SECTION A

This section is for **Python**, please answer the questions according to the Python version that we are using in class, namely, version 3.6.0.

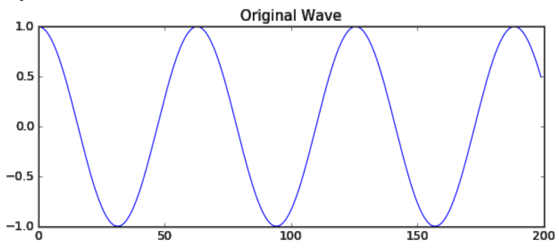
Question 1 [4 marks]

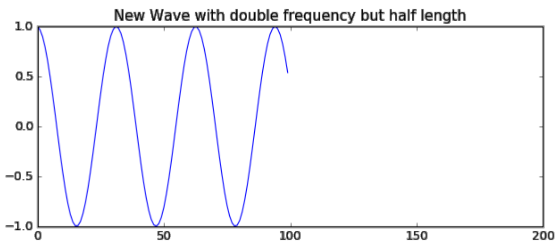
Evaluate the following terms. If any of these causes an error, please write "error" instead.

Evaluate the Followings:	Answer:
`2 ** False`	`1`
`1 + 2 * 3 ** 2`	`19`
`'abc'[2]`	`'c'`
`int(1.6)`	`1`
`'abcdef'[1::3]`	`'be'`
`not ''`	`True`
`[1,2,3,(4,5,6),7][3][2]`	`6`
`--1`	`1`

Question 2 [4 marks]: Doubling the frequency of a wave

You are given a wave like your lab exercise before. The wave is stored as a list of numbers. This time, your job is to double the frequency of a wave but shortening its length by half.





Write a function ‘`double_freq(wave)`’ to return a new wave with double frequency and half of the original length following the details below:

- For every position in the original wave with an even index, we average this sample with the next one to produce the sample in the new wave.
- Let `len(wave)` be L. You can assume that L > 0 and L is even.
- Your new wave should be a list with length L/2.
- As in the lab before, you should **NOT** modify the original wave input
- You should NOT use any packages at all.

```
def double_freq(wave):
    l = len(wave)
    new_wave = list(range(int(l/2)))
    for i in range(int(l/2)):
        new_wave[i] = (wave[i*2] + wave[i*2+1])/2

    return new_wave
```

Question 3 [4 marks]

Each row of the table is a separate program/file. What is the output of each of them? If the code produces errors or runs into infinite loops, please state ‘error’ or ‘infinite loop’ respectively.

Code	Output

`l = [1,2]`
`a = 1`
 	
`def double(x):`
`    x += x`
 	
`double(l)`
`double(a)`
 	
`print(l)`
`print(a)`
	`[1, 2, 1, 2]`
	`1`

`b = 128`
 	
`def half(x):`
`    return x / 2`
 	
`print(half(half(half(b))))`
	`16.0`

`c = (2,)`
 	
`def complicate(x):`
`    if x[0]==0 :`
`        return x`
`    else:`
`        return complicate([x[0]-1,x])`
 	
`print(complicate(c))`
	`[0, [1, (2,)]]`

`d = {}`
`s = 'IT1007'`
`for i in range(len(s)):`
`    d[s[i]] = i`
`print(d)`
	`{'I': 0, 'T': 1, '1': 2,`
	`'0': 4, '7': 5}`

Question 4 [4 marks]

Write a function to calculate the mean RGB values for a picture. Your function should return a list with three values as red, green and blue respectively. The first few lines of code are written for you. To gain the full mark:

- You are NOT allowed to use functions from any packages such as the functions '`mean()`' and '`average()`' in package `numpy`, other than the line with '`imread`' which is written for you.
- And also, cannot use the built-in function '`sum()`'. Sorry….

Sample output:

```
>>> print(meanRGB('Ji_Chang_Wook.jpg'))
[ 234.30944011  231.27928453  232.10331584]
```

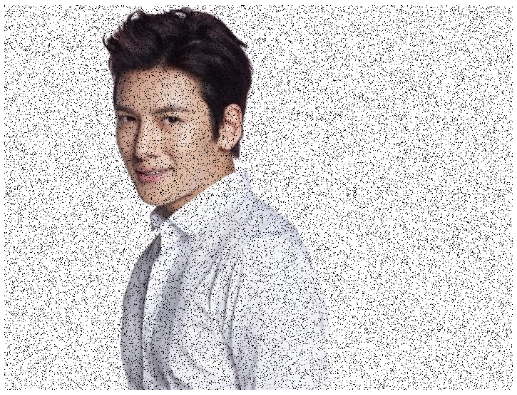
Answer:

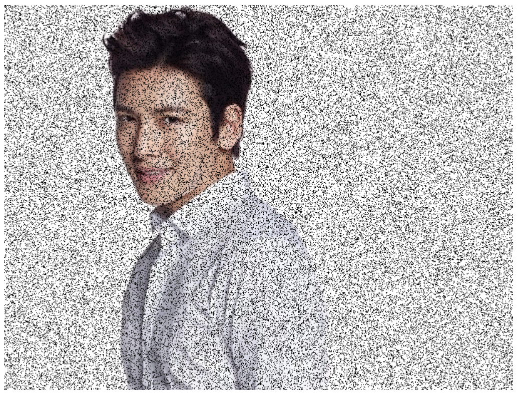
```
from scipy import misc,ndimage

def meanRGB(filename):
    pic = misc.imread(filename)

    RGB = [0,0,0]

    for i in range(pic.shape[0]):
        for j in range(pic.shape[1]):
            RGB += pic[i][j]
    RGB = RGB / (pic.shape[0]*pic.shape[1])

    return RGB
```

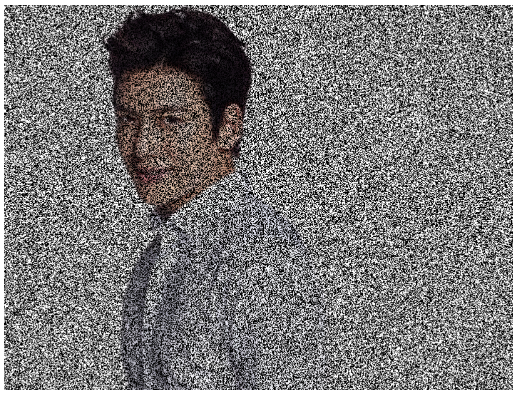
Question 5 [4 marks]

Write a function to add noise into an image. Your function will take in a filename and an integer p for $0 \leq p \leq 100$, such that every pixel of the image will have a chance of %p to become totally black. And, of course, you cannot use any already-built 'noise' functions from packages. Here are some example outputs.

Original Image


p = 10


p = 20


p = 50

Answer:

```
from scipy import misc,ndimage
import numpy as np
import random

def add_noise(filename,percent):
    pic = misc.imread(filename)

    for i in range(pic.shape[0]):
        for j in range(pic.shape[1]):
            if random.randint(0,100)<percent:
                pic[i][j] = 0

    return pic
```