With pointers and reference

C- Language

Python is reference but can’t be dereference

Find the K in

An element to array if heap is not full

Stored the biggest K in the heap

Drain the entire heap? Or pop the biggest K?

Modification of quick sort?

Pivot? 2 partitions?

Galvanize recommended a guy for Starbucks dev job

Python machine-learning

**Codewars**

**Sessions**

Function that is going to take 2 objects and going to take 2 attributes in the objects

2 values are a range

Write a code if the is an overlap in a range.

A {‘a’:2, ‘b’:5}

B {‘c’:4, ‘d’:6}

‘b’ overlaps with ‘c’

Write a function true or false

2 values

2 attributes

Solve it static then move into dynamic

range\_overlap def (a,b):

a[‘a’]2 a[‘b’]5

b[‘c’]4 b[‘d’]6

If object 1 attribute 2 is higher than object 2 attribute 1, then.

A {‘a’:2, ‘b’:5}

B {‘c’:4, ‘d’:6}

# static

If b1 < a2

return true #overlaps

Elif a1 < b2

Return true

Else

False

# dynamic

Sort B and A first, who gets A or B. evaluate.

If b2 > a1

S = ‘b’ # string

S = [b] # list

S = {whaterver: ‘b’}

**Session**

If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23. Finish the multiples so that it returns the sum of all the multiples of 3 or 5 **below the number passed in**. note if the number is a multiple of both 3 and 5, only count it once.

Pseudo code

Input variable

Pass variable to array

Mod 3 OR mod 5

Store in array

Results sum

def threeorfive(a)

result = []

for i in range(1, a): # excludes 15

if i % 3 is 0 \

OR i % 5 is 0;

results.append(i)

return sum(result)

print(threeorfive(15))

**repl**

**session**

a rectangle can be split up into a grid of 1x1 squares, the amount of which being equal to the product of the two dimensions of the rectangle. Depending on the size of the rectangle, that grid of 1x1 squares can also be split up into larger squares, for example a 3x2 rectangle has a total of 8 squares, and two possible 2x2 square. A 4x3 rectangle contains 20 squares

your task is to write a function findSquares that returns the total number of squares for any given rectangle, the dimensions of which being given as two integers with the first always being equal to or greater than the second.