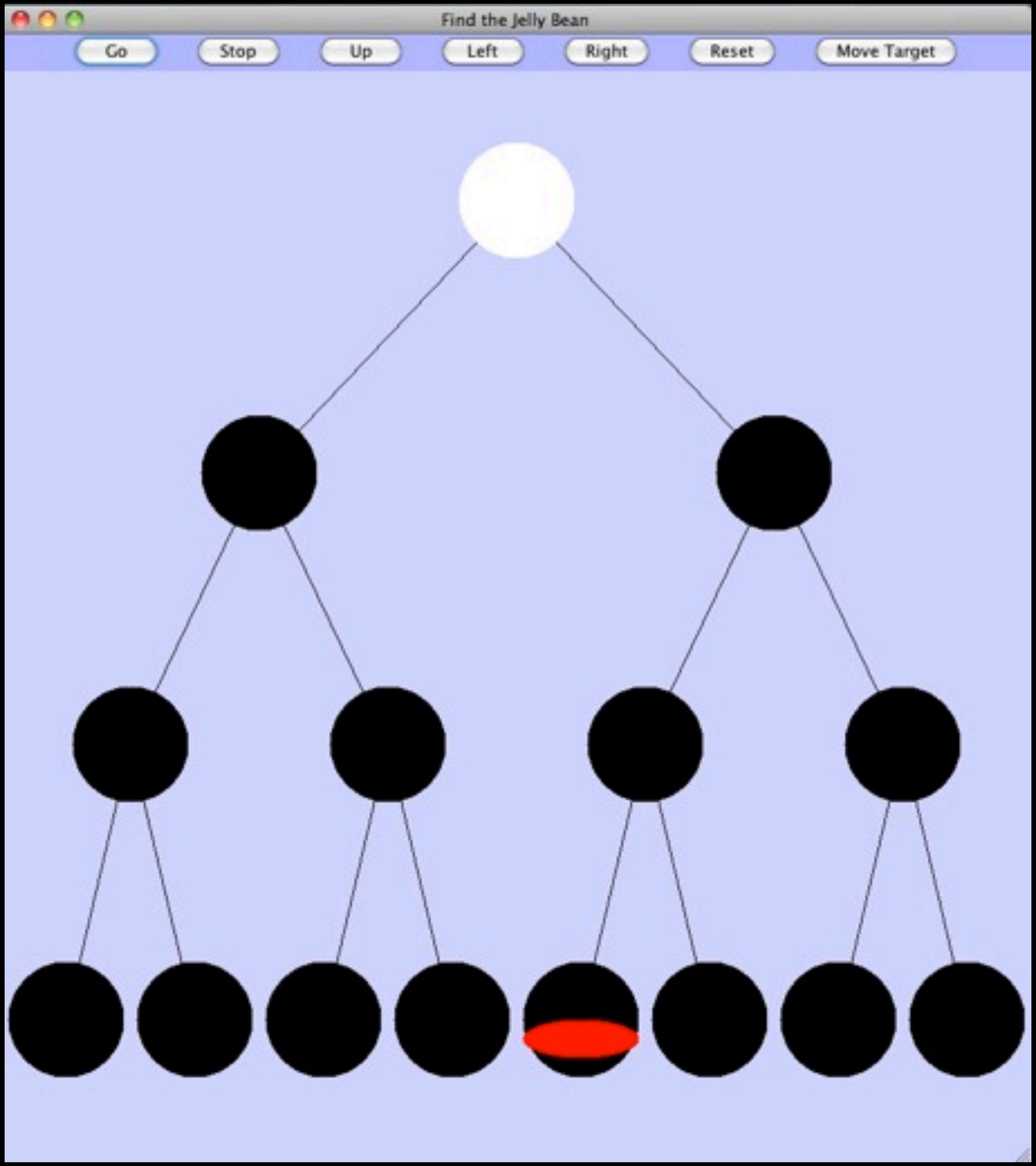
Slides for the Week

CS273 Laboratory 12

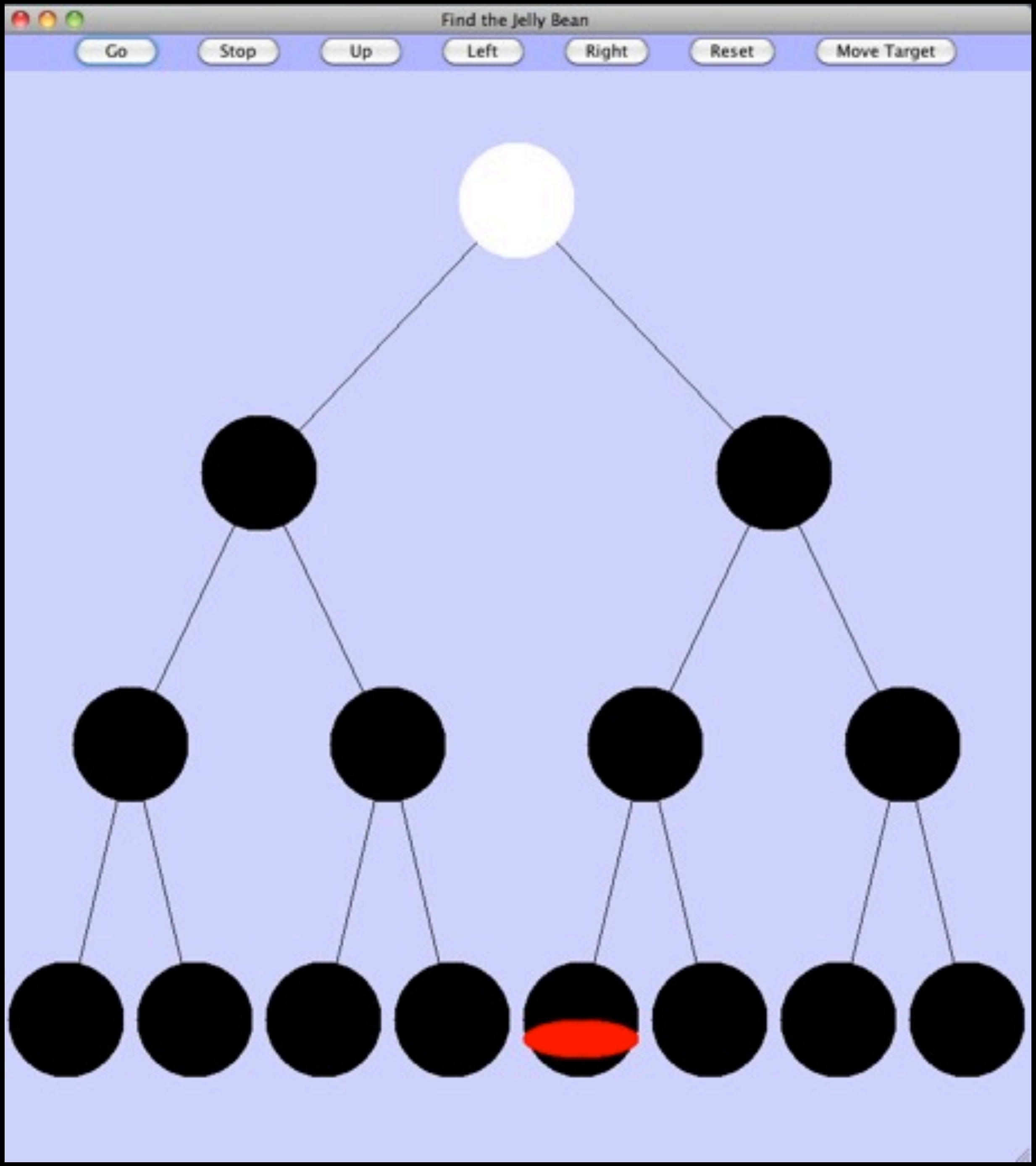
This week’s lab focuses on recursion and file I/O.

Recursion:

You must implement the traverse()method so that the white finder automatically locates the red jellybean in this tree:



You must do so using recursion.



A recursive method is a method that makes a call to itself.

Suppose you wanted to implement a factorial() method.

You could do it iteratively.

That is, use a loop:

