**Python and HOFs**

**Match the Expressions**

Draw lines to connect equivalent expressions. Multiple lines allowed!

|  |  |
| --- | --- |
|  | foo += 5  foo == 5  foo = 5  foo += [5]  foo.append(5)  foo[5]  foo = foo + 5  5 in foo |

**FizzBuzz**

def fizzbuzz():

“““ Print out the numbers 1 through 100: if the number is

divisible by 3, print fizz instead. If it is divisible by 5,

print buzz instead. If it is divisible by 15, print fizzbuzz

instead. ”””

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

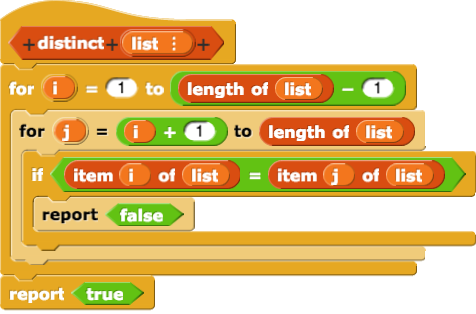
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



def distinct(lst):

for\_\_\_in\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:

for\_\_\_in\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:

if\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:

return\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Return\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Writing Python**

def iter\_even\_lst(lst):

"""Given a list, return a new list containing only the even numbers (use iteration!).

>>> even\_lst([1, 2, 3, 4, 5, 6])

[2, 4, 6]

>>> even\_lst([3, 6, 7, 7, 8, 10])

[6, 8, 10]

"""

result = \_\_\_\_\_

for item in \_\_\_\_\_\_:

if \_\_\_\_\_\_\_\_\_\_ == 0:

result.append(\_\_\_\_\_\_\_)

return \_\_\_\_\_\_\_\_\_

def double\_items(lst): #RECURSION

"""Doubles each item of the list.

>>> double\_items([1, 2, 3])

[1, 1, 2, 2, 3, 3]

"""

if lst == []:

return \_\_\_\_

else:

return \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ + double\_items(\_\_\_\_\_\_)

def factors(n): #NO RECURSION

"""Prints out all of the numbers that divide `n` evenly.

>>> factors(20)

20

10

5

4

2

1

"""

i = 1

while \_\_\_\_\_:

if \_\_\_\_\_\_\_\_:

print(\_\_\_\_\_)

i += 1

def fib(n):  
 """Returns the nth Fibonacci number.  
 >>> fib(0)  
 0  
 >>> fib(1)  
 1  
 >>> fib(2)  
 1  
 >>> fib(3)  
 2  
 >>> fib(4)  
 3  
 >>> fib(5)  
 5  
 >>> fib(6)  
 8  
 """

curr, next = 0, 1  
 while \_\_\_\_\_\_\_\_\_\_:  
 curr, next = next, \_\_\_\_\_\_\_\_\_\_  
 n -= 1  
 return \_\_\_\_\_\_\_\_\_\_\_

**Coding Interview Practice with Will & Lara**

1) Given a list of numbers, find the sum of all the numbers in the list.

def sum(lst):

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2) Find the sum of all numbers from 1 to n:

def sum\_of\_ordered(n):

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3) You’re given a list of numbers, lst, that has all numbers from 1 to n but is missing one number. Write a function that takes in “lst” and “n” and returns the missing number. (Hint: Use the “sum\_of\_ordered” and “sum” functions you defined earlier.)

def missing\_num(lst, n):

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ANSWERS:

def iter\_even\_lst(lst):

result = []

for item in lst:

if item % 2 == 0:

result.append(item)

return result

def double\_items(lst):

if lst == []:

return []

else:

return [lst[0], lst[0]] + double\_items(lst[1:])

def factors(n):

i = 1

while i <= n:

if n%i == 0:

print(n//i)

i += 1

def fib(n):

curr, next = 0, 1  
 while n > 0:  
 curr, next = next, curr + next  
 n -= 1  
 return curr

Coding Interview Practice w/William and Lara:

1) Given a list of numbers, find the sum of all the numbers in the list.

def sum(lst):

total = 0

for num in lst:

total += num

return sum

2) Find the sum of all numbers from 1 to n:

def sum\_of\_ordered(n):

return n \* (n + 1) / 2

3) You’re given a list of numbers, lst, that has all numbers from 1 to n but is missing one number. Write a function that takes in “lst” and “n” and returns the missing number. (Hint: Use the “sum\_of\_ordered” and “sum” functions you defined earlier.)

def missing\_num(lst, n):

return sum\_of\_ordered(n) - sum(lst)