CSCA48 Exercise 4

Due: February 9, 2014. 5:00pm

In a file called `ex4.py`, complete the following functions. All functions must be `recursive`, and only the `Easy` version of each function will be marked. (The `Hard` versions are just for fun).

gcd(a, b):

Easy: Compute the greatest common denominator of `a` and `b`. Hint: `gcd(a, 0) = a` and if `a < b`, then `gcd(a, b) = gcd(a, b-a)`.
Hard: Find a method for calculating gcd that is more efficient than the one listed above (mine can find `gcd(12345678, 87654321)` in fewer than 5 steps).

search(L,s):

Easy: Given a nested list L (such that each element of L is either an integer, or a list, which may itself contain integers, or lists, which may in turn.... etc) return True iff s is in L.
e.g., `search([1, [2, 3], 4, [5, [6 , [], [8, 9]], 10]], 8)` should return `True`.
Hard: Return the sum of all the numbers in a nested list such as L.

binarify(n):

Easy: Given an integer `n`, return a string of `1`s and `0`s representing `n` as a binary number.
Medium: Write a function `intify` that performs the reverse operation.
Very Hard: Make `intify` work for binary numbers in 2's complement.

Submission:

Submit your file containing all three functions to MarkUs. Remember that **Only the Easy versions will be marked**. If you wish to submit the **Hard**er versions, ensure that they have different function names (or else the auto-marker will mark them as incorrect).