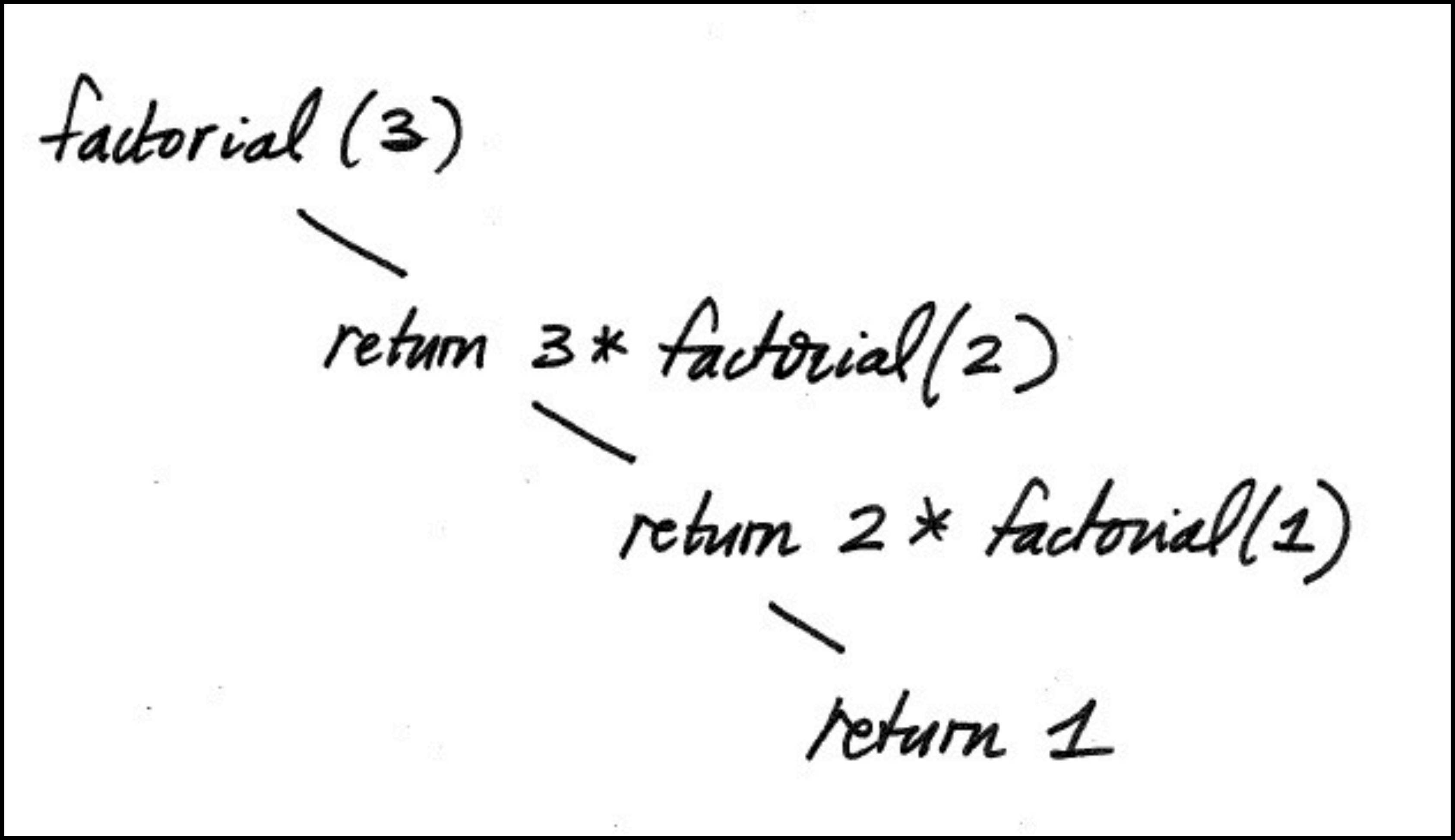
```
// Iterative implementation of factorial.
public int factorial(int n)
{

    int result = 1;

    while (n > 0)
    {
        result = result * n;
        n = n - 1;
    }

    return result;
}
```

You could also do it recursively.

That is, allow factorial() to call itself:

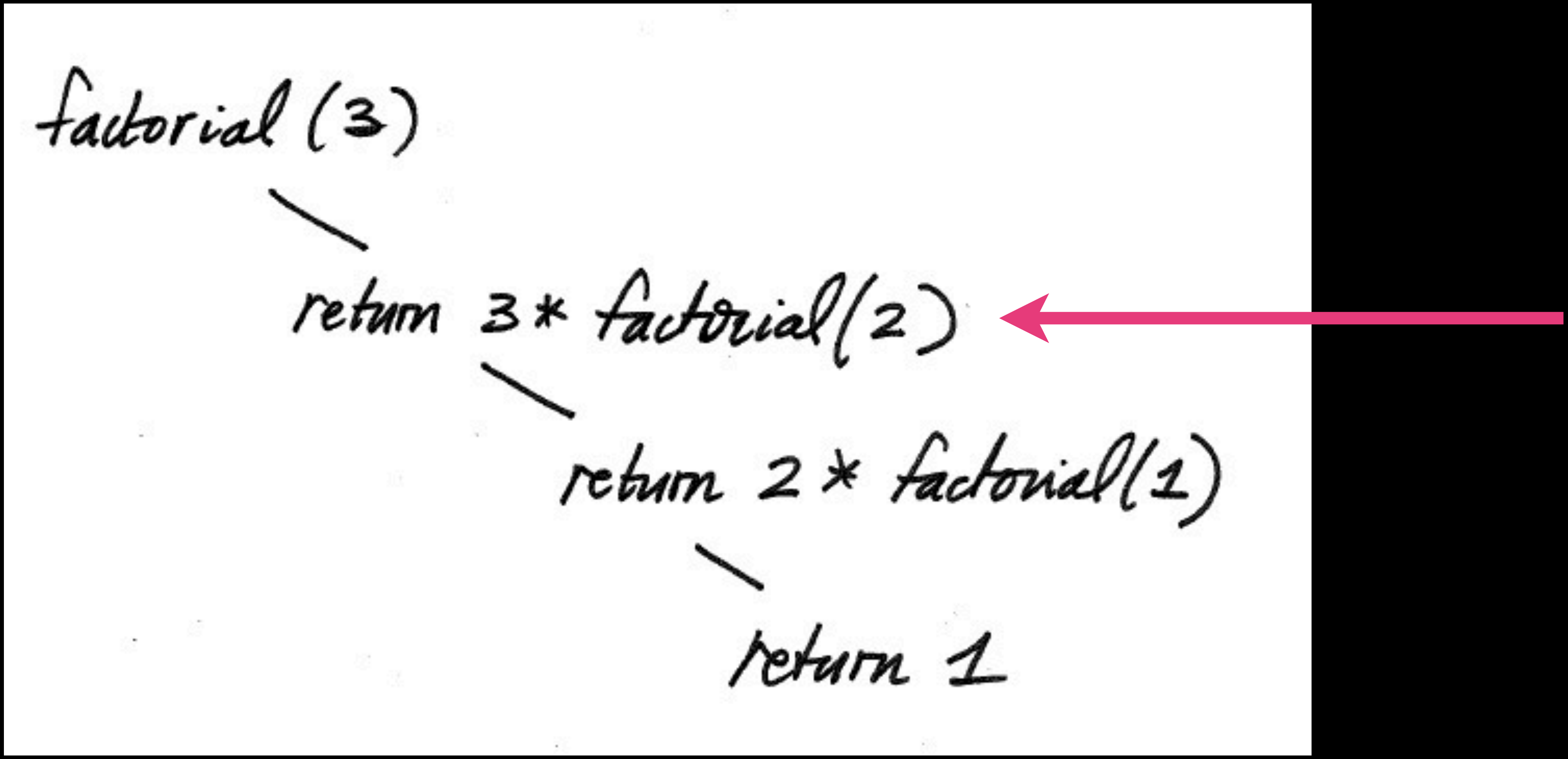
```
// Recursive implementation of factorial.
public int factorial(int n)
{
    // Base case.  Test whether to stop the recursion
    // or continue.  factorial(0) and factorial(1)
    // are both 1.
    if (n <= 1)
    {
        return 1;
    }

    // Recursive case.  Recursion continues with a
    // recursive call.
    else
    {
        // Multiply this value of n with the result
        // of calling factorial(n-1).
        return n * factorial(n-1);
    }
}
```

