INTRODUCTION TO COMPUTER SCIENCE I
Spring 2014
MID-TERM PREP
Sample problems to try

The mid-term exam is on Monday, so here are a handful of sample problems that you can try to solve (individually or in groups) as practice. If you can handle these, you're likely in good shape. The best preparation is to experiment with the code and structures that we've been using so that you feel familiar with them.

1 Sample problems

Here is a short description of some problem you should solve or code you should try to write. The best recommendation is that you sketch out a solution, and then open IDLE, test it, debug it, etc., until you see how to make it work properly. The exam will be paper-and-pen, and you might as well use the machine to verify the answers you come up with here.

1. Write a function `reverse_list (l)` that takes a list as a parameter and reverses its contents. (It should do so within the given list itself; it should not build a new list.)

2. Write a function `reverse_string (s)` that takes a string as a parameter and reverses its contents. (Since strings are immutable, the function must create a new string that contains the results and reverses it.)

3. Write a function `get_negative_number (prompt)` that prompts the user with the given string parameter, and then requires the user to enter some negative number (which can be a `float`). If the user enters a non-negative number, she should be prompted to try again. After 3 failed tries, the function should give up and return `None`.

4. Write a function `print_number_triangle(n)` that prints the following pattern of n rows and columns, like so (using an example of $n = 5$):

```
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

5. Consider the *Fibonacci sequence*:

$$1, 1, 2, 3, 5, 8, 13, 21 \ldots$$

Write a function `fib(n)` that calculates and prints, assuming $n \geq 2$, the first n values in this sequence.

6. We all know that if you flip a coin 10 times, you expect to get 5 heads and 5 tails, or close to that. We also know that there's some variation, and once in a while, you'll get 8 and 2, or 9 and 1, etc.

   Write a function `flip_test(n)` that does n repetitions of 10 coin flips. Have it count the number of heads and store, in a list, how many times each result occurs (e.g., heads occured 0 times, 1 time, 2 times, etc.). Have the function return that distribution so that you can print it and see how many times each case occurred out of n.

7. Imagine that the `int()` function has disappeared! We need to recreate it. Assume that all that you do have is a simpler function, `char_to_int(c)`, that coverts a single character `c`, if `c` is a digit from `'0'` to `'9'`, to its corresponding integer value (i.e., `0` to `9`). Assuming that this function also returns `None` if `c` is not one of those digits.

   Write a function `str_to_int(s)` that converts any string that represents an integer (e.g., `s = '9013'`) into the actual integer itself (e.g., `9013`). Bonus points if you can handle a leading negative sign (e.g., `s = '-136'` is converted to `-136`.)

   Hint: In case you want to test this solution, here is some code for `char_to_int(c)`; don't freak out if you don't fully know what it's doing...

```
def char_to_int (c):
    if len(c) != 1 or not '0' <= c <= '9':
        return None
    else:
        return ord(c) - ord('0')
```

2 Submitting your work

There is nothing to submit. You don't have to do these problems; they are merely a helpful aid in preparing for the mid-term.