



[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:

☐ `if n == 0:`

☒ `return n`





else:



return n \* f(n-1)



How should this line be rewritten?

 Answer: return 1

### Explanation:

This is a function known as factorial - 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)`:

$$\begin{aligned}
 f(3) &= 3 * f(2) \\
 &= 3 * (2 * f(1)) \\
 &= 3 * (2 * (1 * f(0))) \\
 &= 3 * (2 * (1 * (0))) \\
 &= 3 * (2 * (0)) \\
 &= 3 * (0) \\
 &= 0
 \end{aligned}$$

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)`:

$$\begin{aligned}
 f(3) &= 3 * f(2) \\
 &= 3 * (2 * f(1)) \\
 &= 3 * (2 * (1 * f(0))) \\
 &= 3 * (2 * (1 * (1))) \\
 &= 3 * (2 * (1)) \\
 &= 3 * (2) \\
 &= 6
 \end{aligned}$$



05/07/2020Exercise 7 | 7. Testing and Debugging | Material didático 6.00.1x | edX

Answers are displayed within the problem

Exercise 7

Ocultar discussão

Topic: Lecture 7 / Exercise 7

Show all posts

▼

por atividade recente

▼

So 0 is an integer (Spoiler)

2 new\_ 5

I guess that is why changing `n==0` to `n==1` does not work.

Spoiler

6

So I figured out that in doing the else statement, the outcome was going to be 0 once the loop was fi...

© All Rights Reserved

https://courses.edx.org/courses/course-v1:MITx+6.00.1x+2T2020/courseware/f0a19f0a8c2d49f3aa78ef3823845271/78648402e8bf4738ade9710... 3/3