Data Representation

Interpreting bits to give them meaning

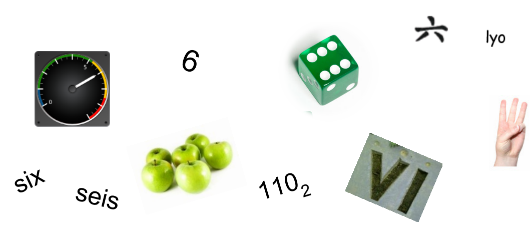
Part 1: Numbers

Notes for CSC 100 - The Beauty and Joy of Computing
The University of North Carolina at Greensboro

What is a number?

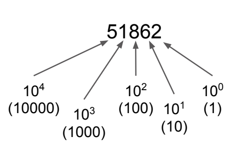
Question: You've been working with numbers (almost) all your life - what *are* they?

Example: What is the number 6?



Decimal Representation

Most common written representation of numbers is "decimal notation":



"Representation" is the converse of "Abstraction"

Makes abstractions concrete

Question: Why powers of ten?
Equivalently, why are there 10 different digits?

Decimal Representation

How can we mathematically extract digits?

51862

Divide by 10:

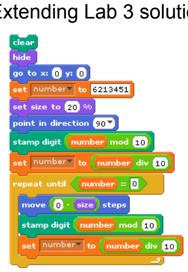
Quotient gives all but last digit

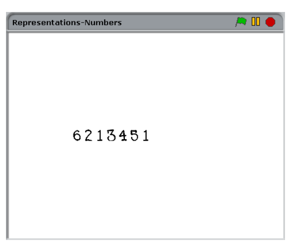
Remainder gives last digit

Using the "div" block from Lab 3

Stamping out decimal representation

Extending Lab 3 solution to stamp out all digits (not just 2):





Now let's see this in action...

Binary Representation

The powers used in the representation (also, number of different "digits") is called the *base*.

- "Decimal" is base 10
- "Binary" is base 2

This number is written in binary

1 0 0 1 1

Instead of "digits" have "bits"

2^4 (16) 2^3 (8) 2^2 (4) 2^1 (2) 2^0 (1)

Numbers below the dashed line are all written in decimal (because they're for our benefit).

$1 * 2^4 + 0 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 = 16 + 2 + 1 =$ *19*

Converting decimal to binary

Just like the BYOB code, we keep dividing by the base (2), recording remainders and keeping quotients.

Operation	*Quotient*	*Remainder*
43 / 2	21	1
21 / 2	10	1
10 / 2	5	0
5 / 2	2	1
2 / 2	1	0
1 / 2	0	1

First bit found is last bit in binary representation.

1 0 1 0 1 1

Using subscripts to denote base:
$43_{10} = 101011_2$

Converting decimal to binary

Just like the BYOB code, we keep dividing by the base (2), recording remainders and keeping quotients.

Operation	*Quotient*	*Remainder*
43 / 2	21	1
21 / 2	10	1
10 / 2	5	0
5 / 2	2	1
2 / 2	1	0
1 / 2	0	1

1 0 1 0 1 1

Using subscripts to denote base:
$43_{10} = 101011_2$

Practice problems:

1_{10} = ________ $_2$
6_{10} = ________ $_2$
8_{10} = ________ $_2$
12_{10} = ________ $_2$
23_{10} = ________ $_2$
31_{10} = ________ $_2$

Converting binary to decimal

Keep a position and a value, and at each step move position to right, multiply value by 2 and add the new bit.

Start position: Leftmost bit *Start value: 1*

101101	1
101101	1*2 + 0 = 2
101101	2*2 + 1 = 5
101101	5*2 + 1 = 11
101101	11*2 + 0 = 22
101101	22*2 + 1 = 45

So $101101_2 = 45_{10}$

Some terminology:

Leftmost bit is "most significant bit" or "msb"

Rightmost bit is "least significant bit" or "lsb"

Converting binary to decimal

Keep a position and a value, and at each step move position to right, multiply value by 2 and add the new bit.

Start position: Leftmost bit *Start value: 1*

101101	1
101101	1*2 + 0 = 2
101101	2*2 + 1 = 5
101101	5*2 + 1 = 11
101101	11*2 + 0 = 22
101101	22*2 + 1 = 45

So $101101_2 = 45_{10}$

Practice problems:

$11_2 =$ ____$_{10}$

$1001_2 =$ ____$_{10}$

$11011_2 =$ ____$_{10}$

$10001_2 =$ ____$_{10}$

$11111_2 =$ ____$_{10}$

$101011_2 =$ ____$_{10}$

Counting in binary without converting

Picture an odometer with only two values, 0 and 1

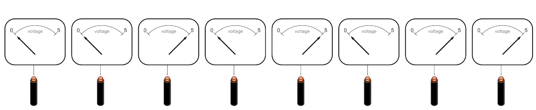
When any wheel goes from 1 to 0, turn the one to the left

0000	$= 0_{10}$	1000	$= 8_{10}$
0001	$= 1_{10}$	1001	$= 9_{10}$
0010	$= 2_{10}$	1010	$= 10_{10}$
0011	$= 3_{10}$	1011	$= 11_{10}$
0100	$= 4_{10}$	1100	$= 12_{10}$
0101	$= 5_{10}$	1101	$= 13_{10}$
0110	$= 6_{10}$	1110	$= 14_{10}$
0111	$= 7_{10}$	1111	$= 15_{10}$

Why binary?

