<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 assignment statement \( T = "POLECAT" \) has 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.

A. \( N = 15 / 2 \)  
B. \( N = 15 / 2.0 \)  
C. \( N = \text{len}(T) \)  
D. \( N = \text{len}([1, 2, "DOG", [6+7, "CAT"], 9.0]) \)  
E. \( N = \text{len}(T + T) \)  
F. \( N = \text{len}(T) / 3 \)  
G. \( N = \text{len}(T) / 3.0 \)  
H. \( N = T[\text{len}(T)/3] \)  
I. \( N = [T[2], 7, "FROG", T[\text{len}(T)-3]] \)  
J. \( N = "C:\" + T + ".JPG" \)  
K. \( N = T + "5" \)  
L. \( N = T + 5 \)  
M. \( N = T[-2] \)  
N. \( N = T[12] \)  
P. \( N = \text{range(len}(T)) \)  
Q. \( N = \text{range}(3,12) \)  
R. \( N = \text{range}(3,12,4) \)

Score as 1 point each. Add +1 for each correct answer, do not take off points for wrong answers. Total is sum of +1 answers, and may be as big as 18.
<2> 20 Points – What is printed out by each of the following statements?

A. `for x in range(5): printNow(x)`

0
1
2
3
4

B. `for x in range(5,10): printNow(x)`

5
6
7
8
9

C. `for x in range(5,10,3): printNow(x)`

5
8

D. `for x in range(len("Hello")): printNow(x)`

0
1
2
3
4

E. `for x in [5,3,7,2]: printNow(x)`

5
3
7
2

Score as 4 points each. Remove 1 point per error, up to 4 per question.
20 Points – Completely rewrite function `Main()` to perform the same task using a `while`-loop instead of a `for`-loop:

```python
def Main(J,K,L):
    for I in range(K,L,J):
        printNow(I*I)
    return
```

```python
def Main(J,K,L):
    I = K
    while (I < L):
        printNow(I*I)
        I = I + J
    return
```

Remove 1 point per error, including syntax, using the wrong variables, etc.

10 Points – Using the function from problem #3, show what is printed out as the result from the call `Main(5,3,10)`

**Answer:**

```
9
64
```

**Explanation:**

Actual parameter 5 goes in for formal parameter J,
Actual parameter 3 goes in for formal parameter K,
Actual parameter 10 goes in for formal parameter L.
Thus, the for-loop runs as follows:

```python
for I in range(3,10,5):
    printNow(I*I)
```

The range generates the list [3, 8], so the squares of those items are 9 and 64.

Common expected errors include wrong order for parameters, such as considering the range to be `range(5,3,10)` instead of `range(3,10,5)`, and omitting the squaring operation. Remove 5 points for each of these infractions.
<5> 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; extra credit for finding more than 10.

# A Very Strange Function

```python
define Weird Function (Temp):
    Glop = Temp + 1
    if If (Temp <= 42): Temp = Glop + 1  # (line needs splitting)
        Glop = Glop / 2
        Result = "Yes"
    elseif (Glop == 5):
        Result = "No"  # This is one result
    else:
        Result = "Toad"  # This is another result
    Frog = Result
    whileWhile (Temp > 10):
        Temp = Temp - 3
    >>>>>return [Result, Frog]  # (wrong indentation)
```

I count at least 14 errors. Add +1 for each correctly identified error, up to 14. Remove ½ point for each correct item identified as incorrect.

<6> 25 Points – Complete the following Python code to change the color of each pixel by setting the new green value to the smaller of the red and blue values without using the min function, and setting both the new red and new blue values to the old green value.

```python
def Process (Canvas):
    for px in getPixels(Canvas):
        R = getRed(px)
        G = getGreen(px)
        B = getBlue(px)
        SaveG = G
        if (R < B):
            if (R < B):
                G = R
            else:
                G = B
        else:
            G = B
        R = SaveG
        R = getGreen(px)
        B = SaveG
        B = getGreen(px)
        setColor(px, makeColor(R,G,B))
    return
```

Score as 10 points for the if statement, 5 points for recognizing that the old value of G needs to be preserved. Use the other 5 points for syntax. Remove 5 points for using setRed, setGreen, or setBlue. Accept either approach above, and accept other correct answers, if any.