CPSC 1301, Computer Science I Quiz

Quiz 6b

This is a timed test. You have thirty minutes to complete the test. Your deliverable will be a plain text file, that is, an ASCII file with a `.txt` file extension. When you finish the test, upload your deliverable to Canvas. Do not publish your answer to your `git` repository.

You may work with your study partner for this quiz. In fact, working together is strongly encouraged. If you work with a partner, you must each make a seperate submission for credit, but you must also include the names of both authors in your submission.

1 Instructions

As a reminder, the key word `return` instructs the interpreter to replace the function call with the result computed by the function. When the function is called, the interpreter evaluates the function with the given arguments, computes the return value as the result of the function, and replaces the called function in its entirety with the result. This means that the function call is completely interchangeable with the result that the function computes.

First, write a function named `hi_low()` that takes two integer arguments, compares them, and returns `HIGHER` if the first argument is higher than the second argument, returns `LOWER` if the first argument is lower than the second argument, and returns `EQUAL` if the first argument is equal to the second argument. See the example below.

Second, write a function named `get_middle()` that takes two integer arguments, representing the beginning and the ending of a list of consecutive integers. For example, `(1, 5)` represents 1, 2, 3, 4, 5, and `(1, 6)` represents 1, 2, 3, 4, 5, 6. See the example below. The function returns the middle number of the list. When the list contains an odd number of integers, the middle number will be the exact middle. When the list contains an even number of integers, the "middle" number will actually consist of the middle two numbers — in this case, select the number on the left, which will be the lower number.

2 Code Template

```
1  # quiz06b_lastname.py
2
3  def hi_low(x, y):
4      pass
5
6  def get_middle(x, y):
7      pass
8
9  print("This is quiz 06b")
10
11 print("\nTesting function hi_low()")
12 print("calling hi_low(%d, %d), result is: %s" % (5, 3, hi_low(5, 3)))
13 print("calling hi_low(%d, %d), result is: %s" % (5, 8, hi_low(5, 8)))
14 print("calling hi_low(%d, %d), result is: %s" % (5, 5, hi_low(5, 5)))
15
16 print("\nTesting function get_middle()")
17 print("calling get_middle(%d, %d), result is: %d" % (1, 3, get_middle(1, 3)))
18 print("calling get_middle(%d, %d), result is: %d" % (1, 4, get_middle(1, 4)))
19 print("calling get_middle(%d, %d), result is: %d" % (10, 33, get_middle(10, 33)))
20 print("calling get_middle(%d, %d), result is: %d" % (100, 304, get_middle(100, 304)))
```

3 Output

```
This is quiz 06b

Testing function hi_low()
calling hi_low(5, 3), result is: HIGHER
calling hi_low(5, 8), result is: LOWER
calling hi_low(5, 5), result is: EQUAL

Testing function get_middle()
calling get_middle(1, 3), result is: 2
calling get_middle(1, 4), result is: 2
calling get_middle(10, 33), result is: 21
calling get_middle(100, 304), result is: 202
```