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.

Output:

```
num = 2
x = 7
y = 2.0
print num + 2.1
print x + num * 4
print type(y)
print (num ** 2) / 3
print x % 3
```

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.

Output:

```
month = "August"
print month[2]
print month[2:] + month[-1]
print month.find("gus")

alist = ["cat", "dog", "parrot", "fox"]
print alist[2]
print alist[1:]
```

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

A. (6 pts) Given two equations with $x$, Curious Pete wants to know if the equations are equal for a given value of $x$. The two equations are $2(x + 1)^2 - 1$ and $(2x + 1)^2$.

Write the function `equationsEqual` that has one int parameter `x`, that returns `True` if the two equations with the value $x$ are equivalent, and returns `False` if they are not equivalent.

<table>
<thead>
<tr>
<th>call</th>
<th>returns</th>
</tr>
</thead>
<tbody>
<tr>
<td>equationsEqual(0)</td>
<td>True</td>
</tr>
<tr>
<td>equationsEqual(1)</td>
<td>False</td>
</tr>
<tr>
<td>equationsEqual(2)</td>
<td>False</td>
</tr>
</tbody>
</table>
For example for the first call above when \( x = 0 \), \( 2((0) + 1)^2 - 1 = 1 \) and \( (2(0) + 1)^2 = 1 \), so True in this case.

```python
def equationsEqual(x):
    
    given one integer parameter x,
    return true if the two equations above with value x are equal
    otherwise return false
```

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 months of July and January.

2. If the day is Monday, there is a $5.00 discount on the item. This discount is taken after any tax. However, there is a minimum price of $1.00 for any item. If the discount reduces the item past the minimum price, then the item is sold for the minimum price.

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:

<table>
<thead>
<tr>
<th>call</th>
<th>returns</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>convertPrice(8.00, &quot;Monday&quot;, &quot;July&quot;)</code></td>
<td>3.00</td>
<td>no tax, subtract 5.00</td>
</tr>
<tr>
<td><code>convertPrice(3.00, &quot;Monday&quot;, &quot;July&quot;)</code></td>
<td>1.00</td>
<td>no tax, subtract 5.00 but use minimum</td>
</tr>
<tr>
<td><code>convertPrice(8.00, &quot;Tuesday&quot;, &quot;April&quot;)</code></td>
<td>8.80</td>
<td>8.00 + 10% tax</td>
</tr>
<tr>
<td><code>convertPrice(8.00, &quot;Monday&quot;, &quot;April&quot;)</code></td>
<td>3.80</td>
<td>8.00 + 10 % tax - 5.00</td>
</tr>
</tbody>
</table>

```python
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 `doHaveCarWithColor` that has two parameters `clist`, which is a list of strings of colors, and `color`, which is one color.

```python
carColors = ['red', 'blue', 'red', 'silver', 'blue']
def doHaveCarWithColor(clist, color):
    for carColor in clist:
        if carColor == color:
            return True
        else:
            return False
    return False
```

This function is suppose to return True if `color` is in the list `carColors`, but does not work as intended!

<table>
<thead>
<tr>
<th>call</th>
<th>returns</th>
<th>should return</th>
</tr>
</thead>
<tbody>
<tr>
<td>doHaveCarWithColor(carColors, &quot;red&quot;)</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>doHaveCarWithColor(carColors, &quot;green&quot;)</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>doHaveCarWithColor(carColors, &quot;silver&quot;)</td>
<td>False</td>
<td>True</td>
</tr>
</tbody>
</table>

Q1. Give another example of a call to this function with the list `carColors` above and a value for color that does not return the expected value.

```python
doHaveCarWithColor(carColors, )
```

Q2. Explain why this function does not work correctly.

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

```python
def doHaveCarWithColor(clist, color):
    for carColor in clist:
        if carColor == color:
            return True
        else:
            return False
    return False
```

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

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


B1. What is the value of x on line 5?

B2. What is the value of y before line 9 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 that when passed to mystery will crash when run. Explain why it crashes.

PROBLEM 4 : (Transformations (10 points))

Write the function stretchOut which has one string parameter word. This function returns the word transformed in the following way.

1. Make the word all lowercase

2. If the word starts with a vowel(a, A, e, E, i, I, o, O, u or U), or is less than three characters, then return the word with no change except all lowercase.

3. Otherwise, return the word with the first letter repeated twice, the second letter repeated twice and ”eh” added to the right end of the word.

You may call the function IsVowel(letter) where letter is a string of size 1, and returns True if the letter is ”a” or ”e” or ”i” or ”o” or ”u”. Otherwise it returns False.
def stretchOut(word):

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.


A. (10 pts) Write the function *listOfYears* which has two parameters, data, that is a nonempty list of strings in the format above, and sect which is a string representing a section number such as "01" or "02". This function returns a list of the years of all students in this section. For example.

<table>
<thead>
<tr>
<th>call</th>
<th>returns</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>listOfYears(data, &quot;03&quot;)</td>
<td>['fr', 'fr', 'fr', 'sr']</td>
<td>three firstyears and one senior</td>
</tr>
<tr>
<td>listOfYears(data, &quot;02&quot;)</td>
<td>['fr', 'so', 'sr']</td>
<td>one firstyear, one soph and one senior</td>
</tr>
</tbody>
</table>

def listOfYears(data, sect):

B. (10 points) Write the function *numberOf* which has three parameters:

1. data, that is a list of strings in the format mentioned earlier, 'firstName:lastName:sectNumber:level' where level is "fr" for first year, "so" for sophomore", "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"

This function returns the number of people from data who are a particular year and in a particular lab section.

In writing `numberOf` you may call `listOfYears` that you wrote in Part A. Assume `listOfYears` works correctly.

<table>
<thead>
<tr>
<th>call</th>
<th>returns</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>numberOf(data, &quot;03&quot;, &quot;fr&quot;)</code></td>
<td>3</td>
<td>3 firstyears in sect number 03</td>
</tr>
<tr>
<td><code>numberOf(data, &quot;03&quot;, &quot;so&quot;)</code></td>
<td>0</td>
<td>no sophomores in sect number 03</td>
</tr>
</tbody>
</table>

```python
def numberOf(data, sect, year):
```