# ICS0015 Fundamentals of Python | Exam

## Question 1

The lines will not print the same thing. The reason for this is that the print function takes in an argument, which it prints to the console. In the first print call, a variable is supplied and the interpreter will evaluate it (grab its current value) and pass that into the functon to be printed so the user will see '7' on the screen. In the second call, a string literal is passed directly into the function and it's evaluated to be literally what it's value is (hence why these are called literals), so the user will see 'a' on the screen.

## Question 2

### a)

No it will not.

### b)

Two changes needed: 1. on the line where eval() is called, there is a missing brace at the end and 2. C = 2πr needs to be moved after the eval() line so that it's value is calculated AFTER we get the radius

## Question 3

### a) 3

Output: 'wow' '4'

### b) 21

Output: 'whoa' '21'

### c) 17

Output: '27'

### d) -4

Output: '-3'

### e) 0

Output: 'whoa' '0'

## Question 4

An empty line. (crlf or cr or something else depending on the platform)

## Question 5

As per documentation, random.randrange will return a number satisfying

0 <= N <= 100

so the numbers that might be outputted are: 34, 100, 0, 99.

## Question 6

### a)

Valid Output: 26

### b)

Invalid, proc expects 2 arguments but only the first one is provided.

### c)

Invalid, proc expects one argument but two were supplied.

## Question 7

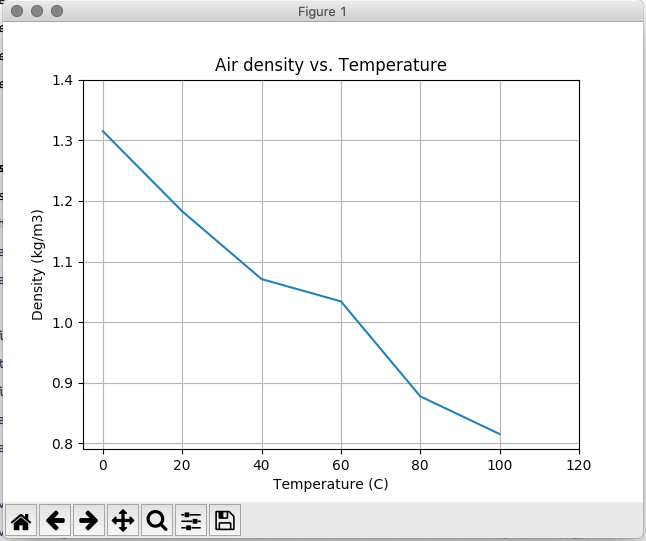
The function is a recursive function since a call to the function itself is found inside the body of the function (as a matter of fact, two calls).

If the number of disks is 3, the first pole A will have 3 disks. Pole B and C will be empty. This algorithm will: 1. Move disk 1 from rod A to rod C 2. Move disk 2 from rod A to rod B 3. Move disk 1 from rod C to rod B 4. Move disk 3 from rod A to rod C 5. Move disk 1 from rod B to rod A 6. Move disk 2 from rod B to rod C 7. Move disk 1 from rod A to rod C

## Question 8

I'll use the excellent matplotlib library to draw the chart.

from matplotlib.pyplot import plot, xlabel, ylabel, show, title, grid, xticks, yticks  
from random import uniform  
  
def main():  
 title('Air density vs. Temperature')  
 grid(True)  
 xlabel('Temperature (C)')  
 ylabel('Density (kg/m3)')  
  
 x\_values = [x \* 20 for x in range(0, 6)]  
 y\_values = [round(x \* 0.1, 1) + uniform(-0.05, 0.05) for x in reversed(range(8, 14))]  
  
 plot(x\_values, y\_values)  
 xticks(x\_values + [120])  
 yticks([round(x \* 0.1, 1) for x in reversed(range(8, 15))])  
 show()  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

This will result in something like this: 

Since the reference image does not state how the values on the y-axis are chosen, I added a bit of random to make it more similar to the provided picture.

## Question 9

from pandas import read\_excel, concat  
  
def main():  
 data1 = read\_excel('filenameA.xlsx')  
 data2 = read\_excel('filenameB.xlsx')  
 data3 = read\_excel('filenameC.xlsx')  
  
 out\_data = concat([data1, data2, data3])  
 out\_data.to\_excel('dataOut.xlsx')  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

## Question 10

I'll use the cv2 library we used in the course for this question.

from cv2 import imread, imshow, waitkey, Canny, getRotationMatrix2D, warpAffine, destroyAllWindows, fastNlMeansDenoisingColored  
  
def main():  
 img = imread('fight.jpg')  
  
 # Edge detection  
 img = Canny(img, 100, 200)  
  
 # Rotation  
 (h, w) = img.shape[:2]  
 rotationMatrix = getRotationMatrix2D((w / 2, h / 2), 60, 1.0)  
 img = warpAffine(img, rotationMatrix, (h, w))  
  
 # Removing noise  
 img = fastNlMeansDenoisingColored(img, None, 20, 10, 7, 21)  
  
 # Display the final product  
 imshow('Processed image', img)  
 waitkey(0)  
 destroyAllWindows()  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 print('Thank You for the course, best regards - Vootele Verrev.')  
 main()