Question 1.  [8 marks]
The following code runs without errors:

```python
import media

def func1(s):
    s = s + "a lot"
    return s

def func2(n):
    n = n * 3

def func3(p):
    p = media.create_picture(2, 2, media.black)
    return p

def func4(p):
    for pixel in p:
        media.set_green(pixel, 98)

if __name__ == "__main__":
    s = "I like roller coasters"
    s = func1(s)
    print s

    amount = 5
    func2(amount)
    print amount

    picture1 = media.load_picture("giraffe.jpg") # A big picture
    func3(picture1)
    if media.get_height(picture1) == 2:
        print "really tiny"
    else:
        print "still big"

    picture2 = media.create_picture(50, 50, media.black)
    func4(picture2)
    one_pixel = media.get_pixel(picture2, 0, 0)
    if media.get_green(one_pixel) > 0:
        print "green changed"
    else:
        print "green still zero"
```

Assume that the picture "giraffe.jpg" has height and width much larger than 2. On the following page, show the four lines of output that this code produces. **Strong hint:** Use the blank space provided to trace the code using the memory model.
Solution:

I like roller coasters a lot
5
still big
green changed
Question 2. [6 marks]

Each of these subquestions contains a block of code. Treat each block of code independently (code in one part is not related to code in another), and fill in the blanks for each question.

Part (a) [1 mark] Assignment

```python
normal = 98.6
feverish = normal + 4
normal = normal + 1
print feverish
```

The output from this block of code is ______________.

Solution:

102.6

Part (b) [3 marks] Conditionals and Booleans

The table to the right shows the decision process of a police offer giving a traffic ticket. Assume you have a int variable speed_above and a boolean variable priors, corresponding to the first two columns in the table. Fill in the boolean conditions below to calculate the police officer’s response.

<table>
<thead>
<tr>
<th>Speed Above Limit</th>
<th>Prior Ticket?</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>50 or more</td>
<td>Does not matter</td>
<td>“Step out of the car.”</td>
</tr>
<tr>
<td>At least 20 but less than 50</td>
<td>Does not matter</td>
<td>“Here’s a big ticket.”</td>
</tr>
<tr>
<td>Less than 20</td>
<td>Yes</td>
<td>“Here’s a small ticket.”</td>
</tr>
<tr>
<td>Less than 20</td>
<td>No</td>
<td>“Be more careful”</td>
</tr>
</tbody>
</table>

```python
if .................................................................:
    print "Step out of the car."

elif .................................................................:
    print "Here’s a big ticket."

elif .................................................................:
    print "Here’s a small ticket."

else:
    print "Be more careful."
```

Solution:

```python
if speed_above >= 50:
    print "Step out of the car."

elif speed_above > 20:
    print "Here’s a big ticket."
```
elif priors:
    print "Here’s a small ticket."

else:
    print "Be more careful."

Part (c) [1 MARK] **Data Types**

Fill in the blank so that when this code is run, the user is asked to enter two numbers and then the difference between those numbers is printed. The values entered by the user will be whole numbers, and the first one will be at least as big as the second.

```python
num1 = raw_input("Please give me the older person’s age: ")
num2 = raw_input("Please give me the younger person’s age: ")

print "The difference between your ages is", ________________________________
```

Solution:

```python
print "The difference between your ages is", int(num1) - int(num2)
```

Part (d) [1 MARK] **Calling Functions**

Suppose this function has been defined:

```python
def calculate_wage(years_exp):
    
    
    
    return wage
```

Fill in the blank to call `calculate_wage` to obtain the hourly wage of an employee with 2 years of prior experience.

```python
print "The wage for 2 years of experience is", ________________________________
```

Solution:

```python
print "The wage for 2 years of experience is", calculate_wage(2)
```
Question 3.  [8 marks]

Write the function below, according to its docstring.

```python
def red_agreement(p1, p2):
    '''p1 and p2 are pictures with the same height and width. Return the number of
    pixels in p1 that have exactly the same red value as the pixel at the same location
    in p2.'''
    count = 0
    for pixel in p1:
        x = media.get_x(pixel)
y        y = media.get_y(pixel)
sister = media.get_pixel(p2, x, y)
        if media.get_red(pixel) == media.get_red(sister):
            count += 1
    return count
```

Solution:

```python
count = 0
for pixel in p1:
    x = media.get_x(pixel)
y    y = media.get_y(pixel)
sister = media.get_pixel(p2, x, y)
    if media.get_red(pixel) == media.get_red(sister):
        count += 1
return count
```