EE306 Introduction to Computing

Lab 3 (due on 4/2, 9pm, on GitHub)
Course Instructor: Dr. Nina Telang

All Lab assignments must be completed individually. You are not permitted to seek help or clarification from anyone other than the instructor or the TAs.

Your file should be named exactly after your EID, for example, xy1234.asm. Your program will not be graded if you fail to follow the file naming convention.

Problem Statement:

Write a program in **LC-3 assembly language** to sort a list of students and the number of credit hours they completed.

The list of credit hours completed by students, **integers** between 0 and 150 **(for the sake of this lab)**, along with the corresponding **unique non-zero** ID of the students will be stored in memory starting at address x4004. The ID and Credit hours of each student will be specified as follows:

- **Bits [15:8]** – Unique ID
- **Bits [7:0]** – Credit Hours Completed

The end of this list is specified using a Unique ID of **zero**, for which the Credit Hours field could be **anything**. (This entry merely acts as the sentinel or terminator.)

Sort the list in **ascending order** of the Credit hours along with the corresponding unique IDs starting at location x4004 and, just as the initial list, it needs to be terminated with null ID.

EE306 Introduction to Computing

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)
x4004	23	28
x4005	10	21
x4006	56	15
x4007	2	19
x4008	13	41
x4009	84	95
x400A	91	89
x400B	45	90
x400C	67	73
x400D	19	35
x400E	114	105
x400F	0	–

Table 2. State of memory after program is executed

Address	Bits[15:8] (in decimal)	Bits[7:0] (in decimal)
x4004	56	15
x4005	2	19
x4006	10	21
x4007	23	28
x4008	19	35
x4009	13	41
x400A	67	73
x400B	91	89
x400C	45	90
x400D	84	95
x400E	114	105
x400F	0	–

2

Notes

1. Unique IDs are all unsigned, and non-zero.
2. There could be zero individuals in the list.
3. If there are multiple individuals with the same number of credit hours, **do not sort by ID number.**
4. You can use any algorithm for sorting. As always, start 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 x4004 onwards (with the last item in the list being the null ID number).
7. Note that the length of the list is not specified, though the maximum number of individuals is determined by the maximum possible unique 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 in the list, and a case with the maximum number.

SUBMISSION INSTRUCTIONS:

The file that you will upload to online repository for this assignment must be named
youreid.asm

For example, if your eid is **ma123** then the file name should be ma123.asm