Suppose you want to call the recursive implementation of factorial(3).



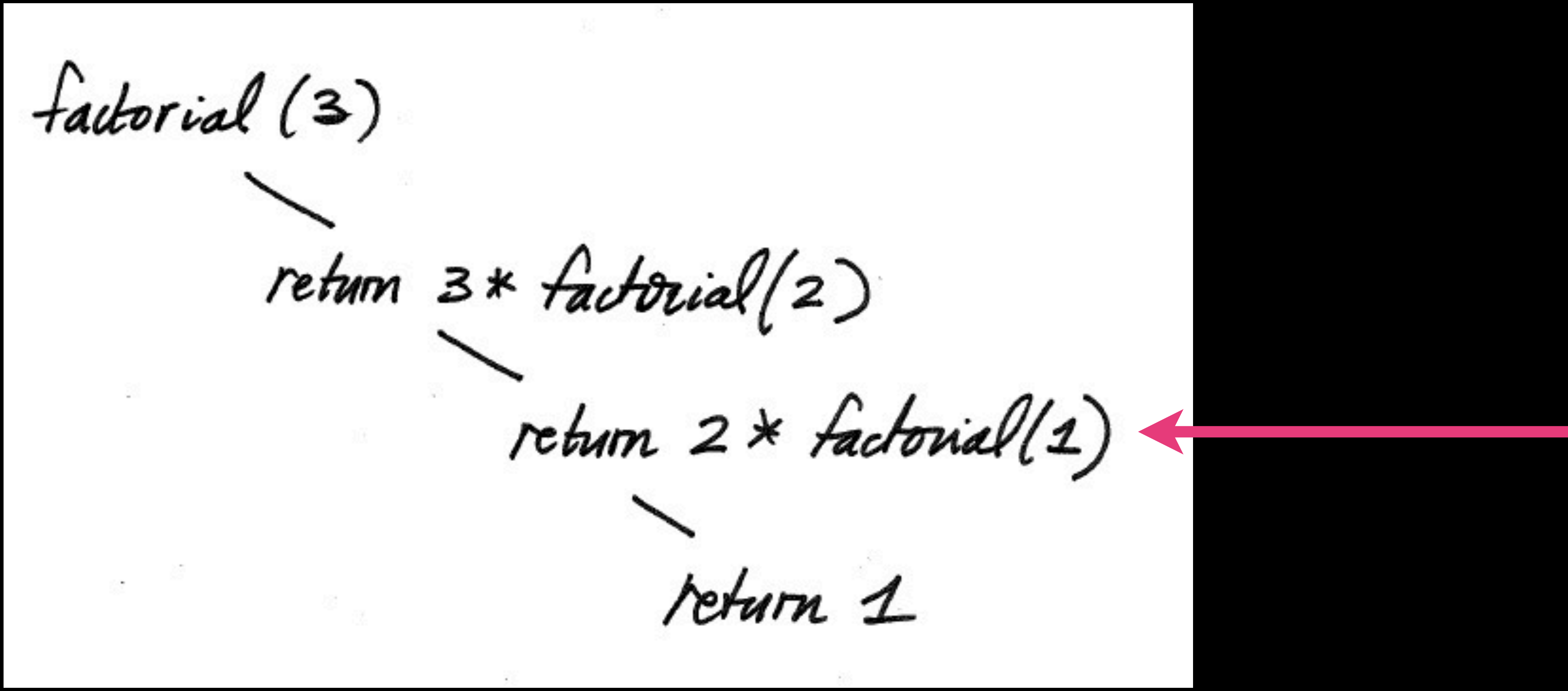
call tree for factorial(3)

factorial(3) will call factorial(2) in its recursive case:



