Problem 1: Types and Values

Variables used in the table below (one space between words in string s):

```python
fruits = ["apple", "pear", "orange", "banana", "blueberry"]
s = "one, two, three o'clock, four o'clock rock"
```

Fill in the type and value for each expression or variable given in the left column. The first row is done as an example.

(24 points, 1 point each entry)

<table>
<thead>
<tr>
<th>variable/expression</th>
<th>type</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>a = s[5]</td>
<td>string</td>
<td>&quot;t&quot;</td>
</tr>
<tr>
<td>x = len(fruits)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>z = s.count&quot;,&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y = s.find&quot;z&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>w = fruits[2]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d = len(fruits) &gt; len(s)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>e = s.split()[1:3]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>g = 30/7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>h = 27 % 12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>q = 100 * 1.0 / 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t = fruits[0][-1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>u = int&quot;123&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>v = fruits[-1]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Problem 2: Simple Functions

Part A (8 points)

The volume of a pyramid with an equilateral triangle as the base has a volume given by the formula below where \( b \) is a side of the base and \( h \) is the height of the pyramid

\[
\frac{\sqrt{3}}{12} \times b^2 \times h
\]

Write the function \texttt{volume} that returns the volume of such a pyramid given two parameters: the length of one side of the base and the height. You can use 1.73205 as the value of \( \sqrt{3} \) in writing the code.

<table>
<thead>
<tr>
<th>call</th>
<th>return</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{volume(3,8)}</td>
<td>10.3923</td>
</tr>
<tr>
<td>\texttt{volume(1,10)}</td>
<td>1.44337</td>
</tr>
<tr>
<td>\texttt{volume(21.5,16.5)}</td>
<td>1100.8807</td>
</tr>
</tbody>
</table>

Note that \( 1.73205 \times \frac{72}{12} = 10.3923 \).

Formula for writing volume \( \frac{\sqrt{3}}{12} \times b^2 \times h \)

```python

def volume(side, height):
    
    # side is side of base of equilateral triangular pyramid
    # height is the height of the pyramid (both in inches)
    # return volume of the pyramid in cubic inches
    return (1/12) * (side**2) * (height)
```


Part B (12 points)

The zapf index of a word is an integer. A word's zapf index is given by the following rules.

- if the word is fewer than 4 characters long the zapf index is 2
- If the word ends in 's', the zapf index is 7
  - unless, if the word ends in 'ies', then the zapf index is 10
- if the word is more than 10 characters long, and does not end in 's', the zapf index is 5
- otherwise the zapf index is 13

<table>
<thead>
<tr>
<th>call</th>
<th>return</th>
</tr>
</thead>
<tbody>
<tr>
<td>zapfIndex(&quot;zebras&quot;)</td>
<td>7</td>
</tr>
<tr>
<td>zapfIndex(&quot;beauties&quot;)</td>
<td>10</td>
</tr>
<tr>
<td>zapfIndex(&quot;autobiography&quot;)</td>
<td>5</td>
</tr>
<tr>
<td>zapfIndex(&quot;its&quot;)</td>
<td>2</td>
</tr>
<tr>
<td>zapfIndex(&quot;amazing&quot;)</td>
<td>13</td>
</tr>
</tbody>
</table>

Write `zapfIndex` below, or on the next (blank) page

```python
def zapfIndex(word):
    ""
    return int: zapf index of word, a string
    ""
    word = word.lower()
```

Problem 3: Without Functions (8 points)

The average of a list of numbers is the sum of the numbers divided by how many numbers there are. For example, the average of [1,3,5,8] is 4.25 because 1 + 3 + 5 + 8 = 17, there are four numbers, and 17 divided by 4 is 4.25. The function below correctly returns the average of a list of numbers.

```python
def average(numbers):
    ""
    numbers is a list of int values, there is at least one int value in numbers
    returns the average of the values in numbers
    ""
    return sum(numbers)*1.0/len(numbers)
```
Rewrite the function without using the Python function `sum` and without using the Python function `len`. The function is started below

```python
def average(numbers):
    total = 0
    count = 0
    # you add code here before return statement
    return total*1.0/count
```

Problem 4: Transform

The function `isVowel` below returns true if it's parameter is a vowel, and returns false otherwise

```python
def isVowel(ch):
    ch = ch.lower()
    if ch in "aeiou":
        return True
    else:
        return False
```

Part A (8 points)

Write the function `noVowels` that returns a new string with every vowel in the string parameter replaced with an asterisk. Letters that aren't vowels remain the same in the new word.

