

CSCA08H Winter 2022 Worksheet: Nested Function Calls

Recall that for a function call the arguments are evaluated left to right, and only then is the function call executed.

As an example, we have underlined parts of the expression below and numbered them to indicate the order in which the subexpressions are evaluated.

```
f(g() + 3, h())
--- -
1   2
----- ---
3   4
-----
5
```

That says that function call `g()` is evaluated first, then the `3`, then the result of the call on `g` is added to `3`, then function call `h()` happens, and finally (now that the arguments to `f` have been evaluated) the call on `f` happens.

1. In function `max_of_min` below, underline and number the subexpressions in the `return` expression (the one starting with `max(min(...))`) to indicate the order in which the subexpressions are evaluated. We've done the first two for you. (The function's type contract has been omitted to save space.)

```
def max_of_min(num1, num2, value1, value2):
    return max(min(num1, num2), min(value1, value2))
----- -
1   2
```

2. In the following expression, underline and number the subexpressions to indicate the order in which they are evaluated. We've done the first one for you.

```
pow(2, pow(pow(2, 1), 2))
-
1
```