Problem #3 - Write a recursive function ddefintion for a function that has one parameter n and that returns the n^th Fibonacci number. Then ask the user to enter an interger n and return the n^th Fibonacci number

```
# Question #1:

def fibonacciNum(n):
  # Recursive statement in order to reach the users input of how many times they wanna run
  # through to the fibonacci sequence in order to return the desired fibonacci number

  if n <= 1:
    return 1
  else:
    return fibonacciNum(n - 1) + fibonacciNum(n - 2)


# Asking user for input
n = eval(input("Enter in a positive integer "))

# Checking User input
if n <= 0:
  n = eval(input("Please enter in a positive integer "))
else:
  print("Fibonacci Sequence: ")
  for n in range(0, n):
    print(fibonacciNum(n))
```

Problem #4 - Write a function that finds the sum of the squares of the first n numbers

```
def sumOfSquares(n):
  if n == 0:
    return n
  else:
    return n**2 + sumOfSquares(n - 1) # Recursive statement to add n squared numbers together
                                      # Ex. 1^2 + 2^2 + 3^2 = 14
# Printing Out Sum of Squares
print("Sum of Squares when n = 3: ")

# Calling sum of squares function
sumOfSquares(3)
```

Problem #5 - Write a function to find the gcd of two non-negative numbers using the Euclidean algorithm

```
def gcd(a,b):
  if b == 0:
    return a # Return the value of a since that would technically be the greatest common divi
  else:
    return gcd(b, a%b) # Recursive Function to retrieve greatest common divisor

print("The greatest common denimonater between 10 and 45 is: ", gcd(10, 45))
```