EE 306: Introduction to Computing

Programming Lab 2

Course Instructor: Prof. Nina Telang
TAs: Aniket Deshmukh, Shruthi Krish

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.

Due date: 3/9, 11:59 PM

Problem Statement

The task for this assignment is to use LC3 assembly to create a histogram based on the scores of several students that are stored in an array. To do this we first sort the array in ascending order, and then examine its entries to create the histogram. The entries of the array adhere to the following format:

- Each student has a **unique ID** which is **3 ASCII characters** long.
- A score is associated with each ID which an **unsigned integer** between (and including) **0 and 100**.
- The entry for one student is stored in **2 consecutive words** in the memory.

(a) Word 1 (lower address):	Bits [15:8] -	Character 1
	Bits [7:0] -	Character 2
(b) Word 2 (higher address):	Bits [15:8] -	Character 3
	Bits [7:0] -	Score

The array starts at address **x4005**. The end of the array is specified using **x0000** for the first two characters of the student ID as the sentinel value; the array terminates when this value is found in the first word, regardless of the second word (the 3rd character and score) associated with it.

After sorting, each entry of the sorted array needs to be stored starting at the same location, **x4005**. For reference, a histogram is a table representing the number of items that fall between a given set of intervals. We create the histogram based on the scores, and the intervals are (**all inclusive**):

- 0-19
- 20-39
- 40-59
- 60-79
- 80-100

The number of students falling in each of the 5 intervals is stored in consecutive memory locations starting at address **x4000**. Bits [7:0] contain the number of student, and bits [15:8] contain the names of each interval in ACSII, which are ‘**F**’, ‘**D**’, ‘**C**’, ‘**B**’, and ‘**A**’.

Example: Tables 1 and 2 show contents of memory locations corresponding to an array before and after execution of the program.

Table 1. State of memory before program is executed

Address	**Bits[15:8]** (in decimal)	**Bits[7:0]** (in decimal)
x4000	-	-
x4001	-	-
x4002	-	-
x4003	-	-
x4004	-	-
x4005	110	105
x4006	110	87
x4007	97	110
x4008	105	79
x4009	115	104
x400A	114	100
x400B	105	100
x400C	107	60
x400D	0	0
x400E	-	-

Table 2. State of memory after program is executed

Address	**Bits[15:8]** (in decimal)	**Bits[7:0]** (in decimal)
x4000	70	0
x4001	68	0
x4002	67	0
x4003	66	2
x4004	65	2
x4005	105	100
x4006	107	60
x4007	97	110
x4008	105	79
x4009	110	105
x400A	110	87
x400B	115	104
x400C	114	100
x400D	0	0
x400E	-	-

Notes

- The maximum number of entries in the array is 256.
- If there are multiple students with the same score, they can be placed in any order.
- You can use any algorithm for sorting, but make sure to understand all the details before you start writing your code.
- It is highly recommended that you make flowcharts for the program flow.
- Make sure to thoroughly test you program so that all corner cases are covered.
- This assignment is to be completed in **assembly**. Push your file titled **lab2.asm** onto branch **lab2** of your online repository for the course.