

Lab04 Sort and Count

Task

In this programming assignment, you are asked to "sort an array."

Now you have **16** students' scores.

If a student scores 85 or above and is in top 25%, he/she will receive an A.

If he/she does not get an A but scores 75 or above and is in top 50%, he/she will get a B.

Note:

- Each score is stored in successive memory locations starting with address **x4000**.
- Each score is an integer between 0 and 100. ($0 \leq score \leq 100$).
- Everyone gets a **different** score.

Your job:

1. Your program should sort scores in **ascending order** (smallest-to-largest) and store them in successive memory locations starting with address **x5000**.
2. Your program should count how many students get an A and store the number in **x5100**.
3. Your program should count how many students get a B and store the number in **x5101**.

R0-R7 are set to zeroes at the beginning, and your program should start at **x3000**.

Here are some examples:

Input:

Memory address	example1	example2	example3
x4000	100	95	88
x4001	95	100	77
x4002	90	0	66
x4003	85	50	55
x4004	80	45	99
x4005	60	40	33
x4006	55	80	44
x4007	50	65	22
x4008	45	70	11
x4009	40	75	10
x400A	35	35	9
x400B	30	20	98
X400C	25	25	97
X400D	20	15	53
X400E	10	10	57
X400F	0	90	21

Output:

Memory address	example1	example2	example3
x5000	0	0	9
x5001	10	10	10
x5002	20	15	11
x5003	25	20	21
x5004	30	25	22
x5005	35	35	33
x5006	40	40	44
x5007	45	45	53
x5008	50	50	55
x5009	55	65	57
x500A	60	70	66
x500B	80	75	77
X500C	85	80	88
X500D	90	90	97
X500E	95	95	98
X500F	100	100	99

Memory address	example1	example2	example3
x5100	4	3	4
x5101	1	2	1

Test

Manually set 16 numbers in memory might be cumbersome. We provide you with [this website](#) for testing. Helper output information contains 18 numbers separated by comma. The first 16 numbers are the memory content between **x5000** and **x500F** after sorting, and the latter two are the count of grade A and B.

Score

Correctness for 50% and the report for other 50%.

Submission

Here are some notifications:

- Your program should start with **.ORIG x3000**
- Your program should end with **.END**
- Your last instruction should be **TRAP x25 (HALT)**
- **Capitalized** keywords (also labels) are recommended (For example, use "ADD" instead of "add", use

"NUMBER" instead of "number")

- **Spaces** after **commas** (`ADD R0, R0, #1` rather than `ADD R0,R0,#1`)
- **Decimal** constants start with #, **hexadecimal** with lowercase x
- Write **comments** when necessary

You may also refer to the textbook for more details about code style.

Your submission be structured as shown below.

```
1 | PB21*****_Name.zip
2 | └─ PB21*****_Name_report.pdf
3 | └─ lab4.asm
```

Reports

Your reports should contain at least the five parts below:

- purpose
- principles (e.g. how to deal with modulus)
- procedure (e.g. bugs you encountered and how to solve them)
- results of your test