25 Points – What is the value of each expression below? Answer any 25; answer more for extra credit. Variable \( S = "APPLE" \) and variable \( L = [3, "DOG", 2.5, 2] \)

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
<td>(integer)</td>
<td>5 + 2</td>
</tr>
<tr>
<td>2</td>
<td>7.0</td>
<td>(float)</td>
<td>5 + float(2)</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>(integer)</td>
<td>int(5.1 + 2.3)</td>
</tr>
<tr>
<td>4</td>
<td>&quot;5&quot;</td>
<td></td>
<td>str(5)</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td></td>
<td>len(str(5))</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td></td>
<td>len(S)</td>
</tr>
<tr>
<td>7</td>
<td>&quot;APPLEDOG&quot;</td>
<td></td>
<td>S + L[1]</td>
</tr>
<tr>
<td>8</td>
<td>&quot;DOGAPPLE&quot;</td>
<td></td>
<td>L[1] + S</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>(integer)</td>
<td>L[0] / 2</td>
</tr>
<tr>
<td>10</td>
<td>1.5</td>
<td>(float)</td>
<td>L[0] / 2.0</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td></td>
<td>L[-1]</td>
</tr>
<tr>
<td>12</td>
<td>&quot;E&quot;</td>
<td></td>
<td>S[len(L)]</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td></td>
<td>len(L[1])</td>
</tr>
<tr>
<td>14</td>
<td>[3]</td>
<td></td>
<td>[L[0]]</td>
</tr>
<tr>
<td>16</td>
<td>&quot;E&quot;</td>
<td></td>
<td>S[-1]</td>
</tr>
<tr>
<td>17</td>
<td>(1,2,3,4,5)</td>
<td>(tuple)</td>
<td>(1, 2, 3, 4, len(S))</td>
</tr>
<tr>
<td>18</td>
<td>[0,1,2,3,4]</td>
<td></td>
<td>range(5)</td>
</tr>
<tr>
<td>19</td>
<td>[3,4,5,6,7,8]</td>
<td></td>
<td>range(3,9)</td>
</tr>
<tr>
<td>20</td>
<td>[5,10,15,20,25,30]</td>
<td></td>
<td>range(5,31,5)</td>
</tr>
<tr>
<td>21</td>
<td>[]</td>
<td></td>
<td>range(10,1)</td>
</tr>
<tr>
<td>22</td>
<td>[10,9,8,7,6,5,4,3,2]</td>
<td></td>
<td>range(10,1,-1)</td>
</tr>
<tr>
<td>23</td>
<td>3</td>
<td></td>
<td>len(str(L[2]))</td>
</tr>
<tr>
<td>25</td>
<td>&quot;L&quot;</td>
<td></td>
<td>S[int(round(L[2]))]</td>
</tr>
<tr>
<td>26</td>
<td>[&quot;A&quot;,&quot;P&quot;,&quot;P&quot;,&quot;L&quot;,&quot;E&quot;]</td>
<td></td>
<td>[C for C in S]</td>
</tr>
<tr>
<td>27</td>
<td>[&quot;A&quot;,&quot;P&quot;,&quot;P&quot;,&quot;L&quot;,&quot;E&quot;]</td>
<td></td>
<td>[S[I] for I in range(len(S))]</td>
</tr>
<tr>
<td>28</td>
<td>[0,0,0,0]</td>
<td></td>
<td>[0 for I in range(4)]</td>
</tr>
<tr>
<td>29</td>
<td>[&quot;F&quot;,&quot;R&quot;,&quot;O&quot;,&quot;G&quot;]</td>
<td></td>
<td>[C for C in &quot;FROG&quot;]</td>
</tr>
<tr>
<td>30</td>
<td>[0,2,4]</td>
<td></td>
<td>[N*2 for N in range(3)]</td>
</tr>
</tbody>
</table>

Scoring: Ignore blank or incorrect answers. For answers that are correct, add 1 point to the total, but only add ½ point for answers that are missing quotes (for strings), brackets (for lists), parentheses (for tuples), or have the wrong type between integers and floats.
25 Points – What is printed out when `Main()` is called:

```python
def FN(A, B, C):
    print A + B * C
    return

def Main():
    A = 3
    B = 5
    FN(A, B, A)

def F2(G, H, I):
    FN(B, B, A)
    FN(H, I, G)
    print G + H
    return

def F2(G, H, I):
    print G + H
    return
```

