

Question 1. [4 MARKS]

Suppose these functions have been defined:

```
import media

def do_something(left, right):
    left = right
    for pixel in left:
        media.set_red(pixel, 0)

def do_stuff(a, b, c):
    a = b + c
    c = a + b
    return a
```

The following code runs without errors. Fill in the boxes below to show the output printed or answer the question, as indicated.

```
a = 145
one = 20
two = 25
one = do_stuff(a, one, two)
print a
```

output:

```
print one
```

output:

```
print two
```

output:

```
pic1 = media.load_picture(media.choose_file())
pic2 = media.load_picture(media.choose_file())
do_something(pic1, pic2)
media.show(pic1)
```

Does the picture that was just displayed have any red in it? Circle one: yes no

```
media.show(pic2)
```

Does the picture that was just displayed have any red in it? Circle one: yes no

Solution:

```
a = 145
one = 20
two = 25
one = do_stuff(a, one, two)
print a
```

output: 145

```
print one
```

output: 45

```
print two
```

output: 25

```
pic1 = media.load_picture(media.choose_file())
pic2 = media.load_picture(media.choose_file())
do_something(pic1, pic2)
media.show(pic1)
```

Does the picture that was just displayed have any red in it? Circle one: yes no

```
media.show(pic2)
```

Does the picture that was just displayed have any red in it? Circle one: yes no

Strictly speaking, the answer for pic1 should be “yes, unless the picture had no red in it to begin with.”

Question 2. [6 MARKS]

Write a function called `rotate_rgb` that takes a `Picture` as a parameter and, for every pixel, sets the red value to the original green value, the green value to the original blue value, and the blue value to the original red value.

You will need to use one variable for the parameter and one variable for the pixel values in the for loop. You can earn the full 6 marks if you use only one more variable. If you use more than one, you can earn at most 5 marks.

```
def rotate_RGB(pic):
    for p in pic:
        red = media.get_red(p)
        media.set_red(p, media.get_green(p))
        media.set_green(p, media.get_blue(p))
        media.set_blue(p, red)
```

Question 3. [8 MARKS]

Write a program that uses `raw_input` to prompt the user for his or her income and then prints the amount of health premium owed on that income. (A health premium is a kind of tax that is used to pay for health coverage.) This program consists of two parts: a function named `health_premium` on this page and a main block on the next page.

The health premium for a given income is computed as follows:

1. For less than \$25,000 income, no premium is owed.
2. For incomes of \$25,000 to less than \$50,000, the premium is 5% of `income`.
3. For incomes of \$50,000 or more, the premium on the first \$100,000 is 7%, and the premium on any income above that is 10%. The total premium owed is the sum of those two values.

Your program should ask the user for input using the string “Please enter your income: ” and print the health premium owed as the string “Your health premium is \$XXX.YY.” The amount printed should be rounded to the nearest cent. XXX can be as many digits as it needs to be, YY must be 2 digits long, and the total amount must be preceded by the character ‘\$’.

```
def health_premium(income):
    '''Return the health premium owed for a given income amount. The parameter
    income is an int, and the health premium returned is a float.
    '''

    if income < 25000:
        return 0.0
    elif income < 50000:
        return income * .05
    else:
        premium = min(income, 100000) * .07
        income -= 100000
        if income > 0:
            premium += income * .1
        return premium
```

or

```
if income < 25000:
    return 0.0
elif income < 50000:
    return income * .05
elif income < 100000:
    return income * .07
else:
    income -= 100000
    return 7000 + income * .1
```

```
if __name__ == "__main__":  
    income = int(raw_input("Please enter your income: "))  
    print "Your health premium is $.2f" % health_premium(income)
```