

Control Structures: exam-level questions

If you need help reviewing Control Structures, take a look at these resources:

- Albert and Robert's slides (https://docs.google.com/presentation/d/1p6rgyp4_IPLT6mtOdyKQ1VsuZOuiqaUIVYSm2c7CCu8/edit#slide=id.ga20165374_0_406)
- Albert and Robert's slides part 2 (https://docs.google.com/presentation/d/11-75T8zaVP1V2rwADDDtyt77hX-LKleUu1hPMmfGXME/edit#slide=id.ga271ab36d_1_245)

Each question has a "Toggle Solution" button -- click it to reveal that question's solution.

What would Python print?

Question 1

The following code is loaded into the Python interpreter

```
def is_even(x):
    if x % 2 == 0:
        print('even')
    print('odd')
    return x - 1

def branch(x):
    if x > 5:
        print('one')
    elif x > 0:
        print('two')
    if x > 10:
        print('three')
    else:
        print('four')
    return x + 5
```

What would Python print for the following lines?

```
>>> a = is_even(4)
_____
>>> b = branch(20)
_____
>>> c = branch(3)
_____
>>> d = is_even(is_even(5))
_____
>>> e = branch(branch(3))
_____
```

Toggle Solution

Code-Writing questions

Question 2

Implement a function `is_ascending`, which takes in a number `n`. `is_ascending` returns `True` if the one's digit of `n` is less than or equal to the ten's digit, and the ten's digit is less than or equal to the hundred's digit, and so on. In other words, the digits of the number going from right to left must be in ascending order.

```
def is_ascending(n):
    """Returns True if the digits of N are in ascending order.

    >>> is_ascending(321)
    True
    >>> is_ascending(123)
    False
    >>> is_ascending(4432221)
    True
    >>> is_ascending(5492)
    False
    >>> is_ascending(5420)
    True
    """
    """*** YOUR CODE HERE ***"""
```

Toggle Solution

Question 3

Implement a function `count_one`, which takes in a number `n`, and returns the number of ones in the digits of `n`.

```
def count_one(n):
    """Counts the number of 1s in the digits of n

    >>> count_one(7007)
    0
    >>> count_one(123)
    1
    >>> count_one(161)
    2
    >>> count_one(1)
    1
    """
    """*** YOUR CODE HERE ***"""
```

Toggle Solution

Question 4

Implement a function `total_one`, which takes in a number `n`, and returns the number of ones in the digits of all numbers from 1 to `n`.

Hint: You can use the `count_one` function from above.

```
def total_ones(n):
    """Returns number of 1s in the digits of all numbers from 1 to
    n.

    >>> total_ones(10) # 1, 10 -> two 1s
    2
    >>> total_ones(15) # 1, 10, 11, 12, 13, 14, 15 -> eight 1s
    8
    >>> total_ones(21)
    13
    """
    """*** YOUR CODE HERE ***"""
```

Toggle Solution