

### 1. Proceed with call-tion

For each of the expressions in the table below, write the output displayed by the interactive Python interpreter when the expression is evaluated. The output may have multiple lines. If an error occurs, write “Error”, but include all output displayed before the error. If a function value is displayed, write “Function”. The first row has been provided as an example. Recall: The interactive interpreter displays the value of a successfully evaluated expression, unless it is None. Assume that you have started `python3` and executed the following statements. Changes to values persist across subproblems.

```
x = 3
```

```
def p(rint):
    print(rint)
```

```
def g(x, y):
    if x:
        print("one")
    elif x:
        print(True, x) # Does x being truth-y affect the printed value?
    if y:
        print(True, y) # Does y being truth-y affect the printed value?
    else:
        print(False, y) # Does y being false-y affect the printed value?
    return print(p(y)) + x
```

Expression	Interactive Output
<code>print(4, 5) + 1</code>	4 5 Error
<code>2 * 2 * 1 + x * x</code>	
<code>print(3 * 3 * 1)</code>	
<code>print(x + 1 * x + 1)</code>	
<code>print(print(x + 1 * x + 1))</code>	
<code>print(print(x + 1 * x + 1) + 1)</code>	
<code>print(p("rint"))</code>	
<code>x, y = 2, x</code> <code>g(y, x)</code>	
<code>g(y, p("rint"))</code>	