This was the second midterm from 2015. I have added a few comments in bold type, like this.

Solutions are provided in a separate document. Be aware, when you are reading the solutions, that they were based on Python 2. The biggest difference (maybe the only one in this case) is that print statements in Python 2 do not contain parentheses, so you will see answers like

```python
print x
```

which in Python 3 would be

```python
print(x)
```

As usual, I advise you to work on these problems before you look at the solutions.
1. The following statements are executed:

\[ x = [3.5, -9, 2.0, 6.7, 4, 2] \]
\[ y = ((3, 7), (4, 2), (2, 3)) \]
\[ z = {1: ['one', 'un', 'uno', 'yi'], 2: ['two', 'deux', 'dos', 'er'], 3: ['three', 'trois', 'tres', 'san']} \]

Answer the following questions about \( x, y \) and \( z \). In some parts, if the question asks you what happens when a particular statement is executed, the right answer might be 'it results in an error'.

(a) What is the type of \( y \)?

(b) What is \( \text{sum}(x[:2]) \)?

(c) What are \( \text{len}(z) \), \( \text{len}(z[2]) \) and \( \text{len}(z[2][2]) \)? (Be careful!)

(d) What is the value of \( x \) after the following statement is executed?
\[ x = x + 3 \]

(e) What is the value of \( x \) after the following statement is executed?
\[ x.append(3) \]

(f) What is the value of \( y \) after executing the following statement? (Be very careful!)
\[ y += (5, 6) \]

(g) What is the value of \( y \) after executing the following statement?
\[ y[2] = (6, 2) \]

Assume for the next question that \( x, y \) and \( z \) are as given at the beginning of this problem (in other words, that the statements described in the questions above have not been executed).

(h) What are the type and value of the following expression?
\[ [z[w[0]][2] for w in y if w[0] in z] \]
2. Consider the function `prob2` defined below:

```python
def prob2(nums):
    u=[(int(math.sqrt(x)),x) for x in nums]
    v=[y for y in u if y[0]**2==y[1]]
    return [y[1] for y in v]
```

Suppose we call `prob2(x)`, where `x` is the list

\[11, 11, 17, 9, 3, 5, 8, 11, 9, 14, 14, 2, 3, 13, 13, 6, 1, 8, 7, 6\]

(a) What are the first three elements of the list `u` created in the first statement of the function?

(b) What are the first three elements of the list `v` created in the second statement of the function?

(c) What is the value returned by the function?

(d) Give a succinct description of what `prob2(nums)` returns in general, when `nums` is a list of integers.
3. Write a function

```python
def columnsums(v)
```

where v is a list of lists of integers (i.e., a two-dimensional array). For example

\[
[[2, -1, 3], [4, 1, 6], [1, 1, 0], [0, -4, 2]]
\]

which we think of as representing the array

\[
\begin{array}{ccc}
2 & 1 & 3 \\
4 & 1 & 6 \\
1 & 1 & 0 \\
0 & -4 & 2 \\
\end{array}
\]

Your function should return a list of the sums of the successive columns, so in the example above, the return value should be [7, -1, 11]. Your function should work with arrays of any width and height, but you may assume that it is only called with arrays that are properly formed (in the sense that all the rows have the same number of elements).
def binsearch(thelist, target):
    count = 0
    low = 0
    high = len(thelist) - 1
    while low <= high:
        mid = (low + high) / 2
        if thelist[mid] == target:
            return mid
        elif thelist[mid] < target:
            low = mid + 1
        else:
            high = mid - 1
    return -1

def binsearch_new(thelist, target):
    low = 0
    high = len(thelist) - 1
    while low <= high:
        mid = (low + high) / 2
        if thelist[mid] == target:
            return (thelist[mid], thelist[mid + 1])
        elif thelist[mid] < target:
            low = mid + 1
        else:
            high = mid - 1
            if low > mid:
                return (thelist[mid], thelist[low])
            else:
                return (thelist[high], thelist[mid])

Figure 1: The original binary search

Figure 2: The modified binary search
Note for the review: Problem 4 concerns two different implementations of binary search. These are different from what was demonstrated in class. The first (correct version) shown above returns the index of the item if it is found in the list, and returns -1 if the item is not in the list. What I demonstrated in class returns the index at which the item is located if it is in the list, but instead of -1, returns the index at which the item would have to be inserted if it were to be added to the list.

The second version tries to do something a bit similar—you have to figure that out!—but has a bug, which part (c) of the problem asks you to discover.
4. Displayed above is the original version of the binary search function for searching a sorted list, as presented in class, along with a modified version. As you may recall, the original implementation returns the index of the target in the list, if the target item is present, and returns -1 otherwise.

In the modified version the return values have been altered. What the resulting function does is actually quite useful, but it suffers from a serious defect, as you should discover in part (c) below.

Suppose the modified function is called with

\[1, 8, 27, 64, 125, 216, 343, 729, 1000]\n
as the first argument.

(a) What is the return value if the function is called with 343 as the second argument?

(b) ...with 600 as the second argument??

(c) ...with 1500 as the second argument?
5. We represent a collection of Boston-area restaurants as a Python dictionary $d_i$. Here is a typical entry (the restaurant is real---the data comes from Yelp).

"Sandrine's" : ('Cambridge', 'French', '$$$', 3.5)

That is, the key of each entry is the name of the restaurant, and the value field is a tuple consisting of: the location, the type of food, an expense rating (the 3 dollar signs), and an average of reviewer ratings (3.5 stars).

(a) My favorite restaurant is $Mom's$ $Eat$ $at$ $Joe's$. I would like to see if others agree with me, so I want to write a Python statement or statements that will print out the star rating of this restaurant. The code you write should not cause a Python-generated error message on the chance that this restaurant is not in the database, but instead print 'Restaurant not found'.

(b) I want to impress someone by taking them to a fancy Italian place in the North End. (I don’t care if the restaurant is good, I just want my guest to see me spending a lot of money.) Write a statement or statements that prints a list of the names of all the restaurants with location 'North End', food type 'Italian', and at least 4 dollar signs. This list should be sorted in alphabetical order.