20 Points – Do any 20; do more for extra credit. Correct answers are worth +1 point, blank answers are worth 0 points, but wrong answers are incur a −½ point penalty; if you don’t know an answer, leaving it blank is usually better than a bad guess. The following statements have all been executed:

\[
\begin{align*}
\text{Frodo} & = 4.5 \\
\text{Bilbo} & = 25 \\
\text{Sauron} & = [3, "EYE", 2.5, [7, 1, 5, 9], 17] \\
\text{Arwen} & = "EVENSTAR" \\
\text{Gollum} & = (5, 2, [6, 9, 4], 8)
\end{align*}
\]

Show the computed result for each problem; all are independent of one another. Indicate where a computation fails because of some form of error. Be careful about the type of the result, particularly int, float, long, bool, and complex types, and put proper quotes around string results, square brackets around lists, and parentheses around tuples.

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Frodo * 2</td>
<td>9.0 float</td>
</tr>
<tr>
<td>2. Frodo * 2.0</td>
<td>9.0 float</td>
</tr>
<tr>
<td>3. Bilbo / 2</td>
<td>12 int</td>
</tr>
<tr>
<td>4. Bilbo / 2.0</td>
<td>12.5 float</td>
</tr>
<tr>
<td>5. len(Sauron)</td>
<td>5 int</td>
</tr>
<tr>
<td>6. len(Gollum[2])</td>
<td>3 int</td>
</tr>
<tr>
<td>7. len(Bilbo)</td>
<td>ERROR len(int)</td>
</tr>
<tr>
<td>8. len(Sauron[1])</td>
<td>3 int</td>
</tr>
<tr>
<td>9. len(Sauron[3])</td>
<td>4 int</td>
</tr>
<tr>
<td>10. Sauron[-1]</td>
<td>17 int</td>
</tr>
<tr>
<td>11. Sauron[3][2]</td>
<td>5 int</td>
</tr>
<tr>
<td>12. Sauron[3]+[Gollum[0]]</td>
<td>[7,1,5,9,5] list</td>
</tr>
<tr>
<td>15. Sauron+Gollum</td>
<td>ERROR list+tuple</td>
</tr>
<tr>
<td>17. [Gollum[0],Bilbo,Arwen[1]]</td>
<td>[5,25,&quot;V&quot;] list</td>
</tr>
<tr>
<td>18. (Frodo,Bilbo,Sauron[1])</td>
<td>(4.5,25,&quot;EYE&quot;) tuple</td>
</tr>
<tr>
<td>20. Arwen[-1]</td>
<td>&quot;R&quot; string</td>
</tr>
<tr>
<td>21. range(Gollum[0])</td>
<td>[0,1,2,3,4] list</td>
</tr>
<tr>
<td>22. range(len(Gollum))</td>
<td>[0,1,2,3] list</td>
</tr>
<tr>
<td>23. range(3,Bilbo,4)</td>
<td>[3,7,11,15,19,23] list</td>
</tr>
<tr>
<td>24. [X*2 for X in range(5)]</td>
<td>[0,2,4,6,8] list</td>
</tr>
<tr>
<td>25. [0 for X in range(Sauron[0])]</td>
<td>[0,0,0] list</td>
</tr>
</tbody>
</table>
10 points – Show what is printed by the following code fragment for each given case:

<table>
<thead>
<tr>
<th>Y=4, Z=2</th>
<th>X = Y / Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y=4.0, Z=2</td>
<td>X = -1</td>
</tr>
<tr>
<td>Y=4.0, Z=0</td>
<td>(divide by zero error)</td>
</tr>
<tr>
<td>Y=2, Z=0.1</td>
<td>20.0</td>
</tr>
<tr>
<td>Y=2, Z=&quot;1&quot;</td>
<td>-1 (type error in division)</td>
</tr>
</tbody>
</table>

15 Points – Write a while-loop with an integer counter variable called `Count`, where `Count` starts at 13, is increased by 5 after each pass through the loop, and then ends if the variable ever reaches 47 or larger. The “payload” of the loop is to print `Count`.

```
Count = 13  # 3 points
while (Count < 47):
    print Count  # 5 points
    Count = Count + 5  # 2 points
    Can be: Count += 5
```

For each line, start with the given points and remove 1 point per syntax error. Remove 1 point overall if the print and the increment are reversed. Do not go below zero

20 Points – Show what is printed out as the result from calling `Main()` (four lines total):

```
def Frog (Cat,Dog,Bat=1):
    Newt = Cat - Dog
    return Newt * Bat

def Toad (Dog,Bunny):
    return Frog (Bunny,Dog)

def Newt (Cat,Bunny,Snake):
    return Toad (Cat,Snake) + Bunny

def Main():
    print Frog (5,2,3)  # Answer #1: 9 (5 points)
    print Toad (4,8)    # Answer #2: 4 (5 points)
    print Newt (3,2,4)  # Answer #3: 3 (5 points)
    print Frog (6,-1)   # Answer #4: 7 (5 points)
    return
```
15 Points – What is printed when the user enters *Dog* at the first prompt, and enters *3* at the second prompt?

```python
S = raw_input("Enter string to repeat --- ")
N = input("Enter number of repeats --- ")
Answer = ""
for I in range(N): Answer = Answer + S + str(I)
print Answer
```

Dog0Dog1Dog2

Remove 3 points if the range is more than or less than [0,1,2]
Remove 3 points if the Dog0, Dog1, and Dog2 are on different lines.
Remove 3 points if numbers appear to the left of each Dog.
Remove 3 points if Dog is not capitalized as shown.
Remove 3 points if there are any spaces between Dog and the numbers.

15 Points – Complete the following function to return -1 if integer parameter *N* is less than zero, return +1 if *N* is greater than zero, and return 0 if *N* is exactly zero. Your solution must **NOT** contain any print statements.

```python
def MyFunction(N):
    if (N < 0): return -1
    if (N > 0): return +1
    return 0
```

- or -

```python
def MyFunction(N):
    if (N < 0): Result = -1
    if (N > 0): Result = +1
    if (N == 0): Result = 0
    return Result
```

- or -

```python
def MyFunction(N):
    if (N < 0): Result = -1
    elif (N > 0): Result = +1
    else: Result = 0
    return Result
```

Remove 1 point per syntax error (indentation, capitalization, using = instead of ==, etc.).
Remove 5 points per major design error.
Spacing after the start of a statement is irrelevant.
5 Points – Short Answer – How are the following statements all similar? You may use the back of this page to answer.

```
N = N + 1    # where N is an int
S = S + "1"  # where S is a string
L = L + [1]  # where L is a list
T = T + (1)  # where T is a tuple
```

All four take a variable’s old value, make a change to that value, then replaces the variable’s value with the new result. (The last three append the new value onto the end of the string, list, or tuple, respectively.)

Accept anything reasonable.