Problem Set 1, Part I

Problem 3: Recursion and the runtime stack

```
3-1)
mystery(20, 6)
-------------
      a = 20
      b = 6
      myst_rest = mystery(14, 6) = 7
      return 7 + 2 = 9

      mystery(14,6)
      -----------------
      a = 14
      b = 6
      myst_rest = mystery(8, 6) = 5
      Return 5 + 2 = 7
                      mystery(8, 6)
                      -----------------
                      a = 8
                      b = 6
                      myst_rest = mystery(2, 6) = 3
                      Return 3 + 2 = 5
                            mystery(2, 6)
                            -----------------
                            a = 2
                            b = 6
                            myst_rest = mystery(-2, 6) = 1
                            Return 1 + 2 = 3



3-2) The value is 9

3-3) The stack stopped when we reached mystery(-2,6)
At that time we have :
   -  mystery(20,6)
   -  mystery(14,6)
   -  mystery(8,6)
   -  mystery(2,6)
   -  mystery(-2,6)
```

In total we have total of 5 stack frames, including main we have 6 stack frames

3-4) a value that made the recursive became infinite is when the a value never reaches the condition a < 0, which is when b is 0<= and a >= 0 because this way a will either increase or stays the same, hence it will never meet the condition a < 0
(a,b) = infinite, when :
a >= 0 (if it's < 0 it will immediately return a value)
and
b <= 0