1. (6 points) What is the value produced by executing each of the following statements?

   a. `2 / 5`

   b. `not ( (5 > len(range(5))) or ("mot" == "one") )`

   c. `"[a, b, c]" + "[d]"`

2. (7 points)

   a. What are the values of variables w, x, y, z at the end of executing the following

```
w = 1
x = 17
y = w + 1
z = [x, y]
y = y + w
z[0] = x + 1
z.append(y)
```

   b. If x has value 12, y has value 3, and z has value 5, what does the following print?

```
if ((y > z) or (z < x)):
   print("EK")
   if (x < 10):
       print("DO")
   elif (z < 10):
       print("TEEN")
   print("CHAAR")
else:
   if (x < 20):
      print("PAANCH")
   else:
      print("CHAH")
   print("SAAT")
if ((z-y) > 0):
   print("AATH")
```

3. (7 points) Write a function that takes as input a list of numbers and prints three things: the largest number in the list, the second largest number in the list, and the average of all the numbers in the list. Thus, `problem3([1, 7, -3, 3])` would print something like "Largest: 7, Second largest: 3, Average: 2"

   IMPORTANT NOTE: You are *not allowed* to use built-in list functions min, max, sort, or sum for this problem. You need to look through the list to find the apopriate items and do the math necessary to compute the average.

```
def problem3(list):
```

4. (7 points) Consider the function:

```
def problem4(listOfLists):
    result = []
    for item in listOfLists:
        result.append(len(item))
    return(result)
```

   a. What does problem4([["girl", "boy"], [0, 2, 7, 4], [[], 'x']]) return?

   b. In general, given a list of strings, what does the the function problem4 do?

   c. Rewrite the problem4 function using a while loop instead of a for loop.

```
def problem4c(listOfLists):
```

5. (7 points)

For this problem, we will represent a person using a four-item list. The first item is the person's name (a string). The second through fourth items are numbers ranging from 0 - 100 representing how happy, healthy, and wealthy that person is. Thus ["joe", 100, 50, 0] represents a person names joe that is completely happy, medium healthy, and not wealthy at all.

Given a big list of 4-item people lists, we want to get a sense of which trait is dominant in the most people. For joe above, the dominant trait is happiness. For ["jane", 0, 1, 0] all of the traits are low, but since the health score is highest, it is the dominant one.

Write a function, mostDominantTrait, that takes as input a list of 4-item people lists, and prints a message indicating what dominant trait occurs most often. NOTE: you may assume there are not ties (people always have three unique trait scores, and there will be different numbers of happy-, healthy-, wealthy-dominant people.

Thus, for the list [['a', 1, 2, 3], ['b, 1, 2, 4], ['c', 4, 2, 1]] there are two wealthy people, one happy person, and no healthy people. So, your program would print something like: "The dominant trait is wealthiness."

```
def mostDominantTrait(listOfPeopleLists):
```