**Practice Quiz 3: Recursion**

**1.What is recursion used for?**

Recursion lets us tackle complex problems by reducing the problem to a simpler one.

**2.Which of these activities are good use cases for recursive programs? Check all that apply.**

Going through a file system collecting information related to directories and files.

Managing permissions assigned to groups inside a company, when each group can contain both subgroups and users.

**3.Fill in the blanks to make the is\_power\_of function return whether the number is a power of the given base. Note: base is assumed to be a positive number. Tip: for functions that return a boolean value, you can return the result of a comparison.**

def is\_power\_of(number, base):

if number == base:

return True

# If number is equal to 1, it's a power (base\*\*0).

if number == 1 and base == 0:

return True

# Base case: when number is smaller than base.

if number < base:

return False

# Recursive case: keep dividing number by base.

return is\_power\_of(number/base, base)

print(is\_power\_of(1,0)) # Should be True

print(is\_power\_of(8,2)) # Should be True

print(is\_power\_of(64,4)) # Should be True

print(is\_power\_of(70,10)) # Should be False

**4.The count\_users function recursively counts the amount of users that belong to a group in the company system, by going through each of the members of a group and if one of them is a group, recursively calling the function and counting the members. But it has a bug! Can you spot the problem and fix it?**

def count\_users(group):

count = 0

for member in get\_members(group):

if is\_group(member):

count += count\_users(member)

else:

count += 1

return count

print(count\_users("sales")) # Should be 3

print(count\_users("engineering")) # Should be 8

print(count\_users("everyone")) # Should be 18

**5.Implement the sum\_positive\_numbers function, as a recursive function that returns the sum of all positive numbers between the number n received and 1. For example, when n is 3 it should return 1+2+3=6, and when n is 5 it should return 1+2+3+4+5=15.**

def sum\_positive\_numbers(n):

if n == 0:

return 0

return n + sum\_positive\_numbers(n - 1)

print(sum\_positive\_numbers(3)) # Should be 6

print(sum\_positive\_numbers(5)) # Should be 15