PROBLEM 1:  (What is the output? (20 points))

A. (10 pts) What is the output of the following code segment? Write the output to the right. Note that there is only output for the print statements.

```
num = 1
x = 3
y = 1.2
print x + y
print x + num * 3
print type(num > 4)
print (num**2)/4
print x % 4
```

B. (10 pts) What is the output of the following code segment? Write the output to the right. Note that there is only output for the print statements.

```
month = "December"
print month[1]
print month[2:4] + month[-1]
print month.find("e")

alist = ["blue", "red", "green", "black"]
print alist[3]
print alist[:2]
```

PROBLEM 2:  (Temperatures and Prices - Simple Functions (14 points))

A. (6 pts) The Formula for converting a Fahrenheit temperature to a Celcius temperature is:

```
celciusTemp = (FahrenehitTemp - 32) * 5/9
```

Write the function `temperature` that has one float parameter `tempF`. This function returns the fahrenheit temperature as a string with F attached to the right end if the temperature in Fahrenheit is less than 100 degrees. Otherwise it calculates the degrees in Celcius and returns the temperature as a string in Celcius with the string C attached to the right end. Write the function below.
B. (8 pts) In the town of Hullaboo, there are rules for selling merchandise.

1. There is a tax of 10% on items, except there is no tax in the month of April.

2. If the day is Saturday or Sunday, there is a $10.00 discount on the item. This discount is taken after any tax. If any price is calculated as a negative number, then the price becomes positive.

Write the function `convertPrice` that has three parameters, `price`, the price of an item as a float, and `day` and `month`, both strings representing the day and month the item was purchased. For example:

```
call               returns         comment
convertPrice(8.00, "Monday", "July")  8.8     tax
convertPrice(8.00, "Sunday", "May")   1.2     tax, discount, and negative number
convertPrice(8.00, "Tuesday", "July") 8.80    tax
convertPrice(8.00, "Monday", "April")  8.0     no tax
convertPrice(4.00, "Saturday", "April") 6.0    no tax, discount and negative
```

def convertPrice(price, day, month):
    ",",
    returns price possibly modified based on rules above
    ",",
PROBLEM 3:  (It’s a mystery (14 points))

A. (6 pts) Consider the following list carColors and function countCarsWithColor that has two parameters clist, which is a list of strings of colors, and color, which is a string of one color.

carColors = ['red', 'blue', 'red', 'silver', 'blue']

def countCarsWithColor(clist, color):
    count = 0
    pos = 0
    while (pos < len(clist)):
        if (clist[pos] == color):
            count += 1
            pos += 1
    return count

This function is suppose to return the count of the number of times a color is in the list clist, but does not work as intended!

<table>
<thead>
<tr>
<th>call</th>
<th>should return</th>
<th>returns</th>
</tr>
</thead>
<tbody>
<tr>
<td>CountCarsWithColor(carColors, &quot;red&quot;)</td>
<td>2</td>
<td>never finishes</td>
</tr>
<tr>
<td>CountCarsWithColor(carColors, &quot;green&quot;)</td>
<td>0</td>
<td>never finishes</td>
</tr>
<tr>
<td>CountCarsWithColor(carColors, &quot;silver&quot;)</td>
<td>1</td>
<td>never finishes</td>
</tr>
</tbody>
</table>

Q1. Give an example list of strings of colors, and one string of color that when passed to CountCarsWithColor finishes and returns the correct answer.

Q2. Explain why this function does not work correctly.

Q3. Here is the code again. Modify the code so it works as intended.

    def countCarsWithColor(clist, color):
        count = 0
        pos = 0
        while (pos < len(clist)):
            if (clist[pos] == color):
                count += 1
                pos += 1
        return count

B. (8 pts) Consider the following mystery function with one parameter club which is a list of strings.
def mystery(club):
    1:     x = []
    2:     prev = club[0]
    3:     for item in club[1:]:
    4:         if prev[0] == item[0]:
    5:             x += [item]
    6:     prev = item
    7:     y = []
    8:     for item in x:
    9:         if len(item)>3:
    10:             y += [item]
    11:     return y[-1]

Consider making the call `mystery(club)` with the value of `club` below. Answer the following questions about tracing what happens with this call

`club = ["Sarah", "Sue", "Jack", "Abe", "Aaron", "Adam"]`

B1. What is the value of `x` on line 7?

