Review: What does a computer know?

- Switches!
- What does a switch represent??

- Keep this in mind as you learn the semantics of programming

Visualizing Memory

999	X
998	75.62
…	…
7	STO 005
6	ADD 003
5	RTV 001
4	H
3	-26
2	0.005
1	354
0	-27.2

- Thus, while we might visualize the computer with all sorts of data in the memory slots…

Visualizing Memory

- … It really consists of an arrangement of 1s and 0s

Cell 7	1	1	0	1	1	1	1	0	1	1	1	1	1	1	0	1
Cell 6	1	0	1	1	0	1	1	1	1	1	1	0	1	1	1	1
Cell 5	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1
Cell 4	1	0	1	1	1	0	1	1	1	1	1	1	0	1	1	1
Cell 3	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	1
Cell 2	0	0	1	1	1	1	0	1	1	1	0	1	1	1	0	1
Cell 1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1
Cell 0	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	0

[FIGURE 1.2] A model of computer memory

Visualizing Memory

- So…what does having a 32-bit computer mean?
  - 64-bit computer?

- Notice that they are multiples of 8…the byte!

Computer Units

- The Bit (Binary Digit) – one digit that is either a 1 or a 0
- A Byte – 8 bits
  - When we reserve memory for our programs, they will be in multiples of 8 bits

How big is a byte?

- If a byte is 8 bits, how big is that?

- How many different numbers are stored in a byte?
  - 2^8 = 256 possible ***permutations***

Larger Units of Measure

- 2^{10} bytes = 1024 bytes = 1KB (1 Kilobyte)
- 2^{10} KB = 1024 KB = 1 MB (1 Megabyte)
- 2^{10} MB = 1024 MB = 1 GB (1 Gigabyte)
- 2^{10} GB = 1024 GB = 1 TB (1 Terabyte)

- Know what comes next?

- Lets say we have 2MB. How many bytes is that?

But we are still missing something

- How does the computer translate from groups of 0's and 1's to something more meaningful?
  - Binary number system!
- For today, we will talk about how binary numbers are translated into ***unsigned integers***

Binary Numbers

- The “switch” nature of transistors make storing numbers in binary a natural fit.
- Binary is a change of base for our number system, base 2
- In a number, its position represents powers of 2

Numeric representation

- We usually work with decimal numbers with digits from 0 to 9 and powers of 10

  7313 = (7 * 1000 + 3 * 100 + 1 * 10 + 3 * 1)

  Or $(7 * 10^3 + 3 * 10^2 + 1 * 10^1 + 3 * 10^0)$

- The binary number system uses digits 0 and 1 and powers of 2

  0101 = (0 * 8 + 1 * 4 + 0 * 2 + 1 * 1)

  Or $(0 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0)$

  = 5

Thinking about this…

- A ***nibble*** is half a byte
  - 0110 is one example
- How many numbers can a nibble store?
- Which numbers can a nibble store?
  - Largest nibble/smallest nibble?

Your turn #1

- What unsigned decimal numbers are represented by the following binary numbers?

Example: 00000101 = 5

01000100

00001101

10110011

Your turn #2

- How would you write the following numbers in binary?

  Example: 14 = 8 + 4 + 2 -> 00001110

  3

  121

  143

Encoding

- Binary numbers can represent more things than just integers
- Another example is ASCII
  - American Standard Code for Information Interchange
  - Character encoding scheme based on the English alphabet
  - http://en.wikipedia.org/wiki/ASCII

Programming in Python

- We will be using Python3
  - Install on your own computers or use one of the ITTC or Wright Hall labs
  - Can install from http://www.python.org
  - Walkthrough

- Be sure to use **version 3** (not version 2)!

Problem Solving

- How do humans solve problems?
  - Once we know that, how do we translate that to how a computer would solve a problem?
- Get in groups of two or three

Problem #1

- Suppose that on a recent trip, you left Cedar Falls with a full tank of gas and your odometer read 64783 miles.
- When you returned, your odometer read 64969 miles. You refilled your gas tank with 8 gallons of gas.
- What was your mileage per gallons (or MPG)?

Problem #1

- What is the answer?
- How did you arrive at this specific answer?
- What is the general purpose algorithm to solve this class of problem?

Problem #1

- Let's try implementing the solution in Python!