Lab 1: Variables & Functions, Control

实验链接：Lab 1: Variables & Functions, Control
如何下载实验压缩包：

```
wget https://inst.eecs.berkeley.edu/~cs61a/sp21/lab/lab01/lab01.zip
```

What Would Python Display(WWPD)? (Part 1)

Q1: WWPD: Control

使用如下命令进行测试：

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

需要注意：

- 若用整型或浮点型做控制语句的判断条件， `0` 会被当作 `False` ， `非0` 会被当作 `True`

测试过程如下：

```
=====================================================================
Assignment: Lab 1
OK, version v1.18.1
=====================================================================

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Unlocking tests

At each "? ", type what you would expect the output to be.
Type exit() to quit

---------------------------------------------------------------------
Control > Suite 1 > Case 1
(cases remaining: 5)

What would Python display? If you get stuck, try it out in the Python
interpreter!

>>> def xk(c, d):
...     if c == 4:
...         return 6
...     elif d >= 4:
...         return 6 + 7 + c
...     else:
...         return 25
>>> xk(10, 10)
? 23
-- OK! --

>>> xk(10, 6)
? 19
-- Not quite. Try again! --

? 23
-- OK! --
```

```
>>> xk(4, 6)
? 6
-- OK! --

>>> xk(0, 0)
? 25
-- OK! --

---------------------------------------------------------------------
Control > Suite 1 > Case 2
(cases remaining: 4)

What would Python display? If you get stuck, try it out in the Python
interpreter!

>>> def how_big(x):
...     if x > 10:
...         print('huge')
...     elif x > 5:
...         return 'big'
...     elif x > 0:
...         print('small')
...     else:
...         print("nothin")
>>> how_big(7)
? big
-- Not quite. Try again! --

? 'big'
-- OK! --

>>> how_big(12)
? 'huge'
-- Not quite. Try again! --

? huge
-- OK! --

>>> how_big(1)
? small
-- OK! --

>>> how_big(-1)
? nothin
-- OK! --

---------------------------------------------------------------------
Control > Suite 2 > Case 1
(cases remaining: 3)

What would Python display? If you get stuck, try it out in the Python
interpreter!

>>> n = 3
>>> while n >= 0:  # If this loops forever, just type Infinite Loop
...     n -= 1
...     print(n)
(line 1)? 2
(line 2)? 1
(line 3)? 0
(line 4)? -1
```

```
-- OK! --

---------------------------------------------------------------------
Control > Suite 2 > Case 2
(cases remaining: 2)

What would Python display? If you get stuck, try it out in the Python
interpreter!

>>> positive = 28
>>> while positive: # If this loops forever, just type Infinite Loop
...     print("positive?")
...     positive -= 3
? positive
-- Not quite. Try again! --

? positive?
-- Not quite. Try again! --

? Infinite Loop
-- OK! --

---------------------------------------------------------------------
Control > Suite 2 > Case 3
(cases remaining: 1)

What would Python display? If you get stuck, try it out in the Python
interpreter!

>>> positive = -9
>>> negative = -12
>>> while negative: # If this loops forever, just type Infinite Loop
...     if positive:
...         print(negative)
...     positive += 3
...     negative += 3
(line 1)? Infinite Loop
-- Not quite. Try again! --

(line 1)? -12
(line 2)? -9
(line 3)? -6
-- OK! --

---------------------------------------------------------------------
OK! All cases for Control unlocked.
```

