

## HOMWORK 4

*Due Thursday, Feb 9, at 11pm*

Please enter your answers into a Jupyter notebook and submit by the deadline via canvas.

**Problem 1.** Write a function `to_binary(n)` that will return the binary representation of an integer. E.g. `to_binary(13)` should return `1101`. Write a function `from_binary(n)` that will return the base 10 representation of a binary number. e.g. `from_binary(1101)` should return `13`. These two functions should be inverses of each other.

**Problem 2.** Goldbach's conjecture.

The famous Goldbach conjecture states that every even number  $n > 2$  is a sum of two prime numbers. For example:  $4 = 2 + 2$ ,  $6 = 3 + 3$ ,  $8 = 5 + 3$ ,  $10 = 7 + 3$ ,  $12 = 7 + 5, \dots$

Write Python code that checks Goldbach's conjecture for  $n$  up to 10000. Print out how the even numbers from 9900 to 9998 can be written as sums of prime numbers. (feel free to use your `primes(n)`, which returned a list of primes up to  $n$  from the previous assignment to save time)

**Problem 4.** Recursion problems. Here is an example of a recursive function: (if you understand how this works, you understand recursion)

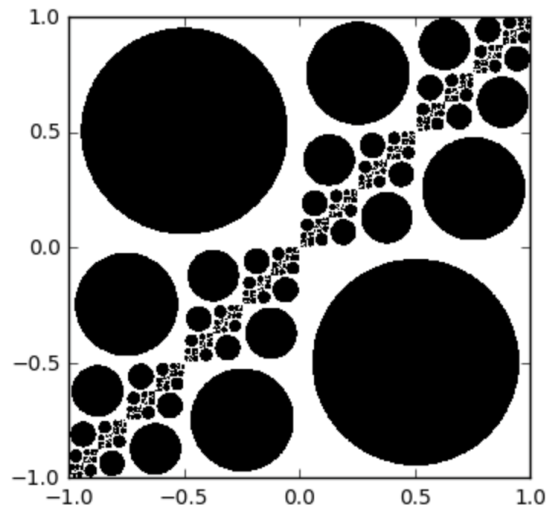
```
# returns the reverse of the list xs
def reverso(xs):
    if xs == []:
        return xs
    return xs[-1:] + reverso(xs[:-1])
```

Write the following functions only using recursion. (Please don't use loops, the idea is for us to learn to use recursion. If I gave you only recursion exercises that are easier to do with recursion and hard to do with loops, they would be very difficult and it would make it much harder for you to learn to use recursion. Also don't use the built-in function `max` for solving `max` etc.)

- (1) `maximo(xs)`, returns the maximum of a list of numbers.
- (2) `lengtho(xs)`, returns the length of a list.
- (3) `ispalin(xs)` returns True if `xs` is a palindrome (a palindrom is a sequence that is the same in reverse, example: "amanaplanacanalpanama").

- (4)  $\text{gcd}(a,b)$ , returns the greatest common divisor of two integers  $a$  and  $b$ . Use Euclid's algorithm, but implement it using recursion. Look at the notes from lecture 6 to remember (key fact: if  $a = b \cdot q + r$ , then  $\text{gcd}(a,b) = \text{gcd}(b,r)$ ).

**Problem 5.** Make the following picture by modifying the code below. (use recursion, you will need the python file `libhw01.py` that came with the first homework)



```
import libhw01 as libhw

def circ(x,y):
    return 0.0 if x*x + y*y < 0.8 else 1.0

def f(x,y,count):
    # if, when you call f recursively you use f(?,?,count+1), then the count
    # variable will keep track of how many times f called itself (how deep you
    # are in the recursion tree if you like) and will stop calling itself
    # indefinitely for bad values
    if count > 500:
        return 1.0

    if . . .

def g(x,y):
    return f(x,y,0)

libhw.drawfunction(g)
```