

Question 1. [5 MARKS]

Consider the following (incomplete) function:

```
def in_range(x, y, p):  
    '''Return True iff x is less than y, and p is in the range x to y  
    inclusive. x, y and p are all floats'''
```

Part (a) [3 MARKS]

Write the body of this function using an if statement.

Solution:

```
    if x < y and x <= p and p <= y:  
        return True  
    else:  
        return False
```

Part (b) [2 MARKS]

Write the body of this function as a single line of code, without using an if statement.

Solution:

```
    return x < y and x <= p and p <= y
```

Question 2. [8 MARKS]

The following program runs without errors:

```
1  def one(a):
2      a = a / 2
3      # What does memory look like now?
4
5  def two(b):
6      x = 10 + b / 2
7      return x
8
9  if __name__ == "__main__":
10     having = 15
11     print one(having)
12     print having
13     fun = 16
14     print two(fun)
15     print fun
```

Part (a) [4 MARKS]

Draw the state of memory at the moment when the program reaches line 3, as part of the call to function **one**. Use the notation we have used in class. Include any namespaces and the names that have been defined within them.

Solution: See separate file with the drawing.

Part (b) [4 MARKS]

This program produces exactly 4 lines of output. Show them below:

Solution:

None
15
18
16

Question 3. [9 MARKS]**Part (a)** [5 MARKS]

For Assignment 1, you wrote function `amount_color`. Suppose it has been written correctly, with the following function def.

```
def amount_color(p):  
    '''Return the total (int) amount of colour in Pixel p.'''
```

Complete the following function according to its docstring description. You **must** call `amount_color` in your function. You do not have to import it. Assume that `media` has been imported.

```
def funky_effect(source, t):  
    '''source is a Picture and t is an int. Set each pixel in source whose amount of  
    color is below the threshold t to the color media.olive, and set each pixel whose  
    amount of color is at least t to the color media.orange.'''
```

Solution:

```
for pixel in source:  
    if amount_color(pixel) < t:  
        media.set_color(pixel, media.olive)  
    else:  
        media.set_color(pixel, media.orange)
```

Part (b) [4 MARKS]

Write a main block that allows the user to choose a file, applies the “funky effect” from part (a) to the picture in that file (using a threshold value of 150), and displays the resulting picture. Assume that the `media` module has been imported and that the user chooses a file that does indeed contain a picture.

```
if __name__ == "__main__":
```

Solution:

```
p = media.load_picture(media.choose_file())  
funky_effect(p, 150)  
media.show(p)
```