Q2: WWPD: Veritasiness

使用如下命令进行测试：

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

需要注意：

- `and` 和 `or` 运算符有 `短路性`

测试过程如下：

```
=====================================================================
Assignment: Lab 1
```

```
OK, version v1.18.1
=====================================================================

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Unlocking tests

At each "? ", type what you would expect the output to be.
Type exit() to quit

---------------------------------------------------------------------
Veritasiness > Suite 1 > Case 1
(cases remaining: 3)

What would Python display? If you get stuck, try it out in the Python
interpreter!

>>> True and 13
? 13
-- OK! --

>>> False or 0
? 0
-- OK! --

>>> not 10
? False
-- OK! --

>>> not None
? True
-- OK! --

---------------------------------------------------------------------
Veritasiness > Suite 1 > Case 2
(cases remaining: 2)

What would Python display? If you get stuck, try it out in the Python
interpreter!

>>> True and 1 / 0 and False  # If this errors, just type Error.
? Error
-- OK! --

>>> True or 1 / 0 or False  # If this errors, just type Error.
? True
-- OK! --

>>> True and 0  # If this errors, just type Error.
? 0
-- OK! --

>>> False or 1  # If this errors, just type Error.
? 1
-- OK! --

>>> 1 and 3 and 6 and 10 and 15  # If this errors, just type Error.
? 15
-- OK! --

>>> -1 and 1 > 0 # If this errors, just type Error.
? True
```

```
-- OK! --

>>> 0 or False or 2 or 1 / 0  # If this errors, just type Error.
? 2
-- OK! --

---------------------------------------------------------------------
Veritasiness > Suite 2 > Case 1
(cases remaining: 1)

What would Python display? If you get stuck, try it out in the Python
interpreter!

>>> not 0
? True
-- OK! --

>>> (1 + 1) and 1  # If this errors, just type Error. If this is blank, just
type Nothing.
? 1
-- OK! --

>>> 1/0 or True  # If this errors, just type Error. If this is blank, just type
Nothing.
? Error
-- OK! --

>>> (True or False) and False  # If this errors, just type Error. If this is
blank, just type Nothing.
? False
-- OK! --

---------------------------------------------------------------------
OK! All cases for Veritasiness unlocked.

Cannot backup when running ok with --local.
```

Q3: Debugging Quiz!

