Question 1. [2 MARKS]

Part (a) [1 MARK] What is the output of the following?

```
pic = media.create_picture(50, 100)
pic2 = media.add_rec_filled(pic, 0, 0, 10, 20, media.yellow)
print type(pic2)
```

`None/NoneType` OR `Error (because of the typo; should be add_rect_filled)`

Part (b) [1 MARK] Rewrite the following code without an `if`-statement.

```
if not skates and helmet:
    return True
else:
    return False
```

```
return not skates and helmet
```

Question 2. [2 MARKS]

In each question below, fill in the box with python code that will make the program behaviour match the comments. You may **not** make any other changes to the code.

Part (a) [1 MARK]

```
day = 16
month = 'February'

# Print the following: The 16th of February.

print 'The %dth of %d.' % (day, month)
```

Part (b) [1 MARK]

```
pic = media.load_picture(media.choose_file())

# create a color with RGB values 50, 100, 150

color = media.create_color(50, 100, 150)

media.set_color(media.get_pixel(pic, 0, 0), color)
```

Question 3. [8 MARKS]

Part (a) [4 MARKS] Complete the following function according to its docstring description.

```
def change_blue(pic, quotient):
    '''(Picture, float) -> Picture
    Return a new picture that is a copy of pic, but with each pixel's blue color
    component set to its original value divided by quotient.  quotient is a value
    between 1.0 and 100.0, inclusive.'''

    new_pic = media.copy(pic)

    for pixel in new_pic:
        blue = media.get_blue(pixel)
        new_blue = int(blue / quotient)
        media.set_blue(pixel, new_blue)

    return new_pic
```

Part (b) [4 MARKS]

Write a main block that allows the user to choose a file, prompts the user with, `'Enter a value between 1.0 and 100.0, inclusive:  '`, applies the `change_blue` function from part (a) to the picture in that file using the value entered by the user, and displays the resulting picture. You may assume that the user chooses a valid picture file and enters a valid value.

```
if __name__ == '__main__':

    pic = media.load_picture(media.choose_file())
    quotient = float(raw_input('Enter a value (between 1.0 and 100.0): '))
    new_pic = change_blue(pic, quotient)
    media.show(new_pic)
```

Question 4. [8 MARKS]

Consider the following two .py files, which are saved in the same directory (folder).

module_a.py:

```
def f(s):
    result = ''

    for char in s:
        if not char.isdigit():
            result = result + char

    return result

if __name__ == '__main__':
    print f('34d')

# this code is not inside the
# body of the if-statement
print f('a1b2c')
```

module_b.py:

```
import module_a

def g(s):
    answer = module_a.f(s)
    return answer[0]

if __name__ == '__main__':
    print module_a.f('98ef7')
    print g('5f56g')
```

This question continues on the next page. You may use the space below for rough work.

Part (a) [1 MARK]

How many lines of output are produced when `module_b` is executed (by clicking Run)?

Circle one: 2 lines **3 lines** 4 lines

Part (b) [4 MARKS]

In the table below, show the output from running `module_b`. If there are fewer than four lines of output, leave the unused box(es) empty.

`abc`
`ef`
`f`

Part (c) [3 MARKS]

Write a good docstring for the function `f` from `module_a`.

```
(str) -> str
Return the characters from s that are not digits.
```