

# Name & Recitation Section:

Due **Thursday, Jan 13 at 2:10 PM in 34-101**. Please print out your code files (`homework.3.py`, `queue.py`, and any code you wrote for optional problems), and staple them to the back of these exercises before turning them in.

## Exercise 3.7 – Mutability

We've learned about many Python data structures (strings, lists, tuples, dictionaries). For both “mutable” and “immutable”, please give a short (5 words or fewer) definition, and then list what data structure(s) have that characteristic.

**Mutable:**

**Immutable:**

## Exercise 3.8 – Finding Bugs

The following set of instructions were given to Ben Bitdiddle, and he produced the code below. Find at least three bugs he made, and say how to fix them.

*Instructions:* Write a `negate` function that takes a number and returns the negation of that number. Also write a `large_num` function that takes a number, and returns `True` if that number is bigger than 10000, and `False` otherwise. Additionally, write some code to test your functions.

```
def negate(num):
    return -num

def large_num(num):
    res = (num > 10000)

negate(b)
neg_b = num
print 'b:', b, 'neg_b:', neg_b

big = large_num(b)
print 'b is big:', big
```

**Bugs:**

- 1.
- 2.
- 3.

## Exercise 3.9 – Mystery Program

Ben next turned in the following uncommented code to the 6.189 LAs. Help us figure out what it does!

```
1 print "Think of a number between 1 and 100, but don't tell me what you choose."
2 min_n = 1
3 max_n = 100
4 right_answer = False
5
6 while not right_answer:
7     mid_n = (max_n + min_n + 1)/2
8     answer = raw_input('Is it ' + str(mid_n) + '? ')
9     if answer[0] == 'y':
10        right_answer = True
11    elif answer.startswith('higher'):
12        min_n = mid_n + 1
13    elif answer.startswith('lower'):
14        max_n = mid_n - 1
15    else:
16        print "Sorry, I don't understand your answer."
17
18 print 'Woohoo! I got it!'
```

1. The `while` loop exits when the variable `right_answer` is `True`. What will cause `right_answer` to be true?
2. How many times will the program print out `'Woohoo! I got it!'`?
3. What are we using the variable `answer` for?
4. The program makes a guess in line 8. What user responses will be understood by the program after it makes its guess?
5. If the program gets the response `'higher'`, what does that tell it about its guess?
6. What are the variables `min_n`, `max_n` and `mid_n` used for?

This is an example of **binary search**, a simple but important algorithm in computer science. If you're curious, or confused, read the Wikipedia article on binary search to find out more and get a good explanation of what's going on here.



2. Consider the following code:

```
class Clock:

    def __init__(self, time):
        self.time = time

    def print_time(self):
        time = '6:30'
        print self.time

clock = Clock('5:30')
clock.print_time()
```

(a) What does the code print out? If you aren't sure, create a Python file and run it.

(b) Is that what you expected? Why?

3. Consider the following code:

```
class Clock:

    def __init__(self, time):
        self.time = time

    def print_time(self, time):
        print time

clock = Clock('5:30')
clock.print_time('10:30')
```

(a) What does the code print out? If you aren't sure, create a Python file and run it.

(b) What does this tell you about giving parameters the same name as object attributes?

4. Consider the following code:

```
class Clock:

    def __init__(self, time):
        self.time = time

    def print_time(self):
        print self.time

boston_clock = Clock('5:30')
paris_clock = boston_clock
paris_clock.time = '10:30'
boston_clock.print_time()
```

- (a) What does the code print out? If you aren't sure, create a Python file and run it.
- (b) Why does it print what it does? (Are `boston_clock` and `paris_clock` different objects? Why or why not?)

MIT OpenCourseWare  
<http://ocw.mit.edu>

6.189 A Gentle Introduction to Programming  
January IAP 2011

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.