<table>
<thead>
<tr>
<th>call</th>
<th>return</th>
</tr>
</thead>
<tbody>
<tr>
<td>noVowels(&quot;vowel&quot;)</td>
<td>&quot;v<em>w</em>l&quot;</td>
</tr>
<tr>
<td>noVowels(&quot;Elephant&quot;)</td>
<td>&quot;<em>l</em>ph*nt&quot;</td>
</tr>
<tr>
<td>noVowels(&quot;nsfw&quot;)</td>
<td>&quot;nsfw&quot;</td>
</tr>
</tbody>
</table>

```python
def noVowels(word):
```
Part B (8 points)

Write the function `newList` that takes a list of strings as a parameter and returns a new list that contains the same words as in the parameter, but words that begin with a vowel are replaced by words in which vowels are replaced by asterisks. Words that do not begin with a vowel are not changed. You must call the function `noVowels` and the function `isVowel`, assume they work correctly.

<table>
<thead>
<tr>
<th>call</th>
<th>return</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>newList([&quot;elvish&quot;, &quot;cats&quot;, &quot;fry&quot;])</code></td>
<td><code>[&quot;*lv*sh&quot;, &quot;cats&quot;, &quot;fry&quot;]</code></td>
</tr>
<tr>
<td><code>newList([&quot;frog&quot;, &quot;elbow&quot;, &quot;eel&quot;])</code></td>
<td><code>[&quot;frog&quot;, &quot;*lb*w&quot;, &quot;**l&quot;]</code></td>
</tr>
<tr>
<td><code>newList([&quot;Ouch&quot;, &quot;OWL&quot;, &quot;oOooOo&quot;])</code></td>
<td><code>[&quot;**ch&quot;, &quot;*WL&quot;, &quot;*****&quot;]</code></td>
</tr>
</tbody>
</table>

```python
def newList(words):
```

Problem 5: Taxi

The description is on this page, the problems are on the next two pages

You've been hired by a startup to market a Tip App. Rather than use 10% or 15% as might be the case with the software at Uber or in a Taxi company, the startup thinks the drivers will get more money if the users get to choose the kind of service rather than the tip rate. Here's a mock up of the screen a user sees:

**Tip Calculator**

- Terrible
- Mediocre
- Good
- Excellent
- Unbelievable!

The function `tipAmount` is called whenever the user clicks on one of these buttons, an intern has developed this function and code to test it as shown here:
def tipAmount(adjective):
    words = ["Terrible", "Mediocre", "Good", "Excellent", "Unbelievable!"]
    percentage = [5.0, 10.0, 15.0, 20.0, 30.0]

    if adjective not in words:
        return 10.0
    dex = words.index(adjective)
    return percentage[dex]

print tipAmount("Mediocre")
print tipAmount("Excellent")
print tipAmount("not so hot")
print tipAmount("Unbelievable!")

The output of the print statements is

    10.0
    20.0
    10.0
    30.0

Part A (8 points)
You are to write the function totalBill for the startup company. The function takes two parameters: An adjective from one of the buttons on the calculator screen and the fare to be paid. The function returns the fare plus the tip amount based on the adjective -- the total bill.

