

Quiz 4

Name: _____

Question 1 (20 points). Each of the following Python programs prints "Hello" some number of times. For each program, indicate the number of times it prints "Hello" and then *briefly justify your answer*. Hint: try working out how the program runs by writing a stack diagram on a piece of scrap paper.

(a).

```
1 def printHello(n):
2     for i in range(n):
3         print "Hello"
4 printHello(3)
5 printHello(2)
6 printHello(0)
```

(b).

```
1 def maybePrintHello(n):
2     if n % 2 == 0:
3         print "Hello"
4     n+=1
5 x=0
6 maybePrintHello(x)
7 maybePrintHello(x)
```

Question 2 (30 points). Write a *function* called `find_multiple`. This function should take two arguments: `lst`, a list of integers; and `n`, an integer. This function should return the *position* of the first element in the list `lst` which is evenly divisible by `n`. If no such element exists, this function should return `None`.

Question 3 (40 points). Consider the following program; then, answer the questions below. You must answer at least five parts of this question correctly to receive full credit.

```
1 def fib(lst, how_many):
2     while len(lst) < how_many:
3         second_to_last = len(lst)-2
4         last = len(lst)-1
5         # scope question here
6         lst.append(lst[second_to_last] + lst[last])
7     # stack trace question here
8     return lst
9 def main():
10    data1 = ["a","b"]
11    fib(data1, 5) # first call
12    print data1
13    data2 = [0, 1]
14    fib(data2, 5) # second call
15    print data2
16 main()
```

- (a). What type of data is stored in the variable `second_to_last`?
- (b). Which variables are in scope on line 5?
- (c). Draw a stack diagram of the program the *first* time it reaches line 7.
- (d). What are the elements of the list printed on line 12?
- (e). What are the elements of the list printed on line 15?
- (f). The variable `data1` in `main` does not change after line 10. Why doesn't line 12 print `["a", "b"]`?

Question 4 (10 points). The following function was intended to convert a list of strings to a new list of integers, putting `None` in the place of any string which cannot be converted. For instance, `int_list(["4", "apple", "80"])` should return `[4, None, 80]`. It does not do this. Explain what it does instead and then describe how to fix it.

```
1 def int_list(str_list):
2     accumulator = []
3     try:
4         for the_string in str_list:
5             accumulator.append(int(the_string))
6     except ValueError:
7         accumulator.append(None)
8     return accumulator
```