CS125 Chronicles Section 2 **"Binary Gangland"**

Pair up with one other student. Your job is to infiltrate the infamous CS binary gang. We have 5 challenges for you. Each should take about 10 minutes.

Challenge #0 “Snakes and Ladders” (easier than it appears)

You have N (an unknown number) of students (aka memory locations). Each student can store one number. I’ve secretly already given each student a number – the address of the next student to visit by the TA (imagine a snakes and ladders game). The TA wants to count the number of students they visit before revisiting a student. The TA can change the student’s number. Here is what the TA does : 1) TA keeps track of the address to visit next 2) If TA comes across -1 stored in a student, he/she stops and prints out the total number of students 3) Otherwise he/she just visits the next address (determined in step 1). There are two errors in this sequence of steps. What are they and how will you fix them?

Addr	Value
…	…
7	2
6	4
5	7
4	3
3	2
2	6
1	9
0	5

Challenge #1 “Representation”

You have five fingers in one hand. The thumb is worth 1, the index finger 2, the other fingers are 4, 8 and the pinkie (smallest), 16. With this binary system you can represent the integers (starting from zero and counting up). e.g. thumb+index+middle+pinkie : 1+2+4+16=23.

Q.1 Using this base 2 counting system what is the maximum number you can show?

Extension #1 Representing signed (Negative) numbers:

Q.2 If the last digit, the pinkie, was worth a negative of it’s original value (i.e. -16), what range of numbers (min to max) could you now represent?

Q.3 How would you represent the number -1?

Extension #2 Look Ma! Two hands *(You can ignore Extension #1 - can you see why?)*

Q.4 If you used 8 fingers how many different numbers can you represent?

Q.5 If you used thumbs too (ie. ten binary digits), how many numbers can you represent now?

Extension #3 Fixed Point Arithmetic

Q.6 Suppose you used five fingers but made them valued as follows: thumb = ¼, index finger= ½, other fingers 1, 2 and 4.

i) What are the smallest and largest numbers in this *unsigned representation*?

ii) Your friend uses a *signed representation*: {(thumb)¼, ½, 1, 2, -4(pinkie)}. What are the smallest and largest numbers they can represent using negative pinkies?

Fixed Point too easy for you? How about some advanced Geek Guesswork?

iii) From it's name, can you guess what a “*floating* point” representation might be like?

Cryptic hint: It encodes two numbers to represent a real number!

A UofI CS Experience by Lawrence Angrave Jan 2011 (Revised Aug 2011).
Image attribution: http://scienceblogs.com/goodmath/2006/10/binary_fingermath.php
Image attribution: http://www.etsy.com/listing/50088757/binary-code-geek-sterling-silver-hand
Binary parity game adapted from CS Unplugged.

Challenge #2 Secret messages (what did their fingers say?)

Write out numbers 0 to 31 (one per row) and then the equivalent in base 2 using 5 binary digits. When you write down the base 2 number include the zeros in front e.g. zero should be written as 00000. Then map a letter to the binary sequence: “A” to 00001, “B” to 00010, “C” to 00011, …

00000
00001 A
00010 B
00011 C
00100 D
00101 E

Q.7 **Decode**
i) the secret handshake: Fingers: 00011, 10011, 10010, 01111, 00011, 01011, 10011

and ii) the binary ASCII dog tag - see the picture above - only use the lowest 5 bits of each byte.

Challenge #3 Adding – Give the truth in a table

Q.8 Your binary adding skills will be tested. Add the two binary (base 2) numbers (without converting them to back to base 10) by adding each column at a time. Show the carry!

Example
```
  0001.                 0101 0111.
+ 1011.               + 0011 1001.
  1100.               ____________.
```
Carry: 1 1

Q.9 Can you simplify adding to a simple set of 8 rules? Is it possible to perform addition without knowing that you’re adding!? For each vertical position there’s one of 8 configurations: There’s either a one or zero from the first number (A), a one or zero from the second number (B) and possible a carry from the previous digit. For each configuration calculate the output and carry. This is called a truth table.

A	B	C	Output	Carry for next digit
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Challenge 4 “I demand binary parity!”

for this challenge join-up with another pair of students

Using the key from challenge #2 write down the word “MAGIC” in binary, one letter per line. Now write the 5 binary lines using the flip cards to make a 5x5 square. The TA will now add 11 more cards, to make a 6x6 square.

You send your secret message to another gang member but there’s a small error in transmission: Secretly flip one of the 36 cards. Can the TA determine which square (bit) was flipped?

Q.10 How do they do this? What algorithm are they using?

Q.11 Extension. Suppose two bits were flipped. What is the likelihood that you could
   i) detect that a transmission error had occurred?
   ii) fix it?