报错信息汇总与调试方法：Debugging
使用如下命令进行测试：

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

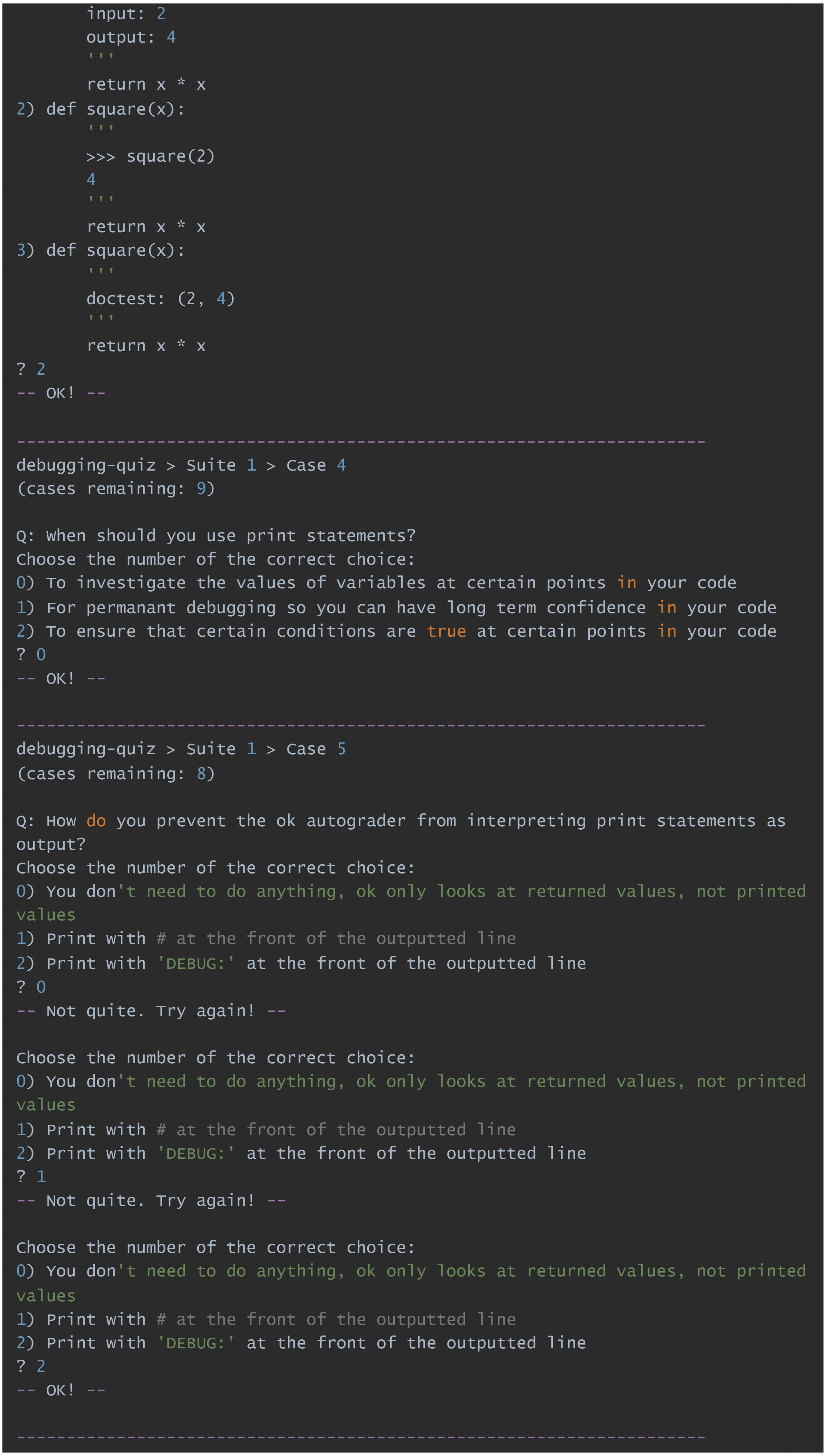
需要注意：

- `Traceback` 中，越后打印的函数就是越晚调用的
- 在输出的内容前加上 `"DEBUG: "`， `ok` 测评器会忽略改行输出
- 在程序运行出错时，中止程序，抛出异常，好过打印错误信息

