Do not turn this page until you have received the signal to start.
(Please fill out the identification section above, write your name on the back of the test, and read the instructions below.)

Good Luck!

This midterm consists of 3 questions on 8 pages (including this one). When you receive the signal to start, please make sure that your copy is complete.

Comments and docstrings are not required except where indicated, although they may help us mark your answers. They may also get you part marks if you can’t figure out how to write the code. No error checking is required: assume all user input and all argument values are valid.

If you use any space for rough work, indicate clearly what you want marked. ~TOTAL: _____/22

2a: 2b: 2c: 2d:
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.
Output produced:

__________________________________________________________________

__________________________________________________________________

__________________________________________________________________

__________________________________________________________________
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 ____________.

Part (b) [3 marks] Conditionals and Booleans

The table to the right shows the decision process of a police officer giving a traffic ticket. Assume you have an 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."
```

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", ______________________________________
```
Part (d) [1 mark] Calling Functions
Suppose this function has been defined:

```python
def calculate_wage(years_exp):
    '''Return the hourly wage of an employee with years_exp years of prior experience.'''
    # The code for this function is not shown.
    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", ________________________________
```
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.'''
```

__builtins__:
abs(number) -> number
    Return the absolute value of the given number.
max(a, b, c, ...) -> value
    With two or more arguments, return the largest argument.
min(a, b, c, ...) -> value
    With two or more arguments, return the smallest argument.
raw_input([prompt]) -> string
    Read a string from standard input. The trailing newline is stripped. The prompt string,
    if given, is printed without a trailing newline before reading.
float:
    float(x) -> float
        Convert a string or number to a float, if possible.
int:
    int(x) -> integer
        Convert a string or number to an integer, if possible. A floating point argument
        will be truncated towards zero.
media:
    choose_file() -> str
        Prompt user to pick a file. Return the path to that file.
create_picture(int, int) -> Picture
    Given a width and a height, return a Picture with that width and height. All pixels are white.
get_blue(Pixel) -> int
    Return the blue value of the given Pixel.
get_color(Pixel) -> Color
    Return the Color object with the given Pixel’s RGB values.
get_green(Pixel) -> int
    Return the green value of the given Pixel.
get_height(pic)
    Return how many pixels high Picture pic is.
get_pixel(Picture, int, int) -> Pixel
    Given x and y coordinates, return the Pixel at (x, y) in the given Picture.
get_red(Pixel) -> int
    Return the red value of the given Pixel.
get_width(pic)
    Return how many pixels wide Picture pic is.
get_x(pix)
    Return the x coordinate of Pixel pix.
get_y(pix)
    Return the y coordinate of Pixel pix.
load_picture(str) -> Picture
    Return a Picture object from file with the given filename.
set_blue(Pixel, int)
    Set the blue value of the given Pixel to the given int value.
set_color(Pixel, Color)
    Set the RGB values of the given Pixel to those of the given Color.
set_green(Pixel, int)
    Set the green value of the given Pixel to the given int value.
set_red(Pixel, int)
    Set the red value of the given Pixel to the given int value.