DATA C88C
Summer 2025

Functions
Discussion 1: June 24, 2025

Functions

Functions (such as pow) are called using *call expressions* (such as pow(10, 3) which returns 1000; 10 to the third power). The body of a function can print or return values (or both).

- When print is called anywhere, its arguments are displayed right away.
- Executing a return statement stops the function call and provides the value for its call expression.
- When the end of a function body is reached without a return, the function returns None.

Tip: If a value needs to be used by code outside of a function, then return the value (instead of printing it).

Q1: Multiply

Implement multiply, which takes two numbers x and y. It prints out an equation showing the result of multiplying x and y, then returns the result.

Try to multiply numbers together just once in the body of multiply so that the Python interpreter doesn't have to perform multiplication more *times* than necessary (pun intended).

```
def multiply(x, y):
    """Multiply x by y and print what happened.

    >>> multiply(3, multiply(multiply(4, 5), 6))
    4 * 5 = 20
    20 * 6 = 120
    3 * 120 = 360
    360
    """
    z = x * y
    print(x, '*', y, '=', z)
    return z
```

Calling print on multiple arguments displays those values separated by spaces. For example, the call expression print(1, '+', 2, '=', 3) displays 1 + 2 = 3.

Digits

Each digit of a positive integer corresponds to a power of 10. Here are some examples of manipulating digits, illustrated with n = 357 and d = 9. - Remove the last digit of n: n // 10 is 35. - Remove the last two digits of n: n // pow(10, 2) is 3. - Add together the last two digits of n: n % 10 + n // 10 % 10 is 12. - Put digit d at the end of integer n: n * 10 + d is 3579.

Q2: Cut One Out

Implement `cut`, which takes non-negative integers `n` and `k` and has only a return statement in its body. It returns a positive integer with all of the digits of `n` except the digit that is `k` to the left of the rightmost digit (the one's digit). If `k` is 0, then it returns `n` without its one's digit. If there is no digit `k` to the right of the one's digit, then it returns `n`.

```
def cut(n, k):
    """Return n with the kth digit from the right removed.

    >>> cut(3579, 2)
    379
    >>> cut(3579, 0)
    357
    >>> cut(3579, 1)
    359
    >>> cut(3579, 5)
    3579
    """
    return n // pow(10, k + 1) * pow(10, k) + n % pow(10, k)
```

Please don't look at the hint until everyone in your group agrees that you're stuck and need some extra help.

To implement a function, it can be helpful to open a Python interpreter and focus on an example. Here are all the pieces you need to put together to solve this problem for the example `n=3579` and `k=2`.

```
>>> n = 3579
>>> k = 2
>>> pow(10, k)
100
>>> pow(10, k + 1)
1000
>>> n // 1000
3
>>> 3 * 100
300
>>> 3579 % 100
79
>>> 300 + 79
379
```

Note: This worksheet is a problem bank—most TAs will not cover all the problems in discussion section.