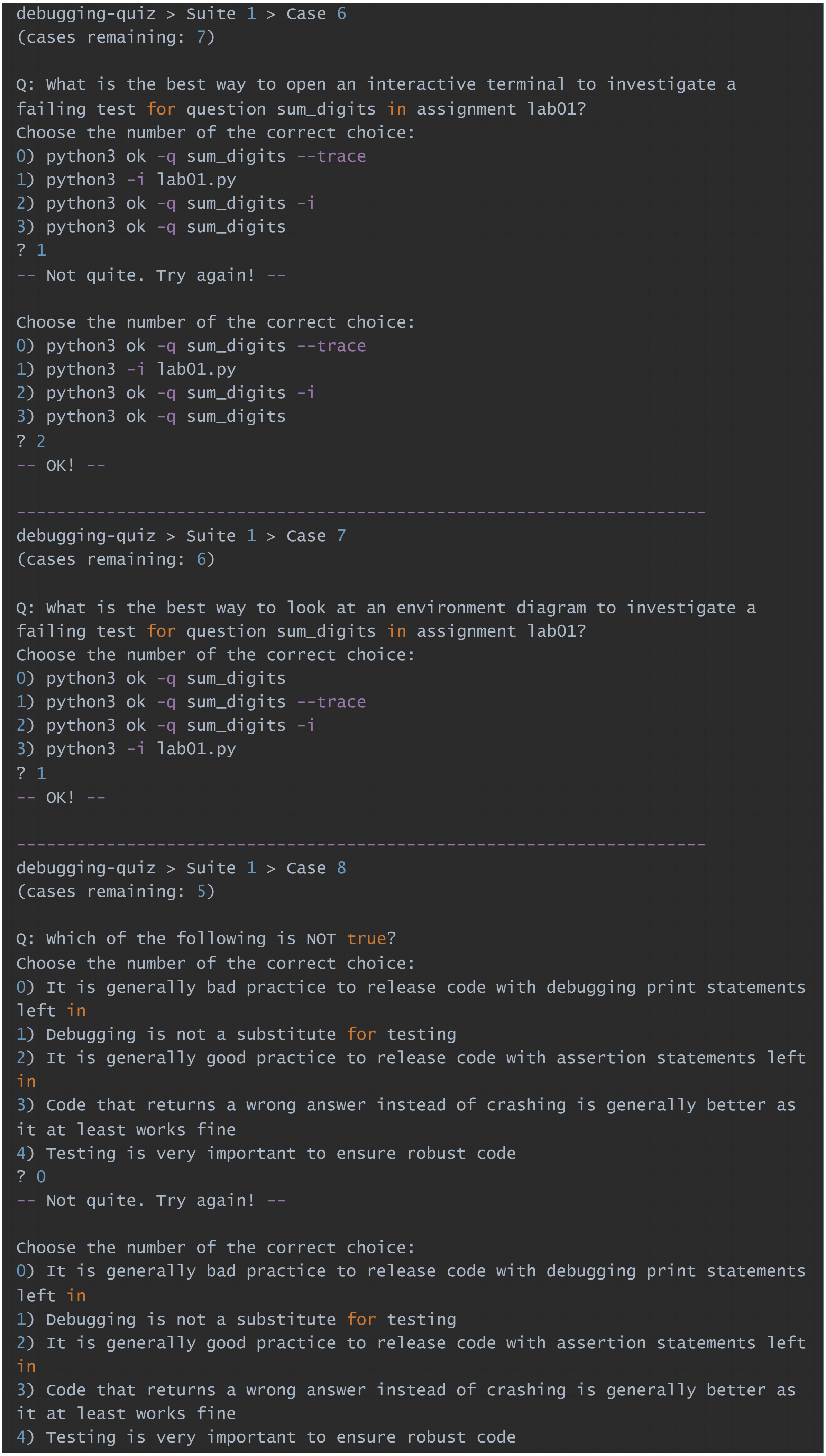
测试过程如下：

```
=====================================================================
Assignment: Lab 1
OK, version v1.18.1
=====================================================================

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Unlocking tests

