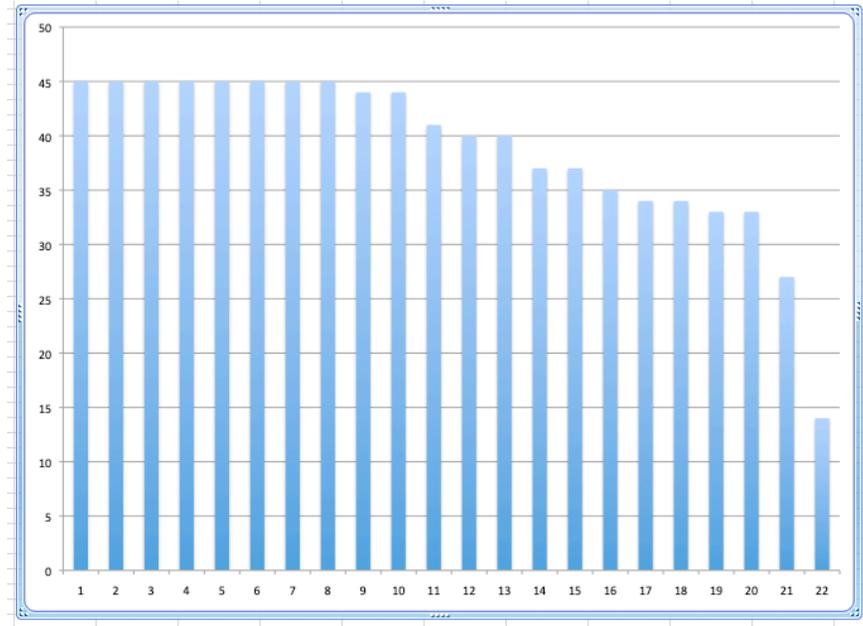


## COS 109 Problem Set 3: Solutions

Graded out of 45. As always, take a careful look at any place where you went wrong and be sure that you understand it for the future. Francisca, Abby and I are happy to help.



**Problem 1:**  $3 + 6 * 2 = 15$

(a) Your three colors, with names and hex values:

Creative and funny names much in evidence; thanks to those who included a visual of their colors.

(b) Write down the hex representations with 7-10-7 allocation of bits.

red	1111 1110 0000 0000 0000 0000	<b>FE 00 00</b>
green	0000 0001 1111 1111 1000 0000	<b>01 FF 80</b>
blue	0000 0000 0000 0000 0111 1111	<b>00 00 7F</b>
yellow	1111 1111 1111 1111 1000 0000	<b>FF FF 80</b>
cyan	0000 0001 1111 1111 1111 1111	<b>01 FF FF</b>
magenta	1111 1110 0000 0000 0111 1111	<b>FE 00 7F</b>

If you blew this question, it's important to figure out where the confusion lies, since questions like this just coincidentally tend to appear on exams, because they test your understanding of how bits, bytes, and hexadecimal notation work.

**Problem 2:  $4 + 4 + 2 + 2 + 2 = 14$**

(a) How many bits for the population of the USA? How many bytes?

300-330M people: **28 or 29 bits; 4 bytes**

(b) How many bits for the population of California? How many bytes?

40M people: **26 bits, 4 bytes**

(c) How many bytes for US GDP of \$2 trillion?

2 trillion =  $2 \times 10^{12}$ ; 41 bits, **6 bytes**

(d) How many bytes for the US national debt of \$28 trillion?

28 trillion =  $28 \times 10^{12}$ ; 45 bits, **6 bytes**

(e) How many bytes for the number of monthly FB users, 2.7 billion?

$2.7 \times 10^9 = 32$  bits = **4 bytes**

Very well done for the most part.

**Problem 3:  $1 + 5 = 6$**

(a) What section of the von Neumann paper discusses multiplication hardware tradeoffs?

**Section 1.5, section 5.2**

(b) What extra code did you add or modify in the simulator?

```
} else if (opcode[pc] == "mul") {  
    accumulator = parseFloat(accumulator) *  
        parseFloat(litoradr(adr[pc]));
```

plus two error messages that warn about erroneous instruction names.

I didn't worry about also fixing the documentation but kudos if you did.

**Problem 4: 5 + 5**

(a) Paste your program for printing square and fourth power here:

```
get
store m
mul m
print
store m (or mul m; works the same)
mul m
print
stop
m
```

(b) Paste your program for counting down here:

This works for non-negative inputs; it's a good exercise to deal with negative inputs as well, which will take one more instruction and some rearrangement.

```
get
top print
sub 2
ifpos top
stop
```

There were a lot of lines like

```
ifzero stop
```

which, though it is intuitively appealing, is not valid because there was no location named stop. You have to explicitly go to a separate STOP instruction. That means that for some inputs, the program will quit early with an error message. If it did, that suggests that you didn't test it thoroughly.