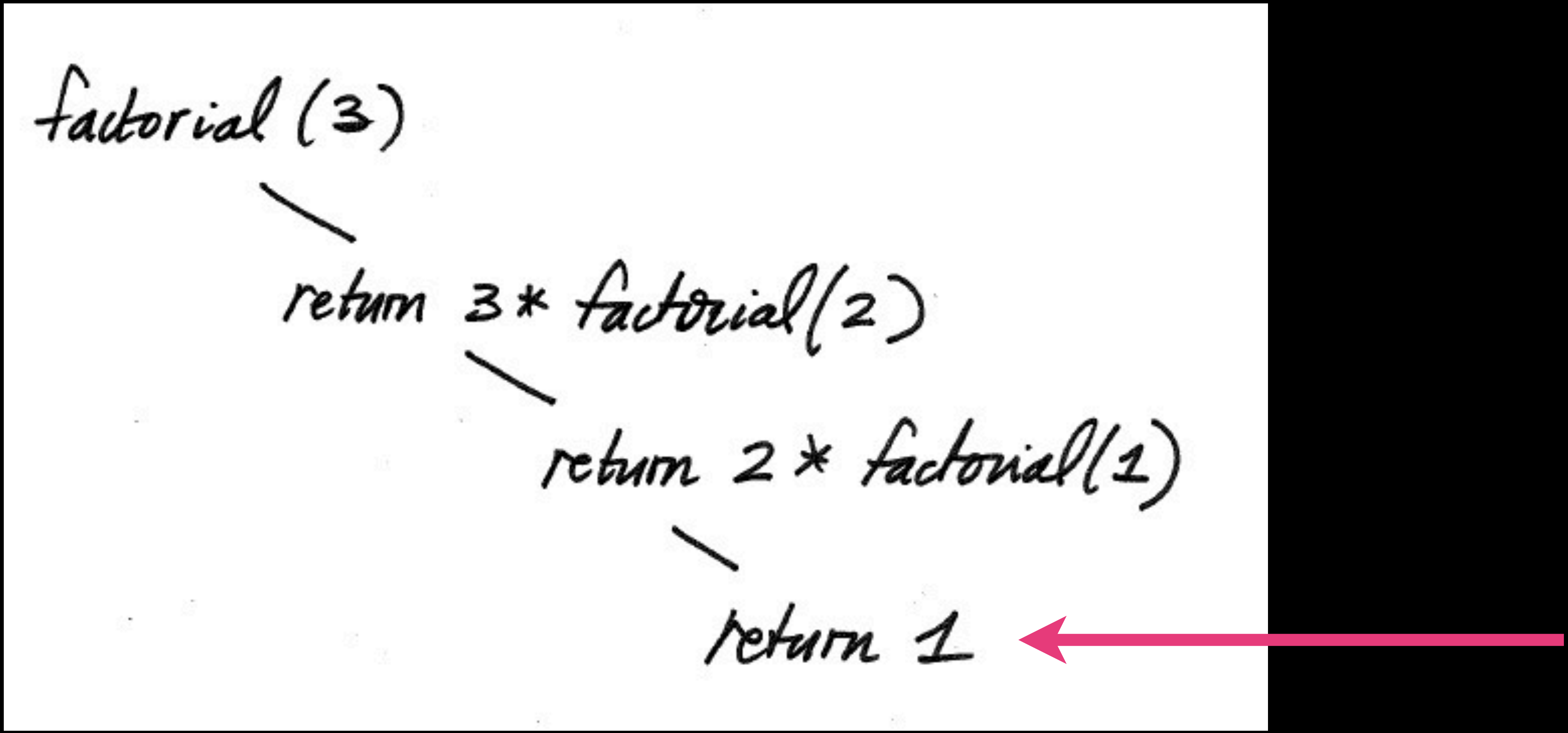
call tree for factorial(3)

factorial(2) will call factorial(1) in its recursive case:



