Question 1. [7 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. If there is nothing that you could write in the box to make the program work as specified, put a big **X** through the box.

Part (a) [1 MARK]

```
answer = ''
while [answer != 'Leafs' and answer != 'Habs'] :

    answer = raw_input('Leafs or Habs? ')

# If we get to this line, answer is either Leafs or Habs.
```

Part (b) [1 MARK]

```
s = "big old cat"
# Replace 'old' with 'fat' so that s becomes 'big fat cat'
```

`s[4:7] =` NOT POSSIBLE - CROSS OFF BOX

Part (c) [1 MARK]

Assume `funny` and `slob` are boolean variables indicating characteristics of your room-mate.

```
# Set boolean variable happy to True if your room-mate is funny but isn't a slob,
# and False otherwise.
```

happy = funny and not slob

Part (d) [1 MARK]

```
name = raw_input("Enter your name: ")
# If name is longer than 20 characters, shorten it to only the first 20 characters.

if [len(name) > 20] :

   name = name[:20]
```

Part (e) [1 MARK]

```
def is_char_at(s, ch, x):
    '''Return True iff string s has character ch at index x.'''
    return s[x] == ch

course = raw_input("What is your favourite course? ")
# Use the function is_char_at to check if the course starts with 'C' and print a
# message if it does.

if is_char_at(course, 'C', 0):

    print 'Your favourite could be CSC 108!'
```

Part (f) [1 MARK]

```
# Students have finished 3 labs so far and each lab grade is either 0 or 1.

lab1 = int(raw_input('Please enter your lab1 grade: '))
lab2 = int(raw_input('Please enter your lab2 grade: '))
lab3 = int(raw_input('Please enter your lab3 grade: '))

# Calculate and print the average lab mark for this student.

print (lab1 + lab2 + lab3 ) / 3.0
```

Part (g) [1 MARK]

```
def raise_to_A(grade):
```

NOT POSSIBLE - CROSS OFF BOX :

```
mark = 79.4
#raise this student's mark to 80.0
raise_to_A(mark)
#print the 80.0
print mark
```

Question 2. [8 MARKS]

Part (a) [4 MARKS]

Complete the following function according to its docstring.

```
def mute_channel(snd, channel):
    '''String channel is either 'right' or 'left'. Set all
    the sample values of Sound snd to zero of the corresponding channel.'''

    for sample in snd:
        if channel == 'right':
            sound.set_right(sample, 0)
        else:
            sound.set_left(sample, 0)
```

Part (b) [4 MARKS]

Write a main block that allows the user to choose a file, and asks the user using the prompt "Which channel? (enter left or right):" to specify which channel to mute. It then plays the sound with the appropriate channel muted. Assume that the user chooses a `.wav` file, and enters a valid answer to your question about which channel.

Make sure that if this module is imported, none of this code executes – just the function definition.

```
if __name__ == '__main__':
    file = media.choose_file()
    snd = sound.load_sound(file)
    channel_to_mute = raw_input("Which channel? (enter left or right:")
    mute_channel(snd, channel_to_mute)
    sound.play(snd)
```

Question 3. [5 MARKS]

Consider this Python program.

```
def mystery(s, c, n):

    result = ''
    for ch in s:
        if ch == c:
            result += c * n
        else:
            result += ch
    return result

if __name__ == '__main__':
    print mystery('Diane', 'a', 3)
    print mystery('Michelle', 'e', 2)
    print mystery('Dan', 'n', 0)
    print mystery('CSC108 instructors', 'c', 3)
```

Part (a) [2 MARKS]

The output from the program is four lines. Show it in this table.

Diaaane
Micheellee
Da
CSC108 instruccctors

Part (b) [3 MARKS]

Write a good docstring for the function.

```
'''Return a copy of string s where every occurence of character c
has been replaced by n copies of c. '''
```