

Curso > Week 4... > 7. Testi... > Exercis...

Audit Access Expires 5 de ago de 2020

You lose all access to this course, including your progress, on 5 de ago de 2020.

Exercise 7

Finger Exercises due Aug 5, 2020 20:30 -03 Completo

Exercise 7

2/2 points (graded)

ESTIMATED TIME TO COMPLETE: 3 minutes

Consider the following function definition:

```
def f(n):
    """
    n: integer, n >= 0.
    """
    if n == 0:
        return n
    else:
        return n * f(n-1)
```

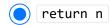
When we call f(3) we expect the result 6, but we get 0.

When we call f(1) we expect the result 1, but we get 0.

When we call f(0) we expect the result 1, but we get 0.

Using this information, choose what line of code should be changed from the following choices:





O else:	
return n * f(n-1)	
✓	

How should this line be rewritten?

Explanation:

This is a function known as <u>factorial</u> - the product of all the numbers from 1 through n. The base case of factorial is 0! = 1, but the original code was written with the base case 0! = 0. You can see why the original code was broken by writing out the recursive expansion of f(3):

$$f(3) = 3 * f(2)$$

$$= 3 * (2 * f(1))$$

$$= 3 * (2 * (1 * f(0)))$$

$$= 3 * (2 * (1 * (0)))$$

$$= 3 * (2 * (0))$$

$$= 3 * (0)$$

$$= 0$$

The fixed version of the code puts the line [return 1], instead of [return n], when [n == 0]. We can see that this modified version of the code fixes the factorial function by again writing out the recursive expansion of [f(3)]:

$$f(3) = 3 * f(2)$$

$$= 3 * (2 * f(1))$$

$$= 3 * (2 * (1 * f(0)))$$

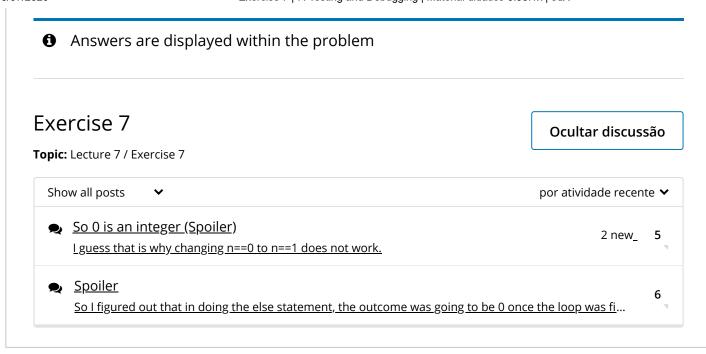
$$= 3 * (2 * (1 * (1)))$$

$$= 3 * (2 * (1))$$

$$= 3 * (2)$$

$$= 6$$

Enviar



© All Rights Reserved