

Question 1. [8 MARKS]

The following code runs without errors:

```
import media

def func1(p):
    for pixel in p:
        media.set_green(pixel, 20)

def func2(p):
    p = media.create_picture(10, 15, media.black)
    for pixel in p:
        media.set_color(pixel, media.orange)

def func3(s):
    s = s + "!"

def func4(n):
    n = n - 1
    return n

if __name__ == "__main__":
    picture1 = media.create_picture(50, 50, media.black)
    func1(picture1)
    one_pixel = media.get_pixel(picture1, 0, 0)
    if media.get_color(one_pixel) == media.black:
        print "still black"
    else:
        print "changed"

    picture2 = media.load_picture("face.jpg") # A picture with no orange
    func2(picture2)
    one_pixel = media.get_pixel(picture2, 0, 0)
    if media.get_color(one_pixel) == media.orange:
        print "changed to orange"
    else:
        print "not orange"

    message = "wow"
    func3(message)
    print message

    how_much = 10
    func4(how_much)
    print how_much
```

Assume that the picture “face.jpg” has no orange pixels in it. 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:

changed
not orange
wow
10

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**

```
var_A = 11
var_B = var_A
var_A = 42
print var_B
```

The output from this block of code is _____.

Solution:

11

Part (b) [3 MARKS] **Conditionals and Booleans**

The table to the right shows how an employee's age and experience affects his or her hourly wage. Assume that you have a `boolean` variable `experienced` and an `int` variable `age`. Fill in the boolean conditions in the code below to calculate the hourly wage for the employee.

Age	Experienced?	
	Yes	No
under 18	\$12.00	\$9.50
18 and over	\$15.00	\$10.50

```
if _____:
    if _____:
        wage = 12
    else:
        wage = 15
else:
    if _____:
        wage = 9.5
    else:
        wage = 10.5
```

Solution:

```
if experienced:
    if age < 18:
        wage = 12
    else:
        wage = 15
else:
    if age < 18:
        wage = 9.5
    else:
        wage = 10.5
```

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 average of those numbers is printed. The user input may contain decimal values (e.g., 9.75 or 4.2).

```
num1 = raw_input("How many pounds does your backpack weigh? ")
num2 = raw_input("How many pounds does your friend's backpack weigh? ")

print "Your average backpack weight is" , _____
```

Solution:

```
print "Your average backpack weight is" , (float(num1) + float(num2)) / 2
```

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

Suppose this function has been defined:

```
def area(r):
    '''r is a float. Return the area of a circle whose radius is r.'''
    # The code for this function is not shown.
    return answer
```

Fill in the blank to call `area` to obtain the area of a circle with radius 19.6.

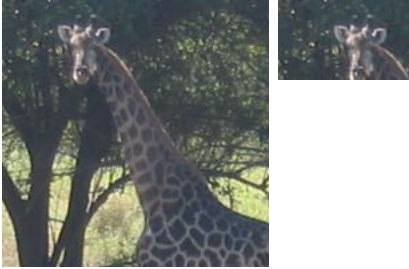
```
print "The area of a circle with radius 19.6 is" , _____
```

Solution:

```
print "The area of a circle with radius 19.6 is" , area(19.6)
```

Question 3. [8 MARKS]

Write the function below, according to its docstring. As an example, it could be used to create the picture on the right from the picture on the left:



You may not use the function `media.crop_picture`.

Strong hint: Iterate through the smaller picture — it is much easier that way.

```
def crop(p, w, h):
    '''p is a Picture; w and h are ints >= 1. Return a new picture of width w and
    height h containing the upperleft portion of the image in p. Assume
    that p has width at least w and height at least h.'''
```

Solution:

```
new_pic = media.create_picture(w, h, media.black)
for pixel in new_pic:
    x = media.get_x(pixel)
    y = media.get_y(pixel)
    c = media.get_color(media.get_pixel(p, x, y))
    media.set_color(pixel, c)
return new_pic
```