

Question 1. [5 MARKS]

Complete the following function according to its docstring description.

```
def modify_colors(pic):  
    '''Modify the Picture pic so that any white pixels are replaced with  
    blue and any black pixels are replaced with white.  Pixels of any other  
    colour remain unchanged.'''  
  
    for pix in pic:  
        if media.get_color(pix) == media.white:  
            media.set_color(pix, media.blue)  
        elif media.get_color(pix) == media.black:  
            media.set_color(pix, media.white)
```

-2 No if statements, or bad logic

-1 doesn't use ==

-1 no for loop

-1 no media.

-1 uses return stmt

-1 each pair - or fewer - of miscellaneous stupidities: indentation, use of RGB tuple instead of color obj, using set_black, etc.

Question 2. [7 MARKS]

In the code below, a picture is created and the function `mysterious` is called:

```
if __name__ == '__main__':
    pic = media.create_picture(2, 2)
    pix = media.get_pixel(pic, 0, 0)
    media.set_color(pix, media.hotpink)
    pix = media.get_pixel(pic, 1, 0)
    media.set_color(pix, media.firebrick)
    mysterious(pic)
```

Based on the call to `mysterious` using the picture above, trace the variable values during execution of the function `mysterious`. For each blank in the table below, fill in the value of the variable specified after the line has executed or write “not reached” if that line was not executed.

Note: the RGB values of `hotpink` and `firebrick` are given on the second last page.

```
def mysterious(pic):
    x = 0

    y = 0

    for pix in pic:

        b = media.get_blue(pix)

        z = b < 150

        if z:

            x = x + 1

        else:

            y = y + 1
```

Show variable values after each line has executed:				
x:	0			
y:	0			
	During iteration:			
	1	2	3	4
b:	180	255	34	255
z:	False	False	True	False
z:	False	False	True	False
x:	NR	NR	1	NR
y:	1	2	NR	3

Note that the order of pixels is (0,0), (0,1), (1,0), (1,1). If students traverse in column order, the first row will be: 180, 34, 255, 255. See marking scheme below:

- 1 if the results are (sort of) correct, but the traversal is in column order
- 1 for every 1-2 rows/columns of the result table that's
- 1 if 0 is used instead of 255: means the picture is assumed to be black, instead of white
- 2 if not reached is wrong - not present or in wrong place

Question 3. [8 MARKS]

Write a program that prompts for a picture file using `choose_file`. It then uses `raw_input` to prompt for a colour component (either “red”, “green”, or “blue”). This program consists of two parts: a function, `diff_colour`, on this page, and a main block on the next page.

For the colour component selected, for each pixel in the picture, your program should set that colour component to the absolute value of the difference of the other two colour components of that pixel. At the end of the program, show the resulting picture.

You may assume that the colour component entered is either “red”, “green” or “blue”.

Your program should define and use the function `diff_colour` according to its docstring description below.

```
import media

def diff_colour(pic, colour):
    '''For each pixel in Picture pic, set the colour component specified in
    string colour ("red", "green" or "blue") to the absolute value of the
    difference of the other two colour components of that pixel.'''

    for pix in pic:

        red = media.get_red(pix)
        green = media.get_green(pix)
        blue = media.get_blue(pix)

        if colour == "red":
            media.set_red(pix, abs(green - blue))
        elif colour == "green":
            media.set_green(pix, abs(red - blue))
        else:
            media.set_blue(pix, abs(red - green))

if __name__ == '__main__':
    file_name = media.choose_file()
    pic = media.load_picture(file_name)
    colour = raw_input("Please enter a colour: ")
    diff_colour(pic, colour)
    media.show(pic)
```

Function definition is out of 5; main part is out of 3. If

- 2 no for-loop
- 1 no media.
- 1 various logic errors in fuction.
- 1 function has return stmt
- 1 function has show stmt

- 1 main module missing show stmt (or uses print
- 1 main module missing load_pic stmt
- 1 main module missing function call
- 3 all done within the main module