

Question 1. [9 MARKS]**Part (a)** [5 MARKS]

Write the function `invert_colour` which has a single `Picture` argument `pic` and returns a `Picture` of the same dimensions as `pic` but with inverted colours. To invert the colours, set each of the red, blue and green components of every pixel according to this table.

Colour	
Original	New
0	255
1	254
2	253
...	...
255	0

The original picture should be unchanged.

```
def invert_colour(pic):
    '''Return a copy of Picture pic where all colour values have been inverted.
    '''
    new_width = pic.get_width()
    new_height = pic.get_height()
    newpic = media.create_picture(new_width, new_height)

    for pixel in pic:
        x = media.get_x(pixel);
        y = media.get_y(pixel);
        newpixel = media.get_pixel(newpic, x, y)
        new_red = 255 - media.get_red(pixel)
        new_blue = 255 - media.get_blue(pixel)
        new_green = 255 - media.get_green(pixel)
        media.set_red(newpixel, new_red)
        media.set_blue(newpixel, new_blue)
        media.set_green(newpixel, new_green)

    return newpic
```

Part (b) [4 MARKS]

Notice that taking the colour-inversion of an already-inverted picture will produce a picture that should be identical to the original you started with. Suppose that you have a function `equals` that takes two `Picture` arguments and returns `True` iff the pictures are the same. Complete the main block below so that it takes a picture, inverts it twice and compares it to the original. If the pictures are the same, your code should print “okay”, but if not, it should print “problem”.

```
if __name__ == '__main__':
    original = media.load_picture(media.choose_file())

    result = invert_colour(invert_colour(original))
    if equals(result, original):
        print 'okay'
    else:
        print 'problem'
```

Question 2. [4 MARKS]

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

Part (a) [1 MARK] **Order of Execution**

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

After this code is executed, the value of `var_B` is 11 .

Part (b) [1 MARK] **Conditionals and Booleans**

The table to the right shows how the size of an egg is determined by its weight. Assume that you have a variable `weight` with a floating point value. Fill in the boolean conditions in the code below so that the correct size is printed based on the weight. If an egg's weight is on the boundary between two size ranges, it is assigned the smaller size.

Size	Weight
Small	50 grams or less
Medium	50-57 grams
Large	57-64 grams
Jumbo	more than 64 grams

```

if weight < = 50:
    print "Small"

elif weight <= 57:
    print "Medium"

elif weight <= 64:
    print "Large"

else:
    print "Jumbo"

```

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 sum of those numbers is printed. Remember that the value returned by function `raw_input` is type `str`.

```

num1 = raw_input("Please give me a number: ")
num2 = raw_input("Please give me another number: ")

print "The sum of those numbers is" , float(num2) + float(num1)
# note: we also accepted int(num1) + int(num2)

```

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

Fill in the blank to call `city_from_postalcode` to obtain the city that has the postal code POA 1A0.

```

def city_from_postalcode(pcode):
    '''Return the name of the city that contains postal code pcode.'''
    # The code for this function is not shown.
    return city

print "The city for POA 1A0 is" , city_from_postalcode("POA 1A0")

```

Question 3. [8 MARKS]**Part (a)** [4 MARKS]

A company pays their employees differently depending on employee age and experience level. Write a function `weekly_pay` that takes three parameters: an integer age, a number of hours worked (may contain a decimal) and a boolean indicating whether the employee has experience. The function should return the weekly pay as a float calculated using the proper rate from this table.

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

```
def weekly_pay(age, hours, experienced) {
    if experienced:
        if age < 18:
            rate = 12
        else:
            rate = 15
    else:
        if age < 18:
            rate = 9.5
        else:
            rate = 10
    return rate * hours
```

Part (b) [4 MARKS]

Complete the `main` block below. Nick is an experienced employee. Your program should use `raw_input` to ask the user to input the number of hours that Nick has worked this week and his age. Then it should use `weekly_pay` to determine Nick's pay for this week and finally print it out.

```
if __name__ == "__main__":

    age = int(raw_input("Please enter Nick's age. "))
    hours = float(raw_input("How many hours has nick worked? "))
    print "Nick should be paid", weekly_pay(age, hours, True)
```