

# CS 1428 Honors

## Lab 4

Jared Wallace

### Questions

1. (15 pts) The memory(RAM) for our processor has a very small capacity. It will only be 256 “cells” of storage, with each cell storing a single integer value. We will implement the memory as an array. How can we declare this array of storage?
2. (15 pts) As already mentioned in question 2, each instruction the user inputs will consist of one, and only one, line. The format of each line is as follows:

operation-code | data-cell-0 | data-cell-1 | data-cell-2

The content placed in each data-cell depends on the instruction. For instance, the arithmetic instructions will use data-cell-0 for the index of the memory(array) cell to write *to*, data-cell-1 as the index of the *left* operand, and data-cell-2 as the index of the *right* operand. Accordingly, assuming we have a memory array that looks like

value	4	23	10	40	5
index	0	1	2	3	4

in order to subtract 23 from 10, and store the result in memory cell 0, we would have the following line:

1(for subtract) 0(store in index 0) 1(index of 23) 2(index of 10)

Now write some code that reads in an instruction line, and then processes and performs the appropriate instruction. (Remember, addition is 0, subtraction is 1, multiplication is 2, division is 3, and exponentiation is 4) (Hint: It would be much easier to use a switch statement, much like the “calculator” we wrote last week)

3. (15 pts) We also need to develop some read and write instructions to allow the program to accept input from, and output to, the user. (Otherwise the assembler wouldn't be very useful) Add the necessary lines to the switch statement you already have, so that read and write instructions can be handled with the following format: (the *read* instruction accepts user input and stores it into the proper cell, and the *write* instruction takes the value of the specified cell and prints it to standard out)

$$\begin{array}{l|l|l|l} 5(\text{op-code read}) & \text{index of the cell to store the value} & 0 & 0 \\ 6(\text{op-code write}) & \text{index of the cell to read from} & 0 & 0 \end{array}$$

4. (15 pts) Lastly, we need a way for the program to directly store constant values into its own memory. This is done by the store command. If store is implemented as follows:

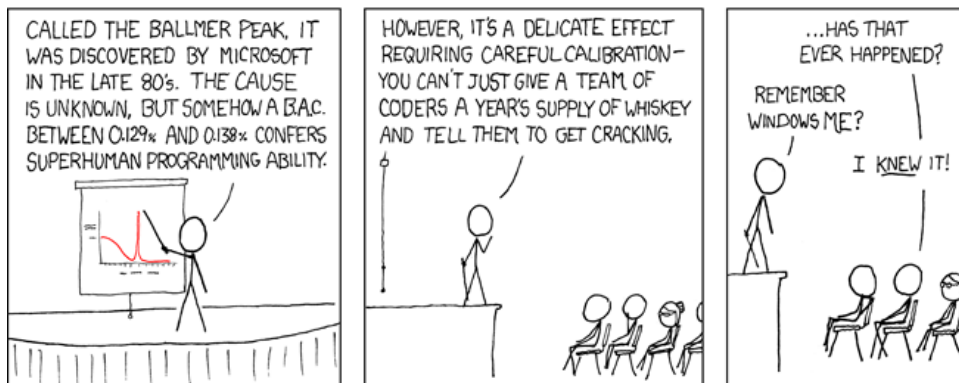
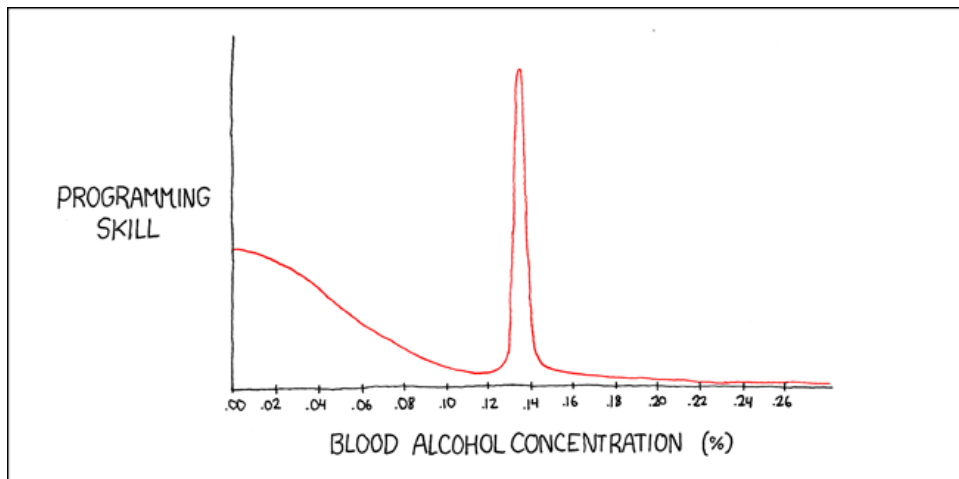
$$7(\text{op-code store}) \mid \text{index of the cell to store to} \mid \text{value to store} \mid 0$$

add the additional lines necessary to your switch statement to implement the store instruction.

5. (40pts) Using the snippets you have written in the previous questions, create a single program (lab4h.cpp) that:
- Prompts the user to enter the number of lines the program will have
  - Prompts the user line by line for the “code”
  - After the specified number of instruction lines have been entered, the program should execute the inputted instruction lines.
  - You can test the correctness of your program by inputting the instruction lines inside the supplied *age.txt* file.

## Deliverables

Hard copy of the source code you wrote (lab4h.cpp) and the answers to the questions. Soft copy (upload to homework upload) of all your source code files.



Apple uses automated schnapps IVs.