20 Points – Modify the following code to use for-loops instead of while-loops. Use as few Python statements as possible without changing how the function works. (For any code that needs to be modified, circle it and write the replacement code next to it.)

```python
def Flood (Canvas, NewColor=white):
    W = getWidth(Canvas)
    H = getHeight(Canvas)
    Y = 0
    while (Y < H):
        X = 0
        while (X < W):
            setColor(getPixel(Canvas,X,Y),NewColor)
            X = X + 1
        Y = Y + 1
    repaint (Canvas)  # 4 pts for any other syntax errors
    return
```

To make each while-loop into a for-loop, the initialization of the variable is removed, the increment is removed, and the while is replaced by a for that (a) uses the same control variable, and (b) uses a range to get the correct span of values.

Some students replaced the entire function with the following code, pulled from a similar question on an exam from a previous semester:

```python
def Flood (Canvas, NewColor=white):
    for PX in getPixels(Canvas):
        setColor(PX,NewColor)
    repaint (Canvas)
    return
```

While this version has the same final result, there is a subtle difference between it and the expected answer. This version does not show the progression of the painting in a line-by-line manner, but repaints only at the end. If you received zero points for this answer, please come see me.
<2> 20 Points – What is printed out when \texttt{Main()} is called?

\begin{verbatim}
Temp = 7

def Glop(N):
    global Temp
    print Temp
    Temp = Temp + N
    return

def Splat(P):
    global Temp
    Glop (Temp+P)
    return

def Main():
    global Temp
    Glop (5)
    print Temp
    Splat (9)
    print Temp
    return
\end{verbatim}

\textbf{EXPLANATION:} \\
\textbf{ANSWERS:}

Temp is initialized to 7. \\
Main is called. \\
Glop is called where \( N = 5 \). \\
Glop prints Temp. \\
Temp goes to 7+5 = 12. \\
Glop returns to Main. \\
Main prints Temp. \\
Splat is called where \( P = 9 \). \\
Splat calls Glop where \( N = 12+9 = 21 \). \\
Glop prints Temp. \\
Temp goes to 12+21 = 33. \\
Glop returns to Splat. \\
Splat returns to Main. \\
Main prints Temp. \\
Main returns.

5 points per answer.
20 Points – Variable L contains the value [4, "Frog", ["Toad", 9, 5], 8]. Variable T contains the value (5, 3). Variable S contains the value "WOMBAT". Variable D contains the value {"Name":"Fred", "Grades":[98, 72, 83]}. What are the results of each of the following statements? If there is any kind of error in the statement, respond with “Error”.

<table>
<thead>
<tr>
<th>Statement</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X = 5/7</td>
</tr>
<tr>
<td>&quot;Frog&quot;</td>
<td>X = L[1]</td>
</tr>
<tr>
<td>9</td>
<td>X = L[2][1]</td>
</tr>
<tr>
<td>&quot;a&quot;</td>
<td>X = L[2][0][2]</td>
</tr>
<tr>
<td>&quot;Fred&quot;</td>
<td>X = D[&quot;Name&quot;]</td>
</tr>
<tr>
<td>error</td>
<td>X = D[6]</td>
</tr>
<tr>
<td>72</td>
<td>X = D[&quot;Grades&quot;][1]</td>
</tr>
<tr>
<td>4</td>
<td>X = len(L)</td>
</tr>
<tr>
<td>2</td>
<td>X = len(T)</td>
</tr>
<tr>
<td>2</td>
<td>X = len(D)</td>
</tr>
<tr>
<td>6</td>
<td>X = len(S)</td>
</tr>
<tr>
<td>[0,1,2,3,4,5]</td>
<td>X = range(6)</td>
</tr>
<tr>
<td>[2,3,4,5]</td>
<td>X = range(2,6)</td>
</tr>
<tr>
<td>[2,5,8,11]</td>
<td>X = range(2,12,3)</td>
</tr>
<tr>
<td>&quot;WOMBATS&quot;</td>
<td>X = &quot;S&quot; + &quot;S&quot;</td>
</tr>
<tr>
<td>&quot;SWOMBAT&quot;</td>
<td>X = &quot;S&quot; + S</td>
</tr>
<tr>
<td>&quot;SS&quot;</td>
<td>X = &quot;S&quot; + &quot;S&quot;</td>
</tr>
<tr>
<td>&quot;S + S&quot;</td>
<td>X = &quot;S + S&quot;</td>
</tr>
<tr>
<td>[0,1,4,9,16]</td>
<td>X = [I*I for I in range(5)]</td>
</tr>
<tr>
<td>[0,0,0,0,0]</td>
<td>X = [0 for I in range(5)]</td>
</tr>
</tbody>
</table>

