<table>
<thead>
<tr>
<th>QUESTION</th>
<th>POINTS</th>
<th>SCORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
15 Points – Do any 15; do more for extra credit. In this problem assume that the statements 
\[S = "\text{HOUSE}","L = [3,"\text{CAT"},2], \text{and } T = (3,"\text{FROG"},2)\]
have already been executed. Show the values of \(N\) after each statement is executed (all are
independent of one another). Indicate any cases where the computation cannot take
place because of some form of error. Be very careful about the type of the result, particularly
integers and floats, and make certain to put quotes around string results.

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. (N = 3 + 4 \times 2)</td>
<td>11</td>
</tr>
<tr>
<td>2. (N = (3 + 4) \times 2)</td>
<td>14</td>
</tr>
<tr>
<td>3. (N = 5 \div 3)</td>
<td>1</td>
</tr>
<tr>
<td>4. (N = 5.0 \div 3)</td>
<td>1.666666666...</td>
</tr>
<tr>
<td>5. (N = \text{sqrt}(4))</td>
<td>2.0 (\text{sqrt} always returns a float)</td>
</tr>
<tr>
<td>6. (N = \text{len}(S))</td>
<td>5</td>
</tr>
<tr>
<td>7. (N = T[1])</td>
<td>&quot;FROG&quot;</td>
</tr>
<tr>
<td>8. (N = L[1] + S)</td>
<td>&quot;CATHOUSE&quot;</td>
</tr>
<tr>
<td>9. (N = S + L[1])</td>
<td>&quot;HOUSECAT&quot;</td>
</tr>
<tr>
<td>10. (N = L + L)</td>
<td>[3,&quot;CAT&quot;,2,3,&quot;CAT&quot;,2]</td>
</tr>
<tr>
<td>11. (N = T + T)</td>
<td>(3,&quot;FROG&quot;,2,3,&quot;FROG&quot;,2)</td>
</tr>
<tr>
<td>12. (N = L + T)</td>
<td>Error (can’t add list and tuple)</td>
</tr>
<tr>
<td>13. (N = (L[0]+2, T[-1], S[2]))</td>
<td>(5,2,&quot;U&quot;)</td>
</tr>
<tr>
<td>14. (N = (T[0] &gt; 1))</td>
<td>True or 1</td>
</tr>
<tr>
<td>15. (N = S[\text{len}(S)-1])</td>
<td>&quot;E&quot;</td>
</tr>
<tr>
<td>16. (N = S[\text{len}(S)])</td>
<td>Error (index beyond end of string)</td>
</tr>
<tr>
<td>17. (N = &quot;C:&quot; + S + &quot;.gif&quot;)</td>
<td>&quot;C:\HOUSE.gif&quot;</td>
</tr>
<tr>
<td>18. (N = S[0] + L[1][1] + &quot;T&quot;)</td>
<td>&quot;HAT&quot;</td>
</tr>
<tr>
<td>19. (N = \text{range}(3,14,2))</td>
<td>[3,5,7,9,11,13]</td>
</tr>
<tr>
<td>20. (N = (3+4j) + (5-2j))</td>
<td>(8+2j)</td>
</tr>
</tbody>
</table>

Score as 1 point each. Add +1 for each correct answer; do not take off points for wrong or blank
answers. Total is sum of +1 answers, and may be as big as 20. Remove \(\frac{1}{2}\) point for such things as
forgetting the .0 on floats of integers, forgetting quotes around strings, using [] for tuples or
() for lists, etc. It is OK to use either single or double quotes, as in 'FROG' or "FROG".
20 Points – What is printed out by each of the following statements?

