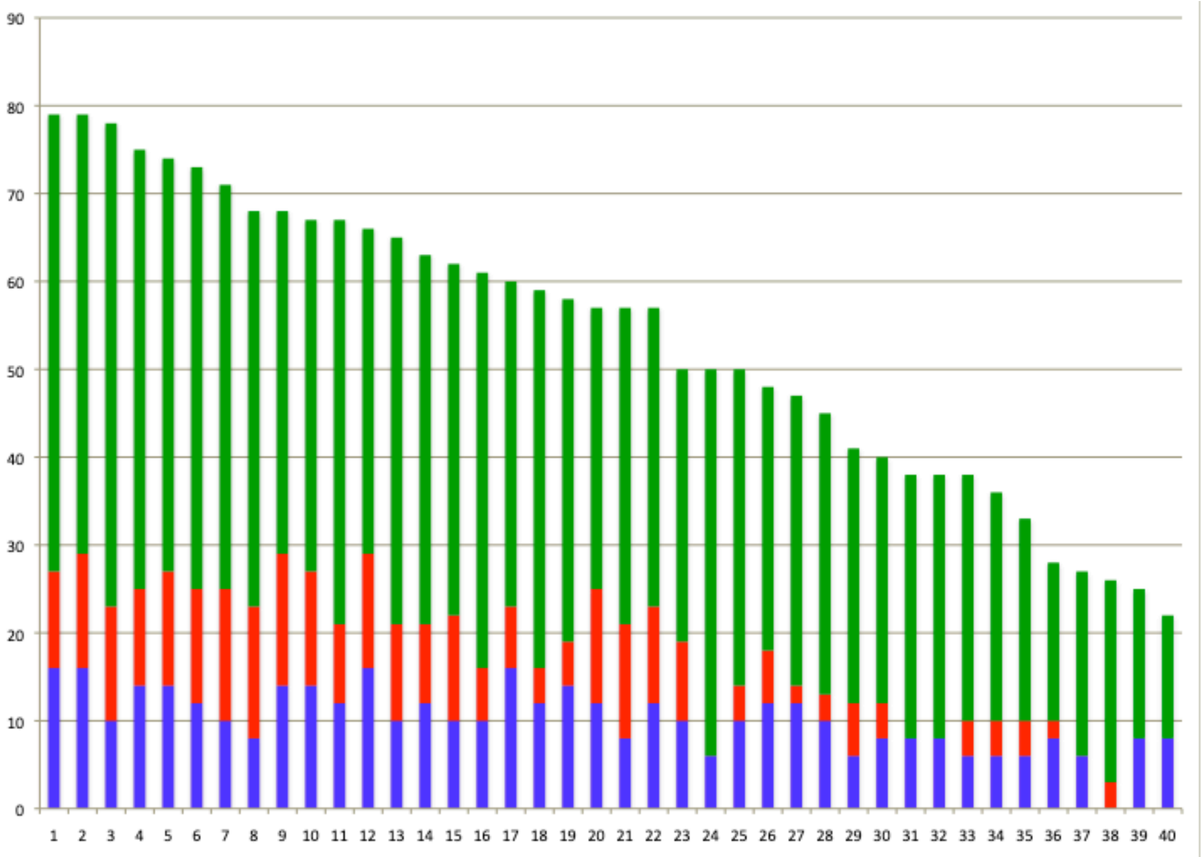
I graded this myself. The class has 40 members this year, compared to 30-ish last year and the circumstances are more or less normal, so I'm not sure whether a comparison reveals anything, but for whatever it's worth, last year's median, lower quartile and upper quartile were 61, 52 and 71 compared to 57, 38 and 67 for this year. This year seems noticeably lower, though the questions were pretty similar.

The troublesome topics are the same: how to represent information in bits and bytes, how many of those would be required, notations like hexadecimal, and generally anything that involves arithmetic in bases other than 10 (and some problems with base 10 too). Many, indeed most, are unsure about how the Toy machine works. A significant number are shaky on how area and volume relate to linear dimensions.

The three values in each column below are for parts 1, 2 and 3, reading from the bottom.



1. (20 points, 2 each) Short Answers. Circle the right answer or write it in the space provided.

