CSCI 150 HW: execution tracing practice

Due: Monday, April 2

To receive full credit, for each exercise you should do the following:

1. **Predict**: First, complete the exercise *without* using the Python interpreter. (You are welcome to refer to your notes or textbook, read Python documentation, look at examples from class, *etc.*; just don't actually run any code.) *Trace the execution of the code, keeping track of the function stack, all variables, and any output produced.* If you wish, you may use the provided tracing template.

2. **Check**: Run the code. Does the actual output agree with what you wrote down in step 1?

3. **Evaluate**: If your answer to part 1 was different than the actual output, keep experimenting with it, consult the textbook or Python documentation, ask a friend or TA or professor, *etc.* until you can explain why the code works the way it does *and* what your misunderstanding(s) were in part 1. (You do not need to do anything for step 3 if the output agrees exactly with what you wrote in step 1.)

Turn in your answers and evaluations either electronically via the usual form, or on paper.

You should consider the code in each exercise separately from the other exercises.

1. Consider the functions defined below. What does `main()` print?

```python
def aaa(words: List[str]) -> Dict[str, int]:
    d = {}
    for w in words:
        if w[0] not in d:
            d[w[0]] = 0
        d[w[0]] += 1

    return d

def main():
    dict = aaa(['aardvark', 'apple', 'bear', 'fish', 'ant'])
    print(dict['a'])
```

2. Consider the functions defined below. What is printed by `main2()`?

```python
def bbb(d: Dict[str, str]) -> Dict[str, List[str]]:
    mystery = {}
    for word in d:
        if d[word] not in mystery:
            mystery[d[word]] = []
```

```
        mystery[d[word]].append(word)
    return mystery

def main2():
    wordmap = {'hi':'there', 'whoa':'there'}
    wordmap['whoa'] = 'nelly'
    wordmap['you']  = 'there'
    print(bbb(wordmap))
```

3. Consider the functions defined below. What is printed by `main3()`?

```
def f(nums: List[int]) -> List[int]:
    nums[1] += 5
    nums = [6,7]
    nums[0] *= 2
    return nums

def g(ns: List[int]):
    print(ns)
    f(ns)
    print(ns)

def main3():
    mylist = [6,1,5,2]
    g([3] + mylist)
    print(mylist[0])
```