From `FN(3, 5, 3)` → print `3 + 5 * 3` → **18**

From `FN(5, 5, 3)` → print `5 + 5 * 3` → **20**

From `FN(5, 5, 5)` → print `5 + 5 * 5` → **30**

From `F2(3, 5, 5)` →

`FN(5, 5, 3)` → print `5 + 5 * 3` → **20**

print `3 + 5` → **8**

Score as 5 points per answer.
<3> 25 Points – Write code inside the **TrashColor** function below to transform each pixel as follows: if the red value is greater than 160 then increase blue by 100, but if the red value is less than 80 then decrease green by 100, and if neither is true change the red to the original value of red plus the original blue value minus the original green value.

```python
def TrashColor(Canvas):
    for Y in range(getHeight(Canvas)):
        for X in range(getWidth(Canvas)):
            PX = getPixel(Canvas, X, Y)

            if getRed(PX) > 160:
                setBlue(PX, getBlue(PX) + 100)
            elif getRed(PX) < 80:
                setGreen(PX, getGreen(PX) - 100)
            else:
                setRed(PX, getRed(PX) + getBlue(PX) - getGreen(PX))

    repaint(Canvas)
    return
```

- or -

```python
R = getRed(PX)
G = getGreen(PX)
B = getBlue(PX)
if R > 160: setBlue(PX, B+100)
elif R < 80: setGreen(PX, G-100)
else: setRed(PX, R+B-G)
```

**Scoring:** Accept any solution that accomplishes the described tasks. Remove 1 point per syntax error (up to 10), 5 points per major conceptual error (up to 15).
9 Points – One of these code fragments is not like the others. 5 points: Identify (circle) the fragment with behavior different from the other seven. 4 Points: Explain how it is different.

\[
\begin{align*}
I &= 0 \\
&\text{while } (I \leq 9): \\
&\quad \text{print } I \\
&\quad I = I + 1 \\
I &= 0 \\
&\text{while } (I < 10): \\
&\quad \text{print } I \\
&\quad I = I + 1 \\
&\text{for } I \text{ in range}(0, 10, 1): \\
&\quad \text{print } I \\
&\text{for } I \text{ in range}(9, -1, -1): \\
&\quad \text{print } 9-I \\
&\text{for } I \text{ in range}(10): \\
&\quad \text{print } I \\
&\text{for } I \text{ in [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]}: \\
&\quad \text{print } I \\
I &= 0 \\
&\text{while } (I < 9): \\
&\quad \text{print } I \\
&\quad I = I + 1 \\
I &= 1 \\
&\text{while } (I \leq 10): \\
&\quad \text{print } I-1 \\
&\quad I = I + 1
\end{align*}
\]

Every code fragment prints out the numbers 0 through 9, except for the example in the lower left (5 points), which prints out the numbers 0 through 8 (4 points).
16 Points – A Squodge is a yellow square of radius 80 (the radius is from center-to-side, not center-to-corner), with a Gronk at each corner and at each side, as shown. A Gronk is a cyan circle of radius 50, with a red square of radius 30 on top, and a green circle of radius 20 on top of that. Fill in the blanks in the functions below to complete the drawing of a Squodge at location \(<X,Y>\) (which is the center of the Squodge, shown with a dot). The addCircle and addSquare functions are provided for you.

```python
def addCircle (Canvas, Xc, Yc, R, NewColor=black):
    addOvalFilled(Canvas,Xc-R,Yc-R,2*R+1,2*R+1, NewColor)
    addOval(Canvas,Xc-R,Yc-R,2*R+1,2*R+1,black)
    return

def addSquare (Canvas, Xc, Yc, R, NewColor=black):
    addRectFilled(Canvas,Xc-R,Yc-R,2*R+1,2*R+1,NewColor)
    addRect(Canvas,Xc-R,Yc-R,2*R+1,2*R+1,black)
    return

def Squodge (Canvas, X, Y):

def Gronk (X, Y):
    addCircle(Canvas, X, Y, 50, cyan)
    addSquare(Canvas, X, Y, 30, red)
    addCircle(Canvas, X, Y, 20, green)
    return

addSquare(Canvas, X, Y, 80, yellow)
Gronk(X+160, Y)
Gronk(X-160, Y)
Gronk(X, Y-160)
Gronk(X, Y+160)
Gronk(X+80, Y+80)
Gronk(X-80, Y+80)
Gronk(X+80, Y-80)
Gronk(X-80, Y-80)
return
```

There are 32 slots; score as \(\frac{1}{2}\) point per slot. The answers to the eight Gronk calls may be in any order.