

COMP1036 Coursework

Release Date: 18 November 2023 16:00

Deadline: 16 December 2023 16:00

Part II (25 marks)

Tasks

Write a program in Hack Assembly Language. The program is able to sort an array of integers in ascending or descending order. The unsorted array has 5 elements, located at a range of memory locations from RAM[11] to RAM[15]. The integers stored in the array are positive, negative, or zero.

The program should allow you to sort the whole array of 5 elements. Alternatively, it should also allow you to sort part of the array, by specifying the number of first few elements to be sorted. You should indicate your preference at the memory location RAM[0] by entering an integer number in the range of [-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5]. A positive integer indicates that the sorting should be done in ascending order, and a negative integer indicates a descending order.

For example, if RAM[0] contains a positive integer 5, the program will sort the entire array of 5 elements at locations RAM[11] to RAM[15] in ascending order (i.e., from the smallest number to the largest number). If RAM[0] contains a negative integer -5, the entire array will be sorted in descending order (i.e., from the largest number to the smallest number).

Another example, if RAM[0] contains a positive integer 3, only the first 3 elements of the array will be sorted in ascending order (i.e., from the smallest number to the largest number). The last 2 elements will remain unsorted. If RAM[0] contains a negative integer -3, the first 3 elements of the array will be sorted in descending order (i.e., from the largest number to the smallest number).

Some Input and Output Examples:

	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out
RAM[0]	1	1	3	3	5	5	-4	-4	-5	-5	5	5	-5	-5
<i>Ignore the values in RAM[1], RAM[2], ..., RAM[10]</i>														
RAM[11]	3	3	3	2	3	1	3	5	3	5	-3	-5	-3	-1
RAM[12]	2	2	2	3	2	2	2	3	2	4	-2	-4	-2	-2
RAM[13]	5	5	5	5	5	3	5	2	5	3	-5	-3	-5	-3
RAM[14]	1	1	1	1	1	4	1	1	1	2	-1	-2	-1	-4
RAM[15]	4	4	4	4	4	5	4	4	4	1	-4	-1	-4	-5

Requirements

1. RAM[0] is reserved for holding the number of elements of the array to be sorted, as well as indicating the direction of sorting (ascending/descending).
2. RAM[11] to RAM[15] are reserved for holding the array integers before sorting, as well as the results after sorting.
3. You may need to clear the temporary variables after the execution of one test case. Failing to do so may result in improper execution of your program for the subsequent test cases.
4. Each test case is an independent trial. So, you do not need to loop back to the beginning after the execution of one test case.
5. Sample test cases will be provided to evaluate your program. But the final test cases will be different from the sample test cases but in the same format.
6. You should use the sample test file to evaluate your program. Your program will be finally evaluated using the similar test file. Failing to comply with the test file format may result in your program fail on final test cases.
7. Make sure that your program is bug-free. Zero mark will be immediately given to non-executable code.
8. You should complete all tasks in one assembly file. Name your assembly file as **main.asm**

Marking Criteria

1. 25 marks in total.
2. 20 test cases will be used, each test case carries one mark.
3. Your program should be well-structured and properly documented. 5 marks will be allocated for the proper use of remarks, pseudocode, indentation, built-in symbols, labels, and variables.
4. The efficiency of your program will also be considered. A limited number of tick-ticks will be given for each test case. A reasonable number of tick-ticks will be used to test your program. If your program could not generate the expected results within the given number of tick-ticks, you will lose one mark for that test case.

Plagiarism

If you use code you found in a textbook or on the web, you must acknowledge it. The plagiarism detector tools will be used to check for similarities between submissions and web-based material. You are reminded of the School's Policy on Plagiarism.

How to submit

You should zip all your files into one zip file. You should name your file as: YOURSTUDENTID_YOURNAME.zip (e.g., **20514000_Danting_Wang.zip**).

Remember to include your student ID and your name at the beginning of main.asm file. Submit your zip file onto Moodle page. Please note that every next submission overwrites all the files in the previous one. If you submit several times, make sure that your last submission includes all the necessary files.

Do check the submission after you submit it. It happens in the past that the submitted file is corrupted. You should ensure that the submitted file is complete and executable. You will receive **zero mark** if your submitted file is corrupted or not executable.

For late submission, the standard late submission policy applies, i.e., **5% mark deduction for every 24 hours**. (Including weekends and public holidays).