

## EE306 Introduction to Computing

### Programming Assignment # 2

The purpose of this assignment is to write a program in LC-3 assembly language to sort a list of exam scores, and report the range, number of students and median.

The list of exam scores, integers between and including 0 and 100, along with corresponding **unique non-zero** student ID will be stored in memory starting at address x4003. The ID number and score of each student will be specified as follows:

- **Bits [15:8]** – student ID number
- **Bits [7:0]** – student score

The end of this list is specified using a student ID of zero, for which exam score could be **anything**, as sentinel.

After sorting, the exam scores in descending order along with corresponding student IDs must be stored in memory starting at location x4003 and, just as initial list, it needs to be terminated with null student ID. In addition to sorting the list, you are also asked to report the following:

- Range as a concatenation of highest score in upper half (bits [15:8]) and lowest score in lower half (bits [7:0]) in location x4000.
- Number of students as 16-bit integer in x4001.
- Median<sup>1</sup> as 16-bit integer in x4002.

Example: Tables 1 and 2 show content of memory locations corresponding to list and outputs before and after execution of program for a given list.

*Table 1. State of memory before program is executed*

Address	Bits[15:8] (in decimal)	Bits[7:0] (in decimal)
x4000	–	–
x4001	–	–
x4002	–	–
x4003	23	78
x4004	10	91
x4005	56	5
x4006	2	87
x4007	13	61
x4008	84	53
x4009	91	63
x400A	45	48
x400B	67	23
x400C	19	97
x400D	114	78
x400E	0	–

---

<sup>1</sup> For odd number of students, median is well-defined. For even number of students, median is equal to integer part of average of two middle numbers.

Table 2. State of memory after program is executed

Address	Bits[15:8] (in decimal)	Bits[7:0] (in decimal)
x4000	97	5
x4001	0	11
x4002	0	63
x4003	19	97
x4004	10	91
x4005	2	87
x4006	114	78
x4007	23	78
x4008	91	63
x4009	13	61
x400A	84	53
x400B	45	48
x400C	67	23
x400D	56	5
x400E	0	–

### Hints:

1. Use masks to separate the ID numbers from the scores.
2. Note that the length of the list is not specified, though the maximum number of students is determined by the maximum possible unique student IDs that can be specified in bits [15:8]. When you test your program for a variety of test cases make sure that you include a case with zero students, and a case with the maximum number.

### Notes:

1. Student IDs are all unsigned.
2. If there are zero students, the range and median should be set as -1 (xFFFF)
3. If there are multiple students with the same score, you do not need to sort by ID number.
4. You can use any algorithm for sorting. Start the assignment with a flowchart.
5. Start your program at location x3000. So the first line of your assembly program must be .ORIG x3000.
6. You can test your program by manually loading data in locations x4003 onwards (with the last item in the list being the null ID number).
7. The file that you will upload to Canvas for this assignment must be named **sortscores.asm**.