B2. What is the value of `y` before line 11 executes?

B3. What value is returned from the call `mystery(club)`?

B4. Explain in words what `mystery` does.

B5. Give an example of a nonempty list of strings that when passed to `mystery` will crash when run. Give the line number where the code crashes.

**PROBLEM 4 : (Transformations (10 points))**

Write the function `swap` which has three string parameters: `word`, `let1` and `let2`, where the last two are each a single letter. This function returns the word transformed with the first occurrence of `let1` swapped with the first occurrence of `let2`. Assume that `let1` and `let2` are not equal, and that the first occurrence of `let1` and `let2` are not the first or last position in the string.

<table>
<thead>
<tr>
<th>call</th>
<th>returns</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>swap(&quot;computer&quot;, &quot;o&quot;, &quot;e&quot;)</td>
<td>&quot;computor&quot;</td>
<td>first &quot;o&quot; and first &quot;e&quot; swapped</td>
</tr>
<tr>
<td>swap('mississippi', 'i', 'p')</td>
<td>'mpssissiip'</td>
<td>first &quot;i&quot; and &quot;p&quot; swapped</td>
</tr>
</tbody>
</table>
PROBLEM 5:  (Who is in a CompSci 101 lab? (20 points))

Consider information about students who are in a CompSci 101 lab. Assume data is a
list of strings where each string represents 'firstName:lastName:sectNumber:level' where
sectNumber is a two digit string '01', '02', '03' or '04, and level is "fr" for first year, "so" for
sophomore, "jr" for junior and "sr" for senior.
Assume data has the following value for the examples.

```
        'Jas:Dou:01:jr', 'Sarp:Dang:03:fr',
```

A. (10 pts) Write the function `lastNamesInSect` which has three parameters, data, that
is a nonempty list of strings in the format above, and two strings, sect representing a lab
section number, and year representing a year such as "sr". This function returns a list of
the last names of people that have both the section number and year.

<table>
<thead>
<tr>
<th>call</th>
<th>returns</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>lastNamesInSect(data, &quot;03&quot;, &quot;fr&quot;)</td>
<td>['Du', 'Dang', 'Trinh', 'Daly']</td>
<td>last names of &quot;fr&quot; in lab sec 03</td>
</tr>
<tr>
<td>lastNamesInSect(data, &quot;02&quot;, &quot;so&quot;)</td>
<td>['Xu']</td>
<td>only one &quot;so&quot; in lab sec 02</td>
</tr>
</tbody>
</table>

```
def lastNamesInSect(data, sect, year):
```

B. (10 points) Write the function `lastNamesWithLetter` which has four parameters:

1. data, that is a list of strings in the format mentioned earlier, 'first-
Name:lastName:sectNumber:level' where level is "fr" for first year, "so" for sopho-
more", "jr" for junior and "sr" for senior
2. sect which is a two letter string representing a lab such as "01", "02", "03" or "04"
3. year which is a level "fr", "so", "jr" or "sr"
4. let which is a single capital letter such as "B"

This function returns the number of people from data who are a particular year, in a par-
ticular lab section, and whos last names starts with a specific letter.

In writing `lastNamesWithLetter` you may call `lastNamesInSect` that you wrote in Part A.
Assume `lastNamesInSect` works correctly.

```
def lastNamesWithLetter(data, sect, year, let):
```
<table>
<thead>
<tr>
<th>call</th>
<th>returns</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lastNamesWithLetter(data, &quot;03&quot;, &quot;fr&quot;, &quot;D&quot;)</code></td>
<td>3</td>
<td>3 firstyears in sect number 03 that start with &quot;D&quot;</td>
</tr>
<tr>
<td><code>lastNamesWithLetter(data, &quot;03&quot;, &quot;so&quot;, &quot;S&quot;)</code></td>
<td>0</td>
<td>no sophomores in sect number 03 that start with &quot;S&quot;</td>
</tr>
</tbody>
</table>