Lab 1 Solutions lab01.zip (lab01.zip)

Solution Files

Additionally, please fill out this survey (https://go.cs61a.org/setup-survey) with any issues you might have faced in Lab 0 Python installation or if you used the Windows automated installer.

For quickly generating ok commands, you can now use the ok command generator (https://go.cs61a.org/ok-help).

Quick Logistics Review

Topics

Consult this section if you need a refresher on the material for this lab. It's okay to skip directly to the questions and refer back here should you get stuck.

Required Questions

What Would Python Display? (WWPD)

Q1: WWPD: Control

Use Ok to test your knowledge with the following "What Would Python Display?" questions:

```
python3 ok -q control -u
```

Hint: Make sure your `while` loop conditions eventually evaluate to a false value, or they'll never stop! Typing `Ctrl-C` will stop infinite loops in the interpreter.

```
>>> def xk(c, d):
...     if c == 4:
...         return 6
...     elif d >= 4:
...         return 6 + 7 + c
...     else:
...         return 25
>>> xk(10, 10)
______

>>> xk(10, 6)
______

>>> xk(4, 6)
______

>>> xk(0, 0)
______
```

```
>>> def how_big(x):
...     if x > 10:
...         print('huge')
...     elif x > 5:
...         return 'big'
...     elif x > 0:
...         print('small')
...     else:
...         print("nothing")
>>> how_big(7)
______

>>> how_big(12)
______

>>> how_big(1)
______

>>> how_big(-1)
______
```

```
>>> n = 3
>>> while n >= 0:
...     n -= 1
...     print(n)
______
```

Hint: Make sure your `while` loop conditions eventually evaluate to a false value, or they'll never stop! Typing `Ctrl-C` will stop infinite loops in the interpreter.

```
>>> positive = 28
>>> while positive:
...    print("positive?")
...    positive -= 3
______
```

```
>>> positive = -9
>>> negative = -12
>>> while negative:
...     if positive:
...         print(negative)
...     positive += 3
...     negative += 3
______
```

Q2: WWPD: Veritasiness

Use Ok to test your knowledge with the following "What Would Python Display?" questions:

```
python3 ok -q short-circuit -u
```

```
>>> True and 13
______

>>> False or 0
______

>>> not 10
______

>>> not None
______
```

```
>>> True and 1 / 0 and False
------

>>> True or 1 / 0 or False
------

>>> True and 0
------

>>> False or 1
------

>>> 1 and 3 and 6 and 10 and 15
------

>>> -1 and 1 > 0
------

>>> 0 or False or 2 or 1 / 0
------
```

```
>>> not 0
------

>>> (1 + 1) and 1
------

>>> 1/0 or True
------

>>> (True or False) and False
------
```

Q3: Debugging Quiz

The following is a quick quiz on different debugging techniques that will be helpful for you to use in this class. You can refer to the debugging article (/articles/debugging/) to answer the questions.

Use Ok to test your understanding:

```
python3 ok -q debugging-quiz -u
```

Coding Practice

Q4: Falling Factorial

Let's write a function `falling`, which is a "falling" factorial that takes two arguments, `n` and `k`, and returns the product of `k` consecutive numbers, starting from `n` and working downwards. When `k` is 0, the function should return 1.

```
def falling(n, k):
    """Compute the falling factorial of n to depth k.

    >>> falling(6, 3)  # 6 * 5 * 4
    120
    >>> falling(4, 3)  # 4 * 3 * 2
    24
    >>> falling(4, 1)  # 4
    4
    >>> falling(4, 0)
    1
    """
    total, stop = 1, n-k
    while n > stop:
        total, n = total*n, n-1
    return total
```

Use Ok to test your code:

```
python3 ok -q falling
```

Q5: Sum Digits

Write a function that takes in a nonnegative integer and sums its digits. (Using floor division and modulo might be helpful here!)

```
def sum_digits(y):
    """Sum all the digits of y.

    >>> sum_digits(10) # 1 + 0 = 1
    1
    >>> sum_digits(4224) # 4 + 2 + 2 + 4 = 12
    12
    >>> sum_digits(1234567890)
    45
    >>> a = sum_digits(123) # make sure that you are using return rather than print
    >>> a
    6
    """
    total = 0
    while y > 0:
        total, y = total + y % 10, y // 10
    return total
```

Use Ok to test your code:

```
python3 ok -q sum_digits
```

Submit

Make sure to submit this assignment by running:

```
python3 ok --submit
```

Reminder: Please fill out the Lab 0 setup survey (also included at the beginning of this assignment): here (https://go.cs61a.org/setup-survey).

Extra Practice

These questions are optional and will not affect your score on this assignment. However, they are **great practice** for future assignments, projects, and exams. Attempting these questions can be valuable in helping cement your knowledge of course concepts.

Q6: WWPD: What If?

Use Ok to test your knowledge with the following "What Would Python Display?" questions:

```
python3 ok -q if-statements -u
```

Hint: `print` (unlike `return`) does *not* cause the function to exit.

```
>>> def ab(c, d):
...     if c > 5:
...         print(c)
...     elif c > 7:
...         print(d)
...     print('foo')
>>> ab(10, 20)
______
```

```
>>> def bake(cake, make):
...     if cake == 0:
...         cake = cake + 1
...         print(cake)
...     if cake == 1:
...         print(make)
...     else:
...         return cake
...     return make
>>> bake(0, 29)
______

>>> bake(1, "mashed potatoes")
______
```

Q7: Double Eights

Write a function that takes in a number and determines if the digits contain two adjacent 8s.

```
def double_eights(n):
    """Return true if n has two eights in a row.
    >>> double_eights(8)
    False
    >>> double_eights(88)
    True
    >>> double_eights(2882)
    True
    >>> double_eights(880088)
    True
    >>> double_eights(12345)
    False
    >>> double_eights(80808080)
    False
    """
    prev_eight = False
    while n > 0:
        last_digit = n % 10
        if last_digit == 8 and prev_eight:
            return True
        elif last_digit == 8:
            prev_eight = True
        else:
            prev_eight = False
        n = n // 10
    return False
```

Use Ok to test your code:

```
python3 ok -q double_eights
```