Do you speak binary?

Coding Basics

Announcement

- First homework is being graded, grade to be posted by the end of the week
- Revisit homework policy: late submission, request regrading etc.
- Worst quiz and worst homework grade to be dropped.
- However, use this policy wisely!
- A few of you (~4) have not submitted your 1st homework, if you missed the homework submission because you registered for this class at a very late time, please talk to me after class.

How many of you have programming experience prior to taking CS 1060?

Important: think about how YOU have progressed over the course of this class.

COMPUTATIONAL Thinking

Computational Thinking

inputs → algorithms → outputs

Think like a computer scientist!

PROBLEM SOLVING

Binary, Number Bases & converting between bases

Binary
0,1

Decimal

0, 1, 2, 3, …, 9

Why binary for computers?

- Computer use binary – digits 0 and 1 – to store data
- A binary digit, or bit, is the smallest unit of data in computing
- Circuits in a computer's processor are made up of transistors
- The digits 1 and 0 reflect the on and off states of a transistor
- Computer programs get translated into binary machine code for a processor to execute

Advantages of using binary

Claude Shannon, Bell Lab, 1948 paper: "*A Mathematical Theory of Communication*"

- Binary devices are simple and easy to build: e.g. digital calculator
- Binary signals are unambiguous (noise immunity).
- Flawless copies can be made of binary data.
- Anything that can be represented with some sort of pattern can be represented with patterns of bits.

Credit: Bradley Kjell

Decimal

100	10	1
1	2	5
1x100 +	2x10 +	5x1

binary

4	2	1
1	0	0

1x4 + 0x2 + 0x1 = 4 (decimal)

binary

4	2	1
0	0	0

binary

4	2	1
0	0	1

binary

4	2	1
0	1	0

binary

4	2	1
0	1	1

binary

4	2	1
1	0	0

binary

4	2	1
1	0	1

binary

4	2	1
1	1	0

binary

4	2	1
1	1	1

Algorithm:
base-2 to base-10

What number does 10010110 in base 2 represent?

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
1	0	0	1	0	1	1	0

1x128 + 1x16 + 1x4 + 1x2 = 150

What number does 10010110 in base 2 represent?

Using 0-based indexing

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
1	0	0	1	0	1	1	0
D0	D1	D2	D3	D4	D5	D6	D7

1x128 + 0x64 + 0x32 + 1x16 + 0x8 + 1x4 + 1x2 + 0^1 = 150

D0 x 2^(7-0) + D1 x 2^(7-1) +....+Di x 2^(7-i) +...+D7 x 2^(7-7) = 150

What number does 10110 in base 2 represent?

Using 0-based indexing

2^4	2^3	2^2	2^1	2^0
16	8	4	2	1
1	0	1	1	0
D0	D1	D2	D3	D4

1x16 + 0x8 + 1x4 + 1x2 + 0^1 = 22

D0 x 2^(4–0) + D1 x 2^(4–1) + D2 x 2^(4–2) + D3 x 2^(4–3) + D4 x 2^(4–4)

Di x 2^(4–i)

Algorithm: base-2 to base-10

An algorithm is a precise set of steps to solve a problem

1. Input: a binary number with digits D0 D1 D2 Dn-1.
2. Initialization: set Sum = 0, i = 0
3. While (i is less than the number of digits)
   a. Add Di * (2^(n-1-i)) to Sum
   b. Increment i
4. Output Sum

The corresponding Python code

```
D=raw_input ('Enter binary # to be converted: ')
n=len(D); sum=0; i=0
while (i<n):
    sum=sum+int(D[i])*2**(n-i-1)
    i=i+1
print 'The decimal # of the given binary # is', sum
```

http://www.tutorialspoint.com/execute_python_online.php

The corresponding Python code explained

```
D=raw_input('Enter binary #: ')  # raw_input([prompt message]) is a build-in function: it reads a line from input, converts it to a string and returns it.

n=len(D); sum=0; i=0 # initialization, len([string]) another build-in function, it returns the length of an object

while (i<n): #while loop statement

    sum=sum+int(D[i])*2**(n-i-1) # summing up, int([number/string]) returns an integer object from a number or string

    i=i+1 #increament

print 'The decimal # of the given binary # is', sum # print both string and number, print the converted decimal #
```