```

```
At each "? ", type what you would expect the output to be.
Type exit() to quit

---------------------------------------------------------------------
debugging-quiz > Suite 1 > Case 1
(cases remaining: 12)

Q: In the following traceback, what is the most recent function call?
Traceback (most recent call last):
    File "temp.py", line 10, in <module>
      f("hi")
    File "temp.py", line 2, in f
      return g(x + x, x)
    File "temp.py", line 5, in g
      return h(x + y * 5)
    File "temp.py", line 8, in h
      return x + 0
  TypeError: must be str, not int
Choose the number of the correct choice:
0) g(x + x, x)
1) h(x + y * 5)
2) f("hi")
? 1
-- OK! --

---------------------------------------------------------------------
debugging-quiz > Suite 1 > Case 2
(cases remaining: 11)

Q: In the following traceback, what is the cause of this error?
Traceback (most recent call last):
    File "temp.py", line 10, in <module>
      f("hi")
    File "temp.py", line 2, in f
      return g(x + x, x)
    File "temp.py", line 5, in g
      return h(x + y * 5)
    File "temp.py", line 8, in h
      return x + 0
  TypeError: must be str, not int
Choose the number of the correct choice:
0) the code looped infinitely
1) there was a missing return statement
2) the code attempted to add a string to an integer
? 2
-- OK! --

---------------------------------------------------------------------
debugging-quiz > Suite 1 > Case 3
(cases remaining: 10)

Q: How do you write a doctest asserting that square(2) == 4?
Choose the number of the correct choice:
0) def square(x):
       '''
       square(2)
       4
       '''
       return x * x
1) def square(x):
       '''
```



```
        input: 2
        output: 4
        '''
        return x * x
2) def square(x):
        '''
        >>> square(2)
        4
        '''
        return x * x
3) def square(x):
        '''
        doctest: (2, 4)
        '''
        return x * x
? 2
-- OK! --

---------------------------------------------------------------------
debugging-quiz > Suite 1 > Case 4
(cases remaining: 9)

Q: When should you use print statements?
Choose the number of the correct choice:
0) To investigate the values of variables at certain points in your code
1) For permanant debugging so you can have long term confidence in your code
2) To ensure that certain conditions are true at certain points in your code
? 0
-- OK! --

---------------------------------------------------------------------
debugging-quiz > Suite 1 > Case 5
(cases remaining: 8)

Q: How do you prevent the ok autograder from interpreting print statements as
output?
Choose the number of the correct choice:
0) You don't need to do anything, ok only looks at returned values, not printed
values
1) Print with # at the front of the outputted line
2) Print with 'DEBUG:' at the front of the outputted line
? 0
-- Not quite. Try again! --

Choose the number of the correct choice:
0) You don't need to do anything, ok only looks at returned values, not printed
values
1) Print with # at the front of the outputted line
2) Print with 'DEBUG:' at the front of the outputted line
? 1
-- Not quite. Try again! --

Choose the number of the correct choice:
0) You don't need to do anything, ok only looks at returned values, not printed
values
1) Print with # at the front of the outputted line
2) Print with 'DEBUG:' at the front of the outputted line
? 2
-- OK! --

---------------------------------------------------------------------
```



```
debugging-quiz > Suite 1 > Case 6
(cases remaining: 7)

Q: What is the best way to open an interactive terminal to investigate a
failing test for question sum_digits in assignment lab01?
Choose the number of the correct choice:
0) python3 ok -q sum_digits --trace
1) python3 -i lab01.py
2) python3 ok -q sum_digits -i
3) python3 ok -q sum_digits
? 1
-- Not quite. Try again! --

Choose the number of the correct choice:
0) python3 ok -q sum_digits --trace
1) python3 -i lab01.py
2) python3 ok -q sum_digits -i
3) python3 ok -q sum_digits
? 2
-- OK! --

---------------------------------------------------------------------
debugging-quiz > Suite 1 > Case 7
(cases remaining: 6)

Q: What is the best way to look at an environment diagram to investigate a
failing test for question sum_digits in assignment lab01?
Choose the number of the correct choice:
0) python3 ok -q sum_digits
1) python3 ok -q sum_digits --trace
2) python3 ok -q sum_digits -i
3) python3 -i lab01.py
? 1
-- OK! --

---------------------------------------------------------------------
debugging-quiz > Suite 1 > Case 8
(cases remaining: 5)

Q: Which of the following is NOT true?
Choose the number of the correct choice:
0) It is generally bad practice to release code with debugging print statements
left in
1) Debugging is not a substitute for testing
2) It is generally good practice to release code with assertion statements left
in
3) Code that returns a wrong answer instead of crashing is generally better as
it at least works fine
4) Testing is very important to ensure robust code
? 0
-- Not quite. Try again! --

Choose the number of the correct choice:
0) It is generally bad practice to release code with debugging print statements
left in
1) Debugging is not a substitute for testing
2) It is generally good practice to release code with assertion statements left
in
3) Code that returns a wrong answer instead of crashing is generally better as
it at least works fine
4) Testing is very important to ensure robust code
```

```
? 3
-- OK! --

----------------------------------------------------------------------
debugging-quiz > Suite 1 > Case 9
(cases remaining: 4)

Q: You get a SyntaxError. What is most likely to have happened?
Choose the number of the correct choice:
0) You typed a variable name incorrectly
1) You forgot a return statement
2) Your indentation mixed tabs and spaces
3) You had an unmatched parenthesis
? 3
-- OK! --

----------------------------------------------------------------------
debugging-quiz > Suite 1 > Case 10
(cases remaining: 3)

Q: You get a IndentationError. What is most likely to have happened?
Choose the number of the correct choice:
0) You had an unmatched parenthesis
1) You typed a variable name incorrectly
2) Your indentation mixed tabs and spaces
3) You forgot a return statement
? 2
-- OK! --

----------------------------------------------------------------------
debugging-quiz > Suite 1 > Case 11
(cases remaining: 2)

Q: You get a TypeError: ... 'NoneType' object is not ... . What is most likely
to have happened?
Choose the number of the correct choice:
0) You typed a variable name incorrectly
1) Your indentation mixed tabs and spaces
2) You had an unmatched parenthesis
3) You forgot a return statement
? 3
-- OK! --

----------------------------------------------------------------------
debugging-quiz > Suite 1 > Case 12
(cases remaining: 1)

Q: You get a NameError. What is most likely to have happened?
Choose the number of the correct choice:
0) You had an unmatched parenthesis
1) You typed a variable name incorrectly
2) You forgot a return statement
3) Your indentation mixed tabs and spaces
? 1
-- OK! --

----------------------------------------------------------------------
OK! All cases for debugging-quiz unlocked.

Cannot backup when running ok with --local.
```