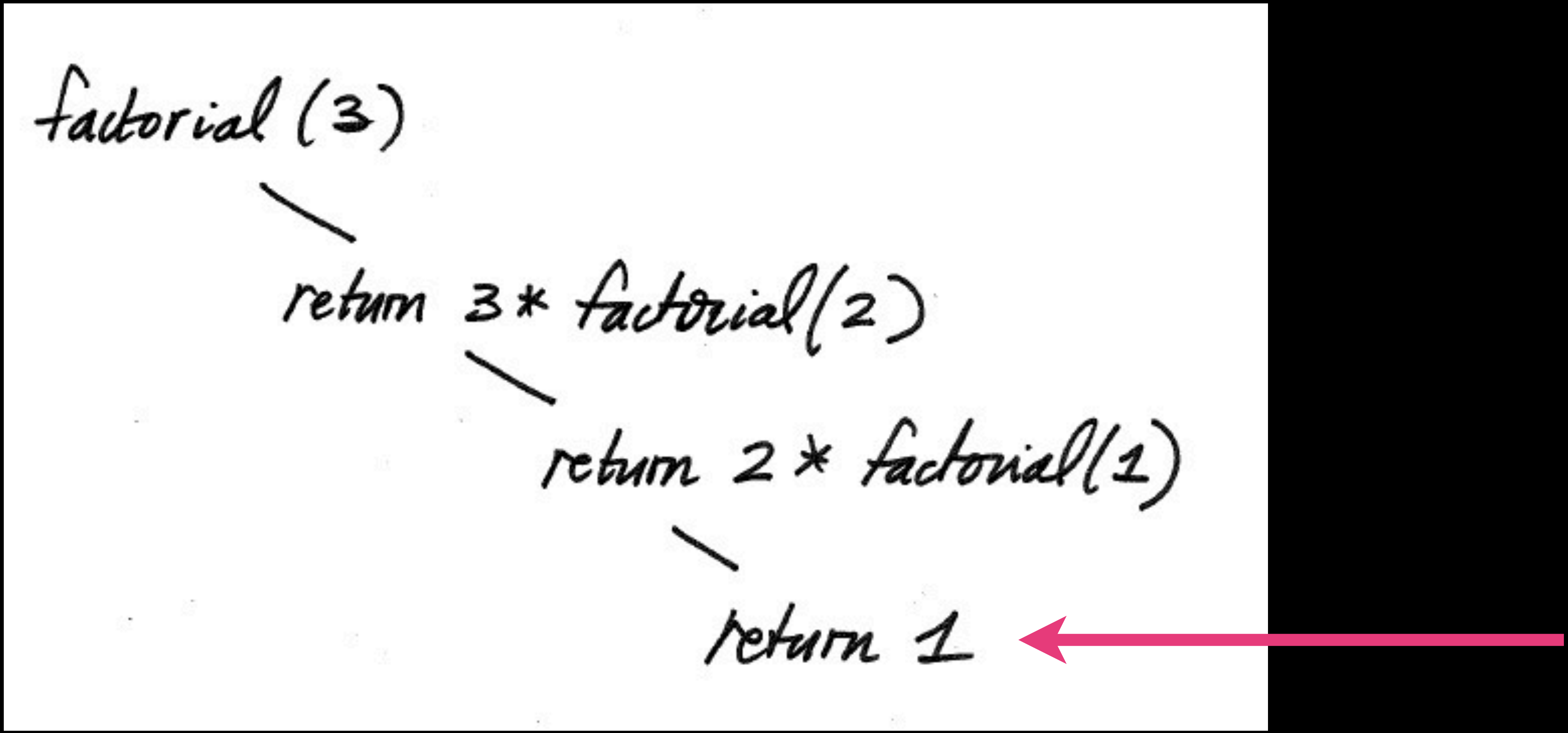
call tree for factorial(3)

And when factorial(1) is called, the base case is hit, and factorial(1) returns 1.