(a) Suppose that a group at a (very boring) party is simulating the Towers of Hanoi algorithm with half a dozen disks, and it takes them 10 minutes to perform the moves. If they decide to play again with a dozen disks, approximately how long will it take them?

640 minutes. Easiest to just double from 6 through 7 (20 minutes), 8 (40 minutes), etc. I accepted nearby answers, since the precise time is a bit uncertain.

(b) While cleaning up the mess in my office, I found a shiny thing that was about 8 inches in diameter, very thin and light, and covered with a repeated pattern on one side. Sadly, it was quite brittle and when I dropped it on the floor, it broke into several pieces. What was it?

Integrated circuit wafer. I passed wafers around with an explicit warning about their brittleness. Not a hard disk, nor a chip (which is a small part of a wafer, often about a centimeter on a side).

(c) The ancient core memory that I passed around in class has four 64 by 64 arrays of ferrite cores. How many bytes does the core memory have?

64 128 256 512 1K 2K 4K 8K 16K 32K none of these

2K. 4 * 2^6 * 2^6 / 8. Each core is one bit. Don't forget to divide by 8 to get bytes.

(d) The speeds of supercomputers are measured in floating-point operations per second, or "flops." The fastest computers listed on top500.com have now exceeded a speed of 1 exaflop. Very roughly, how much faster is this than your laptop?

a thousand times as fast **a million times as fast** **a billion times as fast**

a trillion times as fast **a quadrillion times as fast** **no way to estimate**

Billion. Your laptop is roughly 10^9 flops or 1 gigaflop, as noted when we did the estimates of your heartbeats.

(e) If **n** and **m** are integers, how many 1 bits (i.e., bits that have the value 1) are there in the binary representation of the number $\mathbf{4^n / 2^m}$?

1. As noted many times in class and in the Q/A session, any power of 2 has a single 1-bit followed by zeros.

(f) If on my laptop I start a big program like a browser that I haven't used recently, it takes about 10 seconds before it's ready to use. If I then quit the program and 15 seconds later restart it, the startup is much faster. ***In a word or two***, what explains this behavior?

Caching. Demonstrated in class with the Tor browser.

(g) If you wanted to do the most damage to the world's production of integrated circuit chips, what country would you attack?

Taiwan. Not China, not the US. Discussed a couple of times in the context of news stories about Taiwan's role in the semiconductor industry.

(h) Suppose we choose the next American president as the winner of a tournament with everyone in the country participating. Two people flip a coin. If it's heads, the first person advances to the next round and if it's tails the second person advances; the loser is eliminated. (An odd person out gets a bye to the next round.) We do this for everyone in the country. About how many rounds would it take to determine the ultimate winner? [Food for thought: would this be worse than the current scheme?]

28 or 29. Basically what's the log base 2 of 300 million?

(i) Suppose that I have an old program, written in a long-obsolete assembly language, that originally ran on a computer that no longer exists anywhere. Which of these software components would I need so I could run the old program on a current computer?

assembler **compiler** **simulator** **assembler + compiler**

assembler + simulator **compiler + simulator** **all three**

assembler + simulator. Not a high-level language program, so no compiler is involved.

(j) If I want to store the contents of all of the SSDs in the laptops of all current Princeton undergrads, which of these is the smallest that would certainly be big enough to hold it all?

1 GB **1 TB** **1 PB** **1 EB** **1 ZB** **1 YB** **none of these**

1 EB. I also accepted 1 PB, since one could argue that the number was sort of on the edge.

2. (15 points) Playing with Toys

Here is a short program in assembly language for the toy computer used in class, with reminders about what the instructions do:

```
Foo  GET            get a number from keyboard into accumulator
     IFZERO Bar     if accumulator is zero, go to location Bar
     IFPOS  Foo     if accumulator is >= zero, go to location Foo
```

```
          LOAD    Sum    load accumulator with value in location Sum
          ADD     1      add 1 to accumulator
          STORE   Sum    store accumulator in location Sum
          GOTO    Foo    go to location Foo
     Bar  LOAD    Sum
          PRINT
          STOP
     Sum  0              reserve a memory location called Sum, and set its initial value to 0
```

(a) If you run this program and give it the sequence of inputs **6 5 –3 2 1 –4 7 0** what number(s) does it print?