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.

思路：简单的循环语法题，记得给 `ans` 赋初值
实现代码如下：

```
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
    """
    "*** YOUR CODE HERE ***"
    ans = 1
    while k > 0:
        ans *= n
        n -= 1
        k -= 1
    return ans
```

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!)

思路：简单的循环语法题，把每位的数加在一起，得到答案
实现代码如下：

```
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
    """
    "*** YOUR CODE HERE ***"
    sum = 0
    while y > 0:
        sum += y % 10
        y //= 10
    return sum
```

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 is valuable in helping cement your knowledge of course concepts, and it's fun!

Q6: WWPD: What If?

使用如下命令进行测试：

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

需要注意：
`print()` 打印字符串不会有 `"` 或 `'` ，但字符串做返回值会有 `"` 或 `'` 。
测试过程如下：

```
=====================================================================
Assignment: Lab 1
OK, version v1.18.1
=====================================================================

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Unlocking tests

At each "? ", type what you would expect the output to be.
Type exit() to quit

---------------------------------------------------------------------
What If? > Suite 1 > Case 1
(cases remaining: 2)

What would Python display? If you get stuck, try it out in the Python
interpreter!

>>> def ab(c, d):
...     if c > 5:
...         print(c)
...     elif c > 7:
...         print(d)
...     print('foo')
>>> ab(10, 20)
(line 1)? 10
(line 2)? foo
-- OK! --

---------------------------------------------------------------------
What If? > Suite 1 > Case 2
(cases remaining: 1)

What would Python display? If you get stuck, try it out in the Python
interpreter!

>>> 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)
(line 1)? 1
(line 2)? 29
(line 3)? 29
-- OK! --

>>> bake(1, "mashed potatoes")
(line 1)? mashed potatoes
(line 2)? "mashed potatoes"
-- OK! --

---------------------------------------------------------------------
OK! All cases for What If? unlocked.

Cannot backup when running ok with --local.
```

Q7: Double Eights

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

思路：检查 `n` 的低2位是否是 `88` ，然后除 `10` ，直到 `n` 变成个位数。
实现代码如下：

```
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
    """
    "*** YOUR CODE HERE ***"
    while n >= 10:
        if n % 100 == 88:
            return True
        n //= 10
    return False
```

使用如下命令测试：

```
python3 ok -q double_eights --local
```

```
=====================================================================
Assignment: Lab 1
OK, version v1.18.1
=====================================================================


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    1 test cases passed! No cases failed.

Cannot backup when running ok with --local.
```

最终测试（不包括 `Extra Practice` ）

使用如下命令测试：

```
python ok --local
```

测试结果如下：

```
=====================================================================
Assignment: Lab 1
OK, version v1.18.1
=====================================================================


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    22 test cases passed! No cases failed.

Cannot backup when running ok with --local.
```