative numbers: [-5, -5, -2, -2, -1]	The program incorrectly handles duplicate numbers. There are two duplicate numbers in the input, but the output still not remove them.	Adding a "set()" method after sorted each list. The set method in python will return a non-repeating elements list.
Test case 6	[-15, -14, -13, -12, -11, -10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]	+ Negative numbers: [-15, -14, -13, -12, -11, -10, -9, -8, -7, -6, -5, -4, -3, -2, -1] + Positive numbers: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]	Nothing	The valid input is allowed to have maximum 30 characters, while this program only allow maximum 20 characters.	Change the "if" condition for checking the input length. Instead of 20 as a maximum character, change to 30.