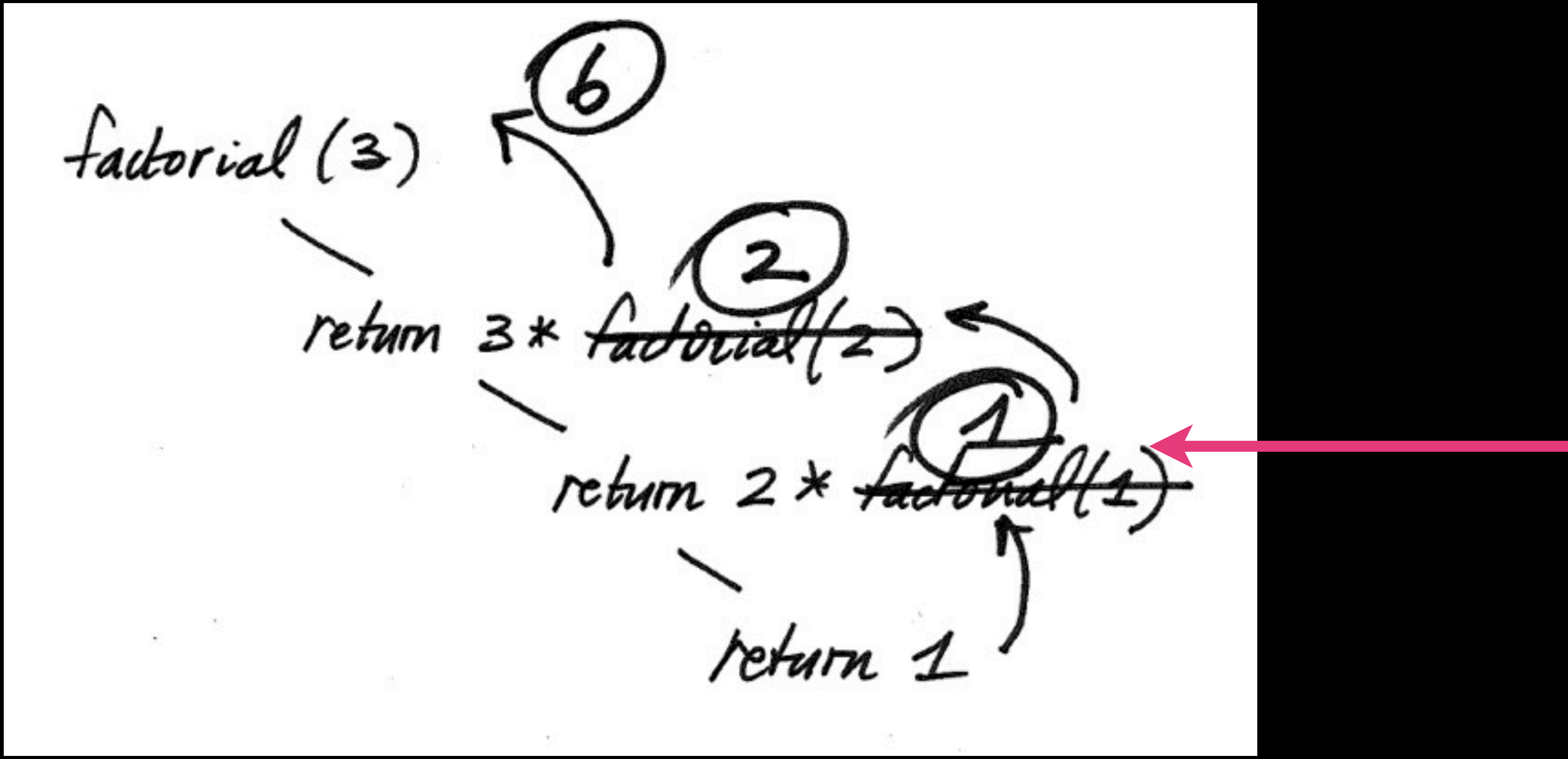
call tree for factorial(3)

As all of the recursive calls return, the multiplication is performed...



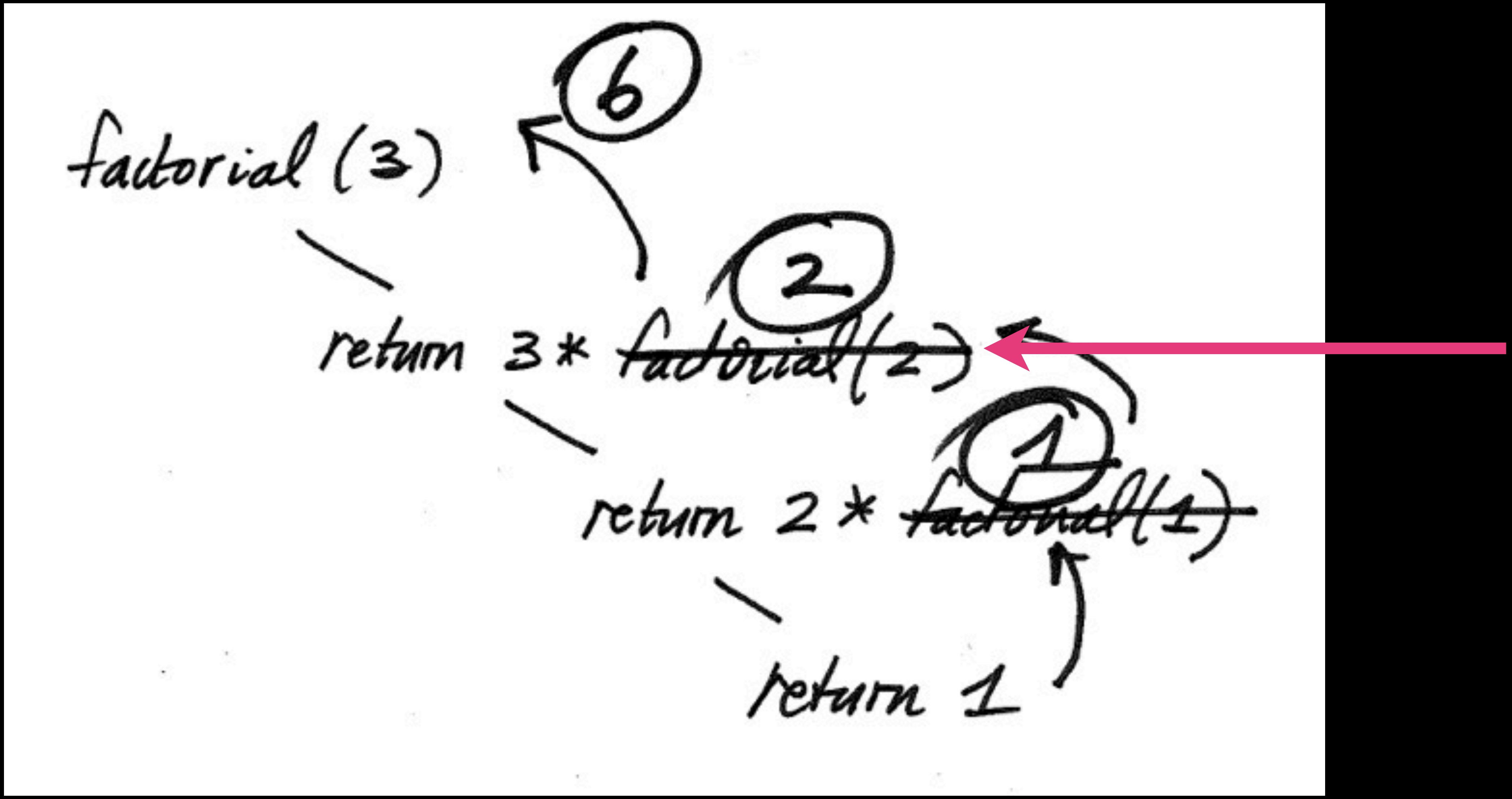
call tree for factorial(3)