1 point each answer, -½ for forgetting quotes (" ) or brackets ([ and ]).
20 Points – On the 3D diagram below, plot and label the positions of the listed points.

4 points each answer. -1 point for each axis in error. (Free point if answered at all.)

Answers must indicate which point goes with which question, but students may use either the index numbers (1-5) or the coordinates to do this. The answer above uses the index numbers. Vertical lines help show where the point projects onto the XZ plane, but this is not required.
15 Points – A **Blodgett** is a magenta square of radius 100, with a **Juntura** centered at each corner. A **Juntura** is a yellow square of radius 40, with a green square of radius 30 on top, and a **Siletz** centered on each side, as shown. A **Siletz** is a cyan circle of radius 20, with a red circle of radius 10 on top.

Fill in the blanks in the functions below to complete the drawing of a **Blodgett** at location <X,Y> (which is the center of the **Blodgett**, shown with a dot).

```python
def addCi rcle (Canvas, X, Y, R, Fill):
    addOvalFilled(Canvas, X-R, Y-R, 2*R+1, 2*R+1, Fill)
    addOval(Canvas, X-R, Y-R, 2*R+1, 2*R+1, black)
    return

def addSquare (Canvas, X, Y, R, Fill):
    addRectFilled(Canvas, X-R, Y-R, 2*R+1, 2*R+1, Fill)
    addRect(Canvas, X-R, Y-R, 2*R+1, 2*R+1, black)
    return

def addBlodgett (Canvas,X,Y):
    def addSiletz (X,Y):
        addCircle(Canvas, X, Y, 20, cyan)
        addCircle(Canvas, X, Y, 10, red)
        return
    
    def addJuntura (X,Y):
        addSquare(Canvas, X, Y, 40, yellow)
        addSquare(Canvas, X, Y, 30, green)
        addSiletz(X, Y+40)
        addSiletz(X, Y-40)
        addSiletz(X-40, Y)
        addSiletz(X+40, Y)
        return
    
    addSquare(Canvas, X, Y, 100, magenta)
    addJuntura(X+100, Y+100)
    addJuntura(X-100, Y+100)
    addJuntura(X+100, Y-100)
    addJuntura(X-100, Y-100)
    return
```
There are 36 blanks to fill in. Count up the number of correct answers, divide by 3, and take the ceiling of the result. That counts for up to 12 points. Assess -1 point if arguments to addCircle in addSiletz or addSquare in addJuntura are reversed, and -1 for any syntax errors, up to a maximum of three.

<6> 5 Points – In problem 5, can any function outside addBlodgett see the functions addJuntura or addSiletz? Explain. Use the back of this page as needed.

(3 points) NO.

(2 points) addSiletz and addJuntura are local to addBlodgett (indented inside), and are only visible to the code inside the addBlodgett function. There is no need for them to be used elsewhere, so they are hidden from all other code by making them local.

NOTE: I didn’t know what to call the final drawn figure, but a “Blodgett” seemed to be appropriate somehow. Blodgett is also the name of a small town on U.S. 20 in western Oregon, so I also gave the other graphics objects similar names that don’t obviously sound like town names: Siletz and Juntura are also small towns in Oregon. I’ve been through all three.