2. Step through one instruction at a time. If the input is zero, print Sum and stop; if the input is positive, get another number; otherwise it's negative so add 1 to Sum and repeat. It's counting the negative numbers. Not very well handled on average, unfortunately.

(b) ***Very briefly***, summarize what it does. Do not just repeat the instructions in words.

Counts negative numbers.

(c) Which of these expressions best describes how the running time of this program depends on **N**, the number of input numbers?

log N **N** **N log N** **N^2** **2^N** **independent of N**

N. It reads each number and processes it in essentially the same number of operations.

(d) The line "**`Sum 0`**" could be moved to one other place in this program and the program would behave identically. Where is that place?

Before the instruction labeled Bar. It can't go at the beginning, since execution starts with the first thing in the program, and that had better be an instruction. Anywhere else will also put the initialization in a place where it will be treated as an instruction.

(e) In a 2007 lecture in McCosh 50, George Dyson (author of *Turing's Cathedral*) quoted von Neumann as saying "Words coding the orders are encoded in the memory just like numbers." Our Toy machine has 10 "orders." If they are encoded in the memory just like numbers, how many bits are needed to encode an instruction?

4. 10 lies between 2^3 and 2^4.

(f) How many *additional* instructions could be added to the Toy's repertoire before another bit is needed in the encoding?

6. 16 instructions fits in 4 bits; 17 requires 5.

3. (55 points, 5 each) Miscellaneous

(a) Suppose that Annual Giving wants to store certain information about alumni donors in *as few bits as possible*. They plan to encode your graduation year (four digits, like 2022), your major (one of 30 departments), and your potential as a donor (in one of eight categories). How many bits will this require per alum?

19. 11 bits for year (0-2047), 5 for department, 3 for potential. The year component was somewhat ambiguous, so I also accepted 22 for people who thought that dates might be as large as 9999 (possible, I suppose). Even though you are not alumni until you graduate, many people took the view that only the current four years were involved and thus only two bits would be needed for that component. Much confusion in evidence, even on the unambiguous aspects.

(b) This course has lots of big numbers and lots of little ones.

(i) How many nanoseconds are there in an exasecond?

10^27

(ii) What power of 2 does this number most closely correspond to?

2^90

Both parts were generally well done.

(c) It is sometimes said that the amount of electrical power needed to perform a computing task falls by half every one and a half years, because newer equipment uses less power to accomplish the same task.

(i) What is the monthly percentage of reduction in the rate of power consumption, assuming a uniform exponential rate of decrease?

4%. If something doubles or halves in 18 months, that's 4% (72/18).

(ii) If a supercomputer uses 10 megawatts for a particular computation today, how many years from now would improved hardware require 10 watts for the same task? (This is pretty improbable, of course.)

30 years. 15 years gets it down to 10 kilowatts and 30 years down to 10 watts. Just powers of two and the corresponding powers of 10.

(d) A specific byte in the RAM of a computer contains the hexadecimal value `0A`. Clearly mark each of the following that this single byte could possibly represent.

The letter A in ASCII	**could represent**	[✓ **could not represent**]
Part of an instruction in the Chrome browser	[✓ **could represent**]	**could not represent**
A cuneiform character in Unicode	**could represent**	[✓ **could not represent**]
The integer value 10	[✓ **could represent**]	**could not represent**
Part of Allegri's *Miserere* in WAV format	[✓ **could represent**]	**could not represent**

(e) A car odometer with 6 decimal digits rolls over to zero after 999,999 miles. Suppose that the odometer in a car works in binary, not in decimal.

(i) If the odometer is 10 binary digits long, what binary value does it show just before it rolls over to zero?

11111 11111

(ii) What is that value expressed in decimal?

1023

(iii) What is that value expressed in hexadecimal?

3FF

I think we talked about this in the Q/A session. The odometer explanation of when numbers wrap around has figured in lectures on multiple occasions.

