Assignment 2

— Number Conversions —

Binary Problems:

- Convert the following numbers from binary to ints:

  00001011 -> 1 + 2 + 8 -> 11

  00001100 -> 4 + 8 -> 12

  00011110 -> 2 + 4 + 8 + 16 -> 29

  00000101 -> 1 + 4 -> 5

- Convert the following from ints to binary:

  31 -> 00011111

  20 -> 00010100

  1000 -> 00000011 11101000

  12 -> 00001100

- How many possible numbers can be represented in X bits?

  2 ^ X are the numbers that can be represented by X bits. 1 bit = 2 ^ 1 (so 0 or 1) 2 bits = 2 ^ 2 (so 0 or 1 or 2 or 3) is 4 total.

- In a signed system, what is the range of positive numbers and negative numbers that can be represented by X bits?

  In a signed system, you can reach the range -(2^{x-1}) - 1 to (2^{x-1}) - 1.

  Ex: -127 to 127

- In an unsigned system, what is the range of positive numbers that can be represented by X bits?

  In an unsigned system you can reach a range of 0 to (2^{x}) -1

  This means there are no negative numbers allowed in this sytem because we don't have a bit to sign it.

Specify an Algorithm

- Write the specific steps for a robot to convert a positive decimal (base 10) number less than (not equal to!) 256 to a binary (base 2) number. This robot can carry out simple instructions and mathematical operations, but it does not have any memory!!! The robot is given the decimal number on one index card, and it must produce the binary number on the blank index card. You may use more index cards if you need, but you must have at least two. These index cards can serve as devices for input and output, as well as for memory locations. You must be very explicit! Your algorithm must specify enough details from getting the decimal number to writing the binary number on a piece of paper! The TAs will test this algorithm by being the robot!!! ☺️

  1. Prepare the input card to be read.
  2. If the input card is greater than 255, write -1 on a new card and stop.
  3. Take out a second card, and write '128' on it
  4. Take a third card and prepare to write on it.
  5. If our input card is greater than or equal to our second card, write a 1 on the third card, then subtract the second card from the input card and replace the value of our input card with the new number. If not, write a 0 on the third card.
  6. Replace the number on the second card with a number half it's size. 128 -> 64
  7. Repeat steps 4 and 5 a total of 8 times until our second card has a value of 1 and the output card has 8 characters.

Metacognative Questions:

- What are bad decimal numbers?

Any decimal number that is out of range of the given byte is considered a bad decimal number. Any bit that isn't signed can be larger, but not negative.

- What will your robot do with bad decimal numbers?

  It rejects it and returns -1 to indicate a failure.

- How would you change your algorithm to handle bad decimal numbers?

  I would change it so that it can encompass larger numbers or negative numbers.

- How would your algorithm change if we didn't know the maximum decimal number?

  I don't think my algorithm would change, it would just error out if the number is too large. I would probably require more space to calculate things with more intermediate cards or a larger output / input card. In this example it would take more time because writing down things take awhile.