In electronics, you can measure voltages on wires

- Consider 8 wires
- Each with at either 0 volts or 5 volts



Interpreting 0V as 0, and 5V as 1, get: 00101011_2 ($= 43_{10}$)

Voltages can turn on/off switches to create logic circuits

Hexadecimal - another useful base

Hexadecimal is base 16.

How do we get 16 different digits? Use letters!

Hexadecimal digits (or "hex digits" for short):
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Counting - now our odometer has 16 digits:

0_{16} (= 0_{10})	6_{16} (= 6_{10})	C_{16} (= 12_{10})	12_{16} (= 18_{10})	...
1_{16} (= 1_{10})	7_{16} (= 7_{10})	D_{16} (= 13_{10})	13_{16} (= 19_{10})	
2_{16} (= 2_{10})	8_{16} (= 8_{10})	E_{16} (= 14_{10})	14_{16} (= 20_{10})	
3_{16} (= 3_{10})	9_{16} (= 9_{10})	F_{16} (= 15_{10})	15_{16} (= 21_{10})	
4_{16} (= 4_{10})	A_{16} (= 10_{10})	10_{16} (= 16_{10})	16_{16} (= 22_{10})	
5_{16} (= 5_{10})	B_{16} (= 11_{10})	11_{16} (= 17_{10})	17_{16} (= 23_{10})	

Hexadecimal/Decimal Conversions

Conversion process is like binary, but base is 16

Problem 1: Convert 423_{10} to hexadecimal:
423/16 = quotient 26, remainder 7 (=7_{16})
26/16 = quotient 1, remainder 10 (=A_{16})
1/16 = quotient 0, remainder 1 (=1_{16})

- Reading digits bottom-up: 423_{10} = $1A7_{16}$

Problem 2: Convert $9C3_{16}$ to decimal:
Start with first digit, 9
9*16 + 12 = 156
156*16 + 3 = 2499

- Therefore, $9C3_{16}$ = 2499_{10}

Hex Digit List
$0_{16} = 0_{10}$
$1_{16} = 1_{10}$
$2_{16} = 2_{10}$
$3_{16} = 3_{10}$
$4_{16} = 4_{10}$
$5_{16} = 5_{10}$
$6_{16} = 6_{10}$
$7_{16} = 7_{10}$
$8_{16} = 8_{10}$
$9_{16} = 9_{10}$
$A_{16} = 10_{10}$
$B_{16} = 11_{10}$
$C_{16} = 12_{10}$
$D_{16} = 13_{10}$
$E_{16} = 14_{10}$
$F_{16} = 15_{10}$

Hexadecimal/Decimal Conversions

Conversion process is like binary, but base is 16

Problem 1: Convert 423_{10} to hexadecimal:
423/16 = quotient 26, remainder 7 (=7_{16})
26/16 = quotient 1, remainder 10 (=A_{16})
1/16 = quotient 0, remainder 1 (=1_{16})

- Reading digits bottom-up: 423_{10} = $1A7_{16}$

Your turn! Convert:
103_{10} = ________ $_{16}$
247_{10} = ________ $_{16}$
952_{10} = ________ $_{16}$

$3C_{16}$ = ________ $_{10}$
$B9_{16}$ = ________ $_{10}$
357_{16} = ________ $_{10}$

Problem 2: Convert $9C3_{16}$ to decimal:
Start with first digit, 9
9*16 + 12 = 156
156*16 + 3 = 2499

- Therefore, $9C3_{16}$ = 2499_{10}

Hex Digit List
$0_{16} = 0_{10}$
$1_{16} = 1_{10}$
$2_{16} = 2_{10}$
$3_{16} = 3_{10}$
$4_{16} = 4_{10}$
$5_{16} = 5_{10}$
$6_{16} = 6_{10}$
$7_{16} = 7_{10}$
$8_{16} = 8_{10}$
$9_{16} = 9_{10}$
$A_{16} = 10_{10}$
$B_{16} = 11_{10}$
$C_{16} = 12_{10}$
$D_{16} = 13_{10}$
$E_{16} = 14_{10}$
$F_{16} = 15_{10}$

Hexadecimal/Binary Conversions

Exactly 16 hex digits, and exactly 16 4-bit binary numbers

Converting between hex and binary is easy - 4 bits at a time:

Problem 1: Convert 01110100110_2 to hexadecimal

011 | 1010 | 0110

3 | A | 6

Answer: $3A6_{16}$

Problem 2: Convert $D49_{16}$ to binary

D | 4 | 9

1101 | 0100 | 1001

Answer: 110101001001_2

Hex Digit List

$0_{16} = 0000_2$
$1_{16} = 0001_2$
$2_{16} = 0010_2$
$3_{16} = 0011_2$
$4_{16} = 0100_2$
$5_{16} = 0101_2$
$6_{16} = 0110_2$
$7_{16} = 0111_2$
$8_{16} = 1000_2$
$9_{16} = 1001_2$
$A_{16} = 1010_2$
$B_{16} = 1011_2$
$C_{16} = 1100_2$
$D_{16} = 1101_2$
$E_{16} = 1110_2$
$F_{16} = 1111_2$

Hexadecimal/Binary Conversions

Exactly 16 hex digits, and exactly 16 4-bit binary numbers

Converting between hex and binary is easy - 4 bits at a time:

Problem 1: Convert 01110100110_2 to hexadecimal

011 | 1010 | 0110

3 | A | 6

Answer: $3A6_{16}$

Problem 2: Convert $D49_{16}$ to binary

D | 4 | 9

1101 | 0100 | 1001

Answer: 110101001001_2

Your turn! Convert:

12_{16}
$B1_{16}$
FF_{16}
48_{16}

1010101_2
10100110_2
00010100_2
01101110_2

Hex Digit List

$0_{16} = 0000_2$
$1_{16} = 0001_2$
$2_{16} = 0010_2$
$3_{16} = 0011_2$
$4_{16} = 0100_2$
$5_{16} = 0101_2$
$6_{16} = 0110_2$
$7_{16} = 0111_2$
$8_{16} = 1000_2$
$9_{16} = 1001_2$
$A_{16} = 1010_2$
$B_{16} = 1011_2$
$C_{16} = 1100_2$
$D_{16} = 1101_2$
$E_{16} = 1110_2$
$F_{16} = 1111_2$

Use of hexadecimal in file dumps

Binary is a very long format (8 bits per byte), but often data files only make sense as binary data. Hexadecimal is great for this - simple one-to-one correspondence with binary, and more compact.

Sample "file dump":

```
0000000: ffd8 ffe1 35fe 4578 6966 0000 4949 2a00  ....5.Exif..II*.
0000010: 0800 0000 0b00 0e01 0200 2000 0000 9200  .......... .....
0000020: 0000 0f01 0200 0600 0000 b200 0000 1001  ................
0000030: 0200 1900 0000 b800 0000 1201 0300 0100  ................
0000040: 0000 0600 0000 1a01 0500 0100 0000 d800  ................
0000050: 0000 1b01 0500 0100 0000 e000 0000 2801  ..............(.
0000060: 0300 0100 0000 0200 0000 3201 0200 1400  ..........2.....
0000070: 0000 e800 0000 1302 0300 0100 0000 0200  ................
0000080: 0000 6987 0400 0100 0000 fc00 0000 2588  ..i...........%.
0000090: 0400 0100 0000 2413 0000 f213 0000 2020  ......$.......
00000a0: 2020 2020 2020 2020 2020 2020 2020 2020
00000b0: 2020 2020 2020 2020 2020 2020 2000 4361                .Ca
00000c0: 6e6f 6e00 4361 6e6f 6e20 506f 7765 7253  non.Canon PowerS
00000d0: 686f 7420 5358 3233 3020 4853 0000 0000  hot SX230 HS....
00000e0: 0000 0000 b400 0000 0100 0000 b400 0000  ................
00000f0: 0100 0000 3230 3131 3a30 373a 3134 2031  ....2011:07:14 1
0000100: 353a 3039 3a32 3700 2100 9a82 0500 0100  5:09:27.!.......
0000110: 0000 8e02 0000 9d82 0500 0100 0000 9602  ................
0000120: 0000 2788 0300 0100 0000 6400 0000 3088  ..'.......d...0.
```

Position in file — Actual binary data (written in hexadecimal) — The same data, showing character representation

Remember....

Don't get lost in the details and manipulations:

Any base is a representation of an abstract number

We are interested in working with the number, and computations are not "in a base" - the base is only useful for having it make sense to us or the computer

Practice!

You should be able to convert from one base to another.

Lots of ways to practice:

- By hand: Pick a random number convert to binary and convert back - did you get the same value?
  - This isn't foolproof: You could have made two mistakes!
- With a calculator: Many calculators (physical and software) do base conversion - check your randomly selected conversions.
- With a web site: Several web sites provide says to practice
  - For example, see http://cs.iupui.edu/~aharris/230/binPractice.html

Next: Other Representations

Now we know all about representing numbers

But computers also deal with text, web pages, pictures, sound/music, video, ...

How does that work?