factorial(1) returns 1. The factorial(2) call computes 2 * 1 and returns the value.



call tree for factorial(3)

factorial(2) returns 2. The factorial(3) call computes 3 * 2 and returns the value of 6.



call tree for factorial(3)

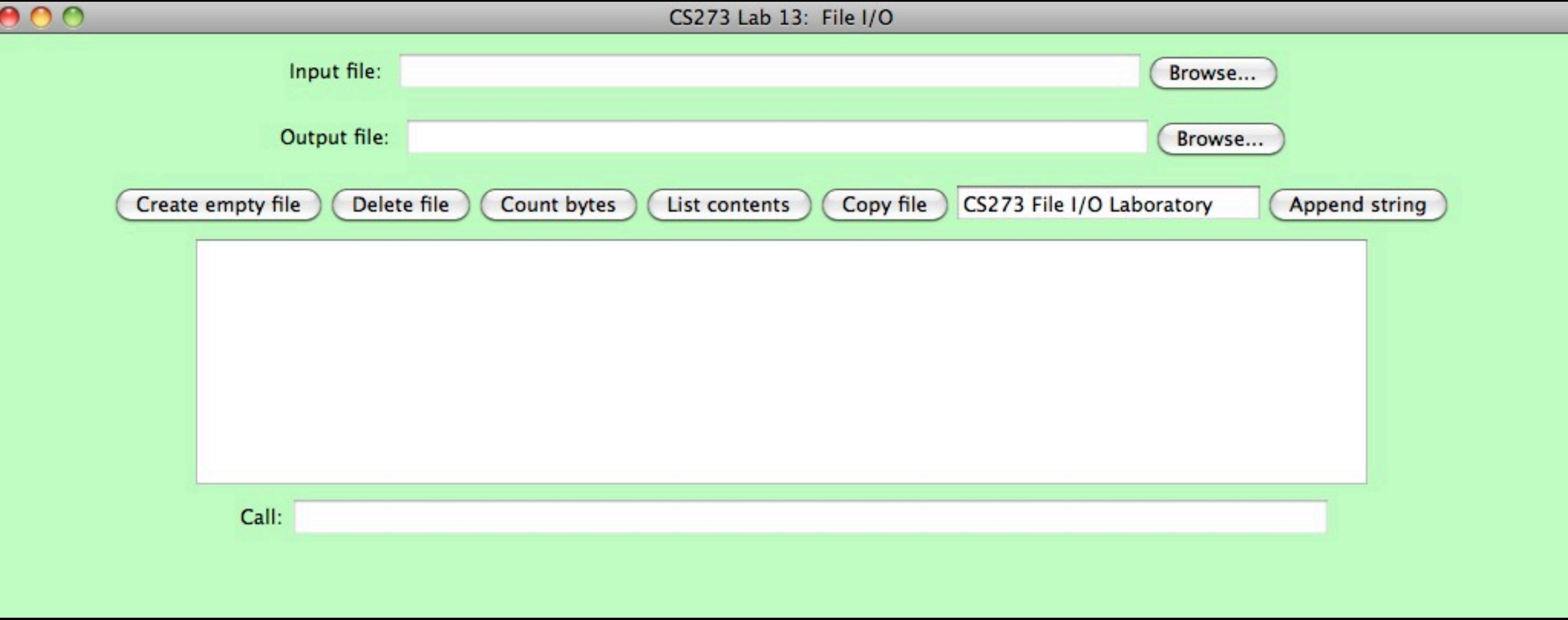
recursion is **powerful** because it allows one to write very small code that can be easy to reason about.

but students who first learn about recursion report that it feels a little bit like black magic.

File I/O:

The starter code provides a small JFrame-based GUI that is built to offer File I/O functionality.

You must implement all of the functions stubbed out in the FileHandler class so that these buttons work.



CS273 File I/O Laboratory

As you implement the File I/O checkpoints, be sure to use the classes recommended by each checkpoint.

Good Luck!

If you have any questions the TAs and I are happy to help.