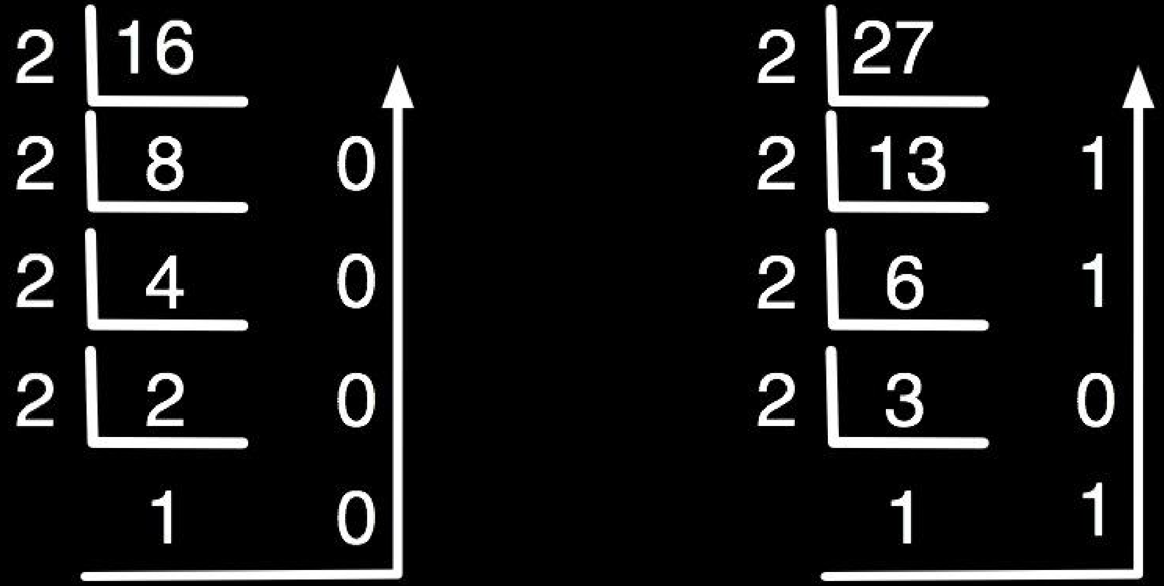
https://docs.python.org/2/library/functions.html#

A SIMPLER VERSION USING BUILD-IN FUNCTIONS

```
binary=raw_input('Enter binary #: ')
decimal=int(binary, 2)
print 'The decimal # of the given binary # is', decimal
```