(f) Suppose that we were to pack Aaron Burr 219 full of laptops, from floor to ceiling. Ignoring chairs, desks, power, wiring, heat dissipation, physical access, the irregular shape of the room, etc., estimate ***very roughly*** how many laptops would fit. You must base your answer on sound estimates and quantitative reasoning. Be brief but clear about your assumptions and computations.

500,000 to a million? If the room is 50 x 40 x 20 feet, that's 40,000 cubic feet. A laptop is about 1 foot by 1 foot by 1/2 inch, so there are about 25 of them in a cubic foot. Your specific numbers didn't matter much, but I did want to see a sensible size for the room and for a laptop, and definitely a realization that the room was three-dimensional. Units conversions for volumes were often wrong: you don't get from cubic feet to cubic inches by multiplying by 12.

(g) Some binary arithmetic:

(i) Add these two binary numbers:

```
110101.01101
  1010.00011
------------
111111.10000
```

Do not convert to decimal and back. No one who tried that got it right.

(ii) Suppose you are adding two **n**-bit binary numbers by hand. Which of these expressions best describes how the amount of work you have to do depends on **n**, the length of each number?

log n **n** **n log n** **n^2** **2^n** **doesn't depend on n**

n. You have to look at each pair of digits. Twice as many digits would take twice as long, the very definition of linear.

(h) A US social security number occupies 9 bytes when stored as ASCII digits.

(i) How many bytes does it occupy if stored as a binary number?

4. SSNs are in the range 0 to 10^9, which is about 2^30 so 30 bits so 4 bytes.

(ii) Eventually the USA will run out of 9-digit social security numbers. If SSNs used 9 hexadecimal digits instead of decimal, how many hexadecimal social security numbers could there be? Express your answer as a power of two

2^36. The number of hex SSNs would be 16^9, which is (2^4)^9.

(i) Quickies:

Ada Lovelace and Alan Turing never met [✓ **true**] **false**

John von Neumann is buried on the grounds of the Institute for Advanced Study **true** [✓ **false**]

Bill Gates and Paul Allen got their start by writing a Basic interpreter [✓ **true**] **false**

Tony Hoare was knighted for creating the first practical Turing Machine **true** [✓ **false**]

"Tap and go" credit cards are powered by a tiny embedded battery **true** [✓ **false**]

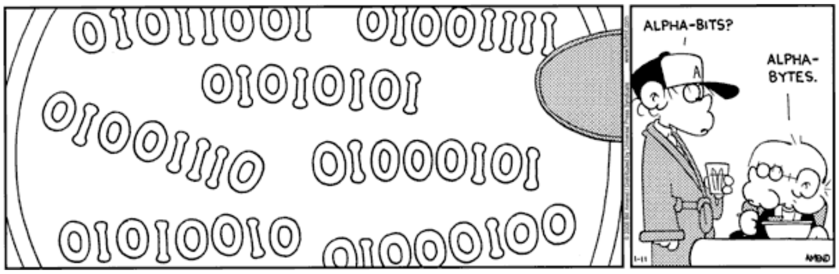
(j) "A common grayness silvers everything" (*Andrea del Sarto*, by Robert Browning). The color "gray" describes any color that has equal amounts of red, green and blue.
(i) In the standard 3-byte representation of RGB colors, how many different shades of gray are there? (Hint: it's not 50.)

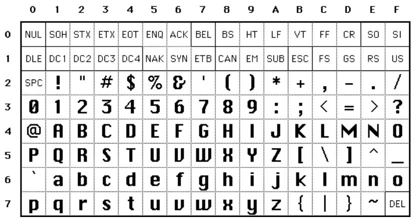
256. "Gray" is equal red, green and blue components. Since there are 8 bits of each, that's 256 distinct shades. I think we talked about this in class one day.

(ii) There are two shades of gray that could be called "medium" gray because they are at the middle of the range of shades. Write out both of these colors in hexadecimal.

7F7F7F, 808080. Maybe easiest to see as the hex equivalents of 127 and 128.

(k) ***Exactly*** what do the alphabytes in Jason's cereal bowl say? Write your answer clearly and ***unambiguously***.





	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SPC	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

YOUNERD. People were a bit sloppy with spaces (there are none) and case (it's all upper case).