<table>
<thead>
<tr>
<th>call</th>
<th>return</th>
<th>reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>totalBill(&quot;Mediocre&quot;, 25)</td>
<td>27.5</td>
<td>25 + .10*25 = 27.5</td>
</tr>
<tr>
<td>totalBill(&quot;Unbelievable!&quot;,35)</td>
<td>45.5</td>
<td>35 + .30*35 = 45.5</td>
</tr>
<tr>
<td>totalBill(&quot;Terrible&quot;, 42.63)</td>
<td>44.76</td>
<td>42.63 + .05*42.63 = 44.76</td>
</tr>
</tbody>
</table>

Write the function totalBill. In calculating the tip amount you must call the function tipAmount shown above, you should not duplicate the functionality of the code in tipAmount in writing totalBill. You may not change the code in tipAmount.
def totalBill(adjective, fare):
    ...
    adjective is a string describing the ride, chosen by clicking on a button
    fare is a float for what the rider owes
    returns the total bill, which is fare + tip amount
    ...

Part B (8 points)

Describe in words the logic in how the function tipAmount works. Be general. Be sure to explain
A. The purpose of the if statement
B. The use of the function/method .index
C. How both lists are used together
D. Also explain in your description how to modify the code to add a tip of 50% for the adjective "Best Trip Ever", but keeping the function mostly the same

Problem 6: Dietary Restrictions

In this problem you'll write code to process a file of information regarding food and calories for a single serving of food. The data file has information on each line in the format:

food type, food name, calories-per-serving

where the first two entries on a line are strings and the third entry is an integer.

For example, here's a data file named 'food.txt':
meat, venison, 134  
meat, ground beef, 218  
meat, filet mignon, 179  
fruit, apple, 60  
fruit, banana, 80  
fruit, grapes, 90  
fruit, orange, 50  
fruit, watermelon, 54  
fish, salmon, 150  
fish, flounder, 100  
fish, catfish, 120  
snack, doritos, 120  
snack, ritz crackers, 80  
snack, pretzels, 107  
meat, rabbit, 131  
meat, pastrami, 198  
meat, sausage, 216

The function call `printData("food.txt")` prints the output below

```
[ 'meat', 'venison', '134' ]
[ 'meat', 'ground beef', '218' ]
[ 'meat', 'filet mignon', '179' ]
[ 'fruit', 'apple', '60' ]
[ 'fruit', 'banana', '80' ]
[ 'fruit', 'grapes', '90' ]
[ 'fruit', 'orange', '50' ]
[ 'fruit', 'watermelon', '54' ]
[ 'fish', 'salmon', '150' ]
[ 'fish', 'flounder', '100' ]
[ 'fish', 'catfish', '120' ]
[ 'snack', 'doritos', '120' ]
[ 'snack', 'ritz crackers', '80' ]
[ 'snack', 'pretzels', '107' ]
[ 'meat', 'rabbit', '131' ]
[ 'meat', 'pastrami', '198' ]
[ 'meat', 'sausage', '216' ]
```

The function `printData` is shown below, it generates the output above.
def printData(fname):
    f = open(fname)
    for st in f:
        data = st.strip().split(",")
        print data
    f.close()

Part A (8 points)

Write a function `category` that takes two parameters: the name of a data file and the name of a category like "meat" or "snack", The function returns a list of the foods in the given category that are stored in the data file.

<table>
<thead>
<tr>
<th>call</th>
<th>returns</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>category(&quot;food.txt&quot;, &quot;fruit&quot;)</code></td>
<td>['apple','banana','grapes','orange','watermelon']</td>
</tr>
<tr>
<td><code>category(&quot;food.txt&quot;,&quot;fish&quot;)</code></td>
<td>['salmon', 'flounder', 'catfish']</td>
</tr>
<tr>
<td><code>category(&quot;food.txt&quot;,&quot;stuff&quot;)</code></td>
<td>[ ]</td>
</tr>
</tbody>
</table>

def category(filename, category):

Part B (8 points)

Write the function `biggest` that returns the name of the food that has the most calories per serving as stored in the file passed to `heaviest`.

For example, `biggest("foods.txt")` should return "ground beef" since that has the most calories per serving, 218, of any food in the data file "foods.txt"

def biggest(filename):