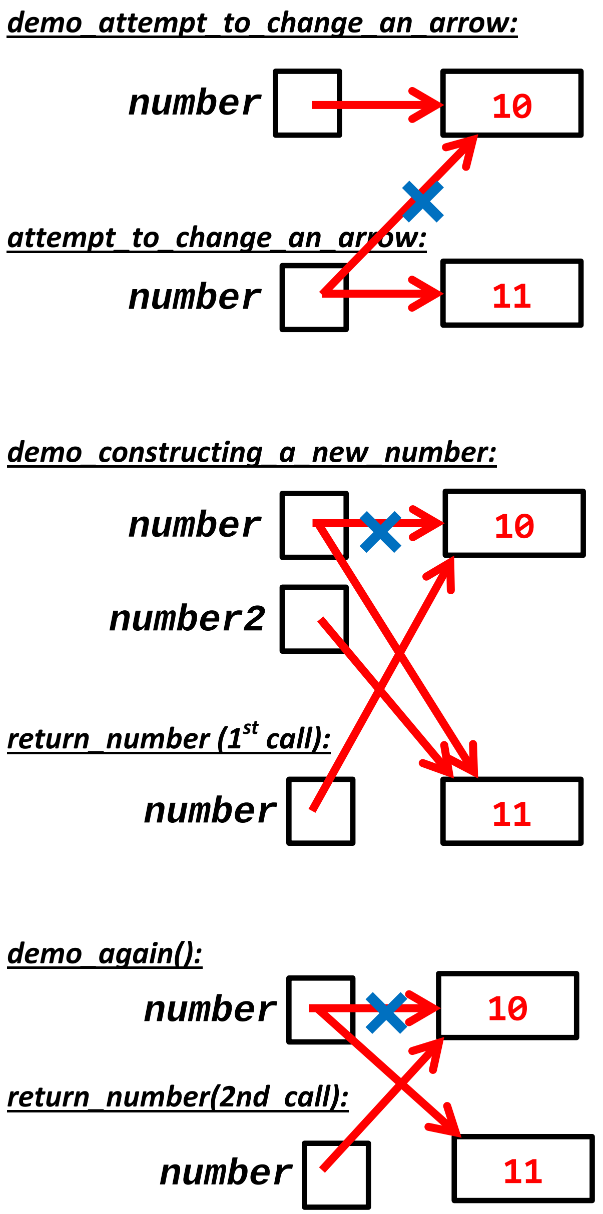
6. It is simply not possible for a function to change the arrow in the ***caller*** that corresponds to one of the function's arguments. If you really want to accomplish something like that, you have to return a value and re-assign the variable that points to the argument to that returned value. Try this one to see those ideas in action:

   Complete the Box-and-Pointer diagram to the right to show what happens when *main* (below) executes. Also show the output that is printed.

```
def main():
    demo_attempt_to_change_an_arrow()
    demo_constructing_a_new_number()
    demo_again()

def demo_attempt_to_change_an_arrow():
    number = 10
    attempt_to_change_an_arrow(number)
    print('B.', number)

def attempt_to_change_an_arrow(number):
    number = number + 1
    print('A.', number)

def demo_constructing_a_new_number():
    number = 10
    number2 = return_number(number)
    print('C.', number, number2)
    number = number2
    print('D.', number, number2)

def demo_again():
    number = 10
    number = return_number(number)
    print('E.', number)

def return_number(number):
    return (number + 1)
```

Box and Pointer diagram:


demo_attempt_to_change_an_arrow:
number
10
attempt_to_change_an_arrow:
number
11
demo_constructing_a_new_number:
number
10
number2
return_number (1st call):
number
11
demo_again():
number
10
return_number(2nd call):
number
11

Output: A. _____11______

B._________10___________

C.______10____11______

D.______11____11_________

E._______11________________