1) Are each of the following True or False (15 points)

1.1. In Python, assignment can change the value of an object. 

1.2. The expression \(1.0/16.0 + 1.0/16.0 == 1.0/8.0\) has the value True.

1.3. The sort method of list has a side effect.

1.4. In a bisection search, the number of points to be examined is reduced by one each iteration.

1.5. Returning None when the input argument to a function does not satisfy the criterion specified by the specification is a defensive programming technique.
2.1.) What does the following code print? (10 points)

```python
def f(L):
    L1 = []
    for i in L:
        for j in L1:
            if i == j:
                break
        L1.append(i)
    return len(L1)

L1 = [0, 1, 2, 3, 4]
L2 = []
for i in L1:
    if i%2 == 0:
        L2.append(i)
print L2
print f(L1 + L2)
```

```
[0, 2, 4]
8
```

2.2. Using big $\mathcal{O}$ notation, give the computational complexity of $f$. (10 points)

$\mathcal{O}(n^2)$, where $n = \text{len}(L)$
3) Write a Python program that satisfies the specification in the box. Your program should run in time that is approximately linear in the length of L. (15 points)

```python
def f(L):
    """assumes: L a list in which each element of type str
    returns: the number of distinct str’s in L.
    E.g., for L = ['ab', 'ba', 'ab', 'c'] it returns 3.""
    distinctVals = {}
    for s in L:
        distinctVals[s] = True
    return len(distinctVals)
```
4) What does the following code print? (10 points)

```python
def f(x):
    def g():
        x = 8
        x = x + 1
    g()
    return x

x = 3
z = f(x)
print 'x =', x
print 'z =', z
```

```
x = 3
z = 4
```
5) Write a **recursive** Python implementation of the function specified in the box. (19 points)

```python
def isPowerOf2(x):
    
    """Assumes: x is an int.
    Returns True if there exists a positive int, p, such that 2**p == x."""
    if x == 2:
        return True
    elif x <= 0 or x % 2 != 0:
        return False
    else:
        return isPowerOf2(x / 2)
```
6) Consider the following code:

```python
def f(s, d):
    for k in d.keys():
        d[k] = 0
    for c in s:
        if c in d:
            d[c] += 1
        else: d[c] = 0
    return d

def addUp(d):
    result = 0
    for k in d:
        result += d[k]
    return result

d1 = {}
d2 = d1
d1 = f('abbc', d1)
print addUp(d1)
d2 = f('bbcaa', d2)
print addUp(d2)
print f('', {})
print result
```

What appears on the screen when it is run? (12 points)

```
1
5
{}
NameError: name 'result' is not defined
```
7) Next to each item in the left column write the letter labeling the item in the right column that best matches the item in the left column. No item in the right column should be used more than once. (9 points)

- b) glass box testing
  - a) induction

- e) abstraction
  - b) implementation

- c) O(1)
  - c) constant time
  - d) linear time
  - e) specification
  - f) recursion

8) Do you think that the lectures are too slow paced, too fast paced, about right?

| Too slow | 1 | 2 | 3 | 4 | 5 | Too fast |

9) Do you think that the problem sets are too short, too long, about right?

| Too short | 1 | 2 | 3 | 4 | 5 | Too long |