Before-the-Coding Quiz-02

Name:___________ **SOLUTION** ______________

Note:

A // B is the integer that results from dividing B into A and rounding down (dropping the “remainder”).
A % B is the integer that is the remainder (“left over”) from A // B.

1. [**With your instructor**] What is:

   a. `17 // 3` **5**

   b. `17 % 3` **2**

   c. `25 // 6` **4**

   d. `25 % 6` **1**

2. [**With your instructor**] Consider the contrived code below.
   What will the code output (“print”) when executed?

Code

```
def square(x):
    print("x is:", x)
    return x ** 2

def main():
    x = square(2)
    y = square(x)
    print(square(y))
    print("x is:", x)

main()
```

Output

```
x is: 2
x is: 4
x is: 16
256
x is: 4
```

3. **[With your instructor]** Suppose that you are ***given*** (and should ***call*** in this exercise) a function ***largest_digit*** that takes an integer ***M*** and returns the largest digit in ***M***. In the box below, write a function called ***largest_in_cube*** that has a single parameter ***n*** and returns the largest digit in the *cube* of ***n***.

```
def largest_in_cube(n):
    cube = n ** 3
    return largest_digit(cube)
```

[The one-line and three-line versions are fine, too. I have chosen the two-line version here to get students started thinking about the “steps” in a solution.]

4. Consider the contrived code below. What will the code output (“print”) when executed?

Code

```
def main():
    print("hi")
    foo()
    x = bar()
    print(x // 3)
    print("bye")

def foo():
    print("ok:", bar())
    print("foo")

def bar():
    print ("bar")
    return 16

main()
```

Output

```
hi
bar
ok: 16
foo
bar
5
bye
```

5. In the box below, write a function called ***back_and_forth*** that has two parameters – ***t*** that is a SimpleTurtle, and ***m*** that is an integer – and makes its SimpleTurtle go ***m*** pixels forward, then 3 * ***m*** pixels backward.

```
def back_and_forth(t, m):
    t.forward(m)
    t.backward(3 * m)
```

6. Consider the contrived code below. What will the code output (“print”) when executed?

Code

```
def main():
    print(foo(13, 2))
    print(foo(2, 13))

def foo(a, b):
    return (a % 4) + b

main()
```

Output

```
3
15
```