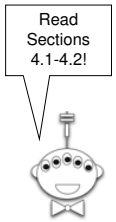
The CS 5 Times

CS 5 Penguin Prepares Revenge

The CS5 penguin arranges revenge.

Claremont (AP): After suffering unmentionably rude treatment at the trailing end of a physics professor's dog, the CS5 penguin filed a formal complaint with the HMC administration, according to a lowly placed source. "This friendly rivalry has gone too far, and we demand justice!" complained one professor. A march in support of the penguin is planned at Pitzer College this evening. "We're not quite sure what happened," explained an incensed Pitzer student, "but we stand ready to protest anything at any time."

Meanwhile, the CS 5 penguin repaired to a local bar. Fellow penguins familiar with the incident explained that he was preparing himself to return the indignity in kind.



Computer Organization

(Or "How Computers Really Work!")

This week…

1. How data is represented in a computer
2. How computers do arithmetic
3. Building digital circuits!

Then: Building digital circuits
From circuits to a computer!

And: Programming the computer in its own "machine language"!

Today: Representing Numbers

1. Representing numbers in different bases
2. Converting between bases
3. Arithmetic in different bases
4. Clever Russian peasants!

Art courtesy of www.auburn.edu/academic/liberal_arts/foreign/russian/art/goncharova-peasants.jpg

Representing Numbers

What is the number 4312?

10^3	10^2	10^1	10^0
4	3	1	2

What is this number in base 20?

20^2	20^1	20^0
1	3	2

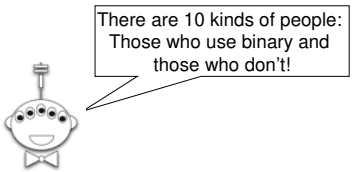
Now we're using powers of 20

Olmec number representation in base 20 (East Mexico 1200 BC-600 AD)

Olmec relief from http://www.meta-religion.com

Base 2

2^3 2^2 2^1 2^0 ⟸ Now we're using powers of 2



Arbitrary Bases (base "*b*")

When using base *b,* the digits permitted are:

What is 5 in…

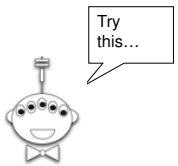
- base 2?
- base 3?
- base 4?
- base 5?
- base 6?
- base 42?

Counting in Base *b*

Count from zero to six in each of the following bases:

Base 2:

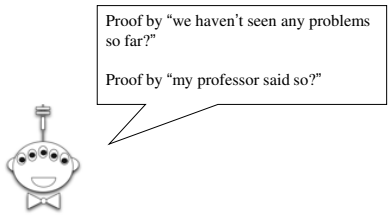
Base 3:



What's the "algorithm" for counting in a general base *b?*

Mathematical Aside

How do we know that all non-negative integers can really be uniquely represented in a given base *b?*



Is There Such a Thing as Base 1?

Unary!

1^3 1^2 1^1 1^0 ⟸ Now we're using powers of 1 (Weird!)

Are we going to use 0 as our only digit?

Comparing Representations in Different Bases

Consider the number 10^9 in base 1, 2, 3, 10, and 20:

Base 1: 111111111111111111111111111111111111111111...

At 10 "1's" per inch, this will be 1578 miles long!

Base 2: 111011100110101100101000000000

Base 3: 2120200200021010001

Base 10: 1000000000

Base 20: FCA0000

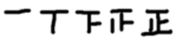
FCA?

Oh! Friendly Cuddly Alien

A Brief History of Bases

Unary: Used since at least 400 BCE

Europe, New Zealand North America



China, Japan, Korea

Base 60 ("Sexagesimal"): Sumerians in Mesopotamia (Iraq) around 300-400 BCE

Base 20 ("Vigesimal"): Olmec and other Mesoamerican cultures - 3000 year period before Columbus arrives in the Americas

Base 8 ("Octal"): Yuki Tribe of Northern CA, Pamean Mesoamerican culture

Members of the Yuki Tribe c. 1858 (from wikipedia.org)

Two "Special" Bases: 2 and 10

Base 10: Elamites in Iran use early form of base 10 system around 3500 B.C.E.

Base 2: References to base 2 appeared in the *I Ching*.

Computers are "simple".
Base 2 is the simplest reasonable base.
Therefore, computers use base 2!

Converting Between Bases

The digits 0 and 1 are referred to as "bits"—that's short for "binary digits"

Convert 1101_2 to base 10

Convert 25_{10} to base 2

Worksheet

The "Power" of Shifting!

"Left Shifting"

10^3 10^2 10^1 10^0

4 2

4 2 0

2^3 2^2 2^1 2^0

1 1 = 3_{10}

1 1 0 = ?

"Right Shifting"

10^3 10^2 10^1 10^0

4 5 7

4 5

2^3 2^2 2^1 2^0

1 0 1 = 5_{10}

1 0 = ?

Base Conversion, Part Deux

2^4 2^3 2^2 2^1 2^0

1

$25_{10} = ?_2$

Addition

Base 10 Addition

10^2 10^1 10^0

4 3

\+ 8 9

Multiplication

Base 10 Multiplication

	10^2	10^1	10^0
	3	4	1
×	1	0	2
	6	8	2
	0	0	0
+	3	4	1
	3	4	7	8	2

Base 2 Multiplication

	2^2	2^1	2^0
	1	1	1
×	1	0	1

Aside: Multiplication with Russian Peasants

Compute `21 × 6:`

21	6
10	12
5	24
2	48
1	96

6+24+96 = 126

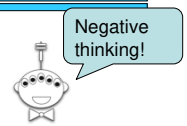
(Translation: "Why does this work?")

Negative Numbers

(with the nifty "two's complement" method)

- Assume that we have only 8 bits to represent numbers
- If we try to increment `11111111` by 1, what happens?
- `00000011` represents 3_{10}. What property should the representation of -3_{10} have so that arithmetic with positive and negative numbers works nicely?

Worksheet…



In two's complement (with 3 bits to keep things simpler)…

- What's the negative of 0?
- How is -1 represented?
- What's the largest positive number that can be represented?
- What's the smallest negative number that can be represented?
- Does addition work as expected?
- Is a double negative a positive?