Algorithm:
base-10 to base-2

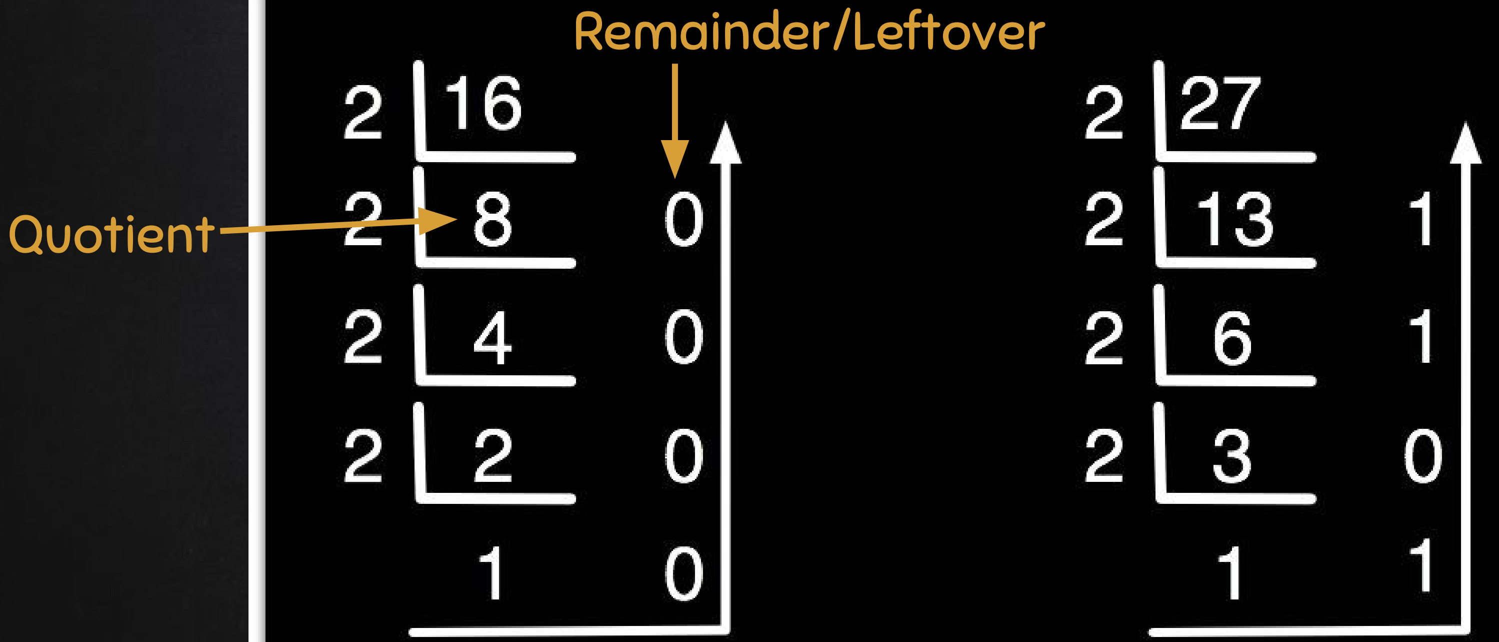
Algorithm by examples



Converting 16 (base-10) to base-2: 10000

Converting 27 (base-10) to base-2: 11011

Algorithm by examples



Converting 16 (base-10) to base-2: 10000

Converting 27 (base-10) to base-2: 11011

Algorithm: base-10 to base-2

1. Input: a decimal number dec
1. Initialization: set s = 0, i = 1
2. While (dec > 0)
   a. remainder = dec % 2
   b. divide dec by 2
   c. append remainder to the left of s, i.e., multiplying by 10 and add to s
3. Output s

In Python

```
dec=input("Enter decimal # to be converted: ")
s=0; i=1
while dec>0:
    remainder=dec%2
    dec=dec/2
    s=s+(i*remainder)
    i=i*10
    print s
print "The binary of the given # is ",s
```

Exercises

1. What is 1000 in base 2 converted to base 10?

2. Convert 36 in base 10 to base 2.

Python programming is fun and productive

Quiz 2: binary and decimal

Think beyond Binary

Quantum Computing

- Theoretical computation systems: quantum computers, use quantum-mechanical phenomena to perform operations on data
- Different from digital electronic computers based on transistors.
- Uses quantum bits (qubits), which can be in superpositions of states: e.g. linear combination of basic states of particles
- Quantum Superposition: any 2+ quantum states can be added together and the result will be another valid quantum state
- Quantum Turing machine or the universal quantum computer
- Non-deterministic and probabilistic
- Paul Benioff, Yuri Manin 1980; Richard Feynman 1982; David Deutsch in 1985.
- Further reading: https://en.wikipedia.org/wiki/Quantum_computing

Quantum Computing

- A quantum bit corresponds to a single electron in a particular state. Using the trajectories of an electron through two closely spaced channels for encoding.
- In principle, two different states are possible: the electron either moves in the upper channel or in the lower channel – a binary system.
- However, a particle can be in several states simultaneously, that is, it can quasi fly through both channels at the same time.
- These overlapping states can form an extensive alphabet of data processing.
- Quantum computer science
- Further reading: http://qist.lanl.gov/qcomp_map.shtml
- http://www.webpronews.com/quantum-computing-beyond-binary-2012-03/

THANKS!

Any questions?

You can find me at
beiwang@sci.utah.edu

http://www.sci.utah.edu/~beiwang/teaching/cs1060.html

Credits

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by SlidesCarnival
- Photographs by Unsplash