A. \texttt{for x in range(3): printNow(x)}

\begin{verbatim}
0
1
2
\end{verbatim}

B. \texttt{for x in range(3,12): printNow(x)}

\begin{verbatim}
3
4
5
6
7
8
9
10
11
\end{verbatim}

C. \texttt{for x in range(3,12,3): printNow(x)}

\begin{verbatim}
3
6
9
\end{verbatim}

D. \texttt{for x in range(len("Frog")): printNow(x)}

\begin{verbatim}
0
1
2
3
\end{verbatim}

E. \texttt{for x in [4,"Dog",6+3]: printNow(x)}

\begin{verbatim}
4
"Dog"
9
\end{verbatim}

Score as 4 points each. Remove 1 point per error, up to 4 per question. It is OK if all items are written on the same line, instead of vertically as shown.
20 Points – Completely rewrite function \texttt{Main()} to perform the same task using a \texttt{for}-loop instead of a \texttt{while}-loop:

\begin{verbatim}
def Main():
    I = 7
    while (I < 29):
        printNow(I*I)
        I = I + 3
    return

def Main():
    for I in range(7,29,3):
        printNow(I*I)
    return
\end{verbatim}

Remove 1 point per error, including errors in syntax (forgetting the colon, capitalizing \texttt{range}, etc), using the wrong variables, forgetting the \texttt{I*I}, including an explicit \texttt{I = 7} or \texttt{I = I + 3} statement, etc. It is OK to have \texttt{printNow} on the same line as \texttt{for}.

10 Points – Show what is printed out as the result from calling \texttt{F2(5,9,4)}

\begin{verbatim}
def F1(L,M,N):
    printNow(M)
    printNow(L)
    printNow(N)
    return

def F2(L,M,N):
    F1(N,M,L)
    return
\end{verbatim}

\textbf{Answer:} 9 4 5

\textbf{Explanation:}
The call to \texttt{F2} is in the order \texttt{F1(5,9,4)}.
In \texttt{F2}, \texttt{L=5}, \texttt{M=9}, \texttt{N=4}.
The call to \texttt{F1} is in the order \texttt{F1(4,9,5)}.
In \texttt{F1}, \texttt{L=4}, \texttt{M=9}, \texttt{N=5}.
\texttt{F1} prints \texttt{M} then \texttt{L} then \texttt{N}, or \texttt{9 then 4 then 5}.

The most common expected error involves using a wrong parameter order, thus printing the numbers in the wrong order. Remove 5 points for \texttt{9-5-4}, remove all 10 points for any other order.
10 Points – Find and correct all the syntax errors in the following code (don’t worry about what the code is supposed to do; it doesn’t do anything useful). There are at least 10 errors; ½-point extra credit per extra error for finding more than 10.

# A Very Strange Function

def MyFunction(Temp):
    Glop = Temp + 1
    While (Temp <= 42): Result = "Yes"
    Glop == 2*(Glop)
    Temp += 7
    if (Glop == 5):
        Result = "Yes"
    elseif (Glop < Temp):
        Result = "Toad"
    else:
        Pass
    Frog = Result
    If (Temp == 10):
        Temp = Temp - 3
    >>>print (Result, Frog)

I count at least 15 errors. Add +1 for each correctly identified error, up to 10, +½ for each correctly identified error more than 10. Remove ½ point for each correct item identified as incorrect (for example, Temp += 7 is OK). By my estimation, this problem can have a score at least as high as 12½ points.
def Process (Canvas):
    for px in getPixels(Canvas):
        R = getRed(px)
        G = getGreen(px)
        B = getBlue(px)

        #--------------------- Solution 1
        SaveB = B
        if (R > G):
            B = R
        else:
            B = G
        R = SaveB
        G = SaveB

        #--------------------- Solution 2
        if (R > G):
            B = R
        else:
            B = G
        R = getBlue(px)
        G = getBlue(px)

        #--------------------- Solution 3
        if (R > G): B = R
        else: B = G
        R = getBlue(px)
        G = R

        #--------------------- Solution 4
        B = max(R,G)
        R = getBlue(px)
        G = R

        setColor(px, makeColor(R,G,B))
    return

Score as 10 points for the if or max and for setting B correctly, 5 points for recognizing that the old value of B needs to be preserved. Use 5 points for syntax. Remove 5 points for including other code, such as using setRed, setGreen, or setBlue. Accept any approach above, or any other correct variation.