Dictionary – part 1

1. Bitwise operation

Shifting, bitwise &, bitwise |

<< left
>> right

x & 0 = 0
&: x & 1 = x
|: x | 0 = x
x | 1 = 1

Exercise
Take the last two bits from red and return it

```
def getLast2(red):
```

return red & 0b00000011

red: 0b x_8 x_7 x_6 x_5 x_4 x_3 x_2 x_1
return: 0b x_2 x_1

red: 0b x_8 x_7 x_6 x_5 x_4 x_3 x_2 x_1
& 0b 0 0 0 0 0 0 1 1
0b 0 0 0 0 0 0 x_2 x_1

Exercise
Put the first two bits of red1 into the last two bits of red2

```
def put2Digits(red1, red2):
```

tmp1 = red1 >>6
tmp2 = red2 & 0b11111100
return tmp1 | tmp2

red1: x_8 x_7 x_6 x_5 x_4 x_3 x_2 x_1
red2: y_8 y_7 y_6 y_5 y_4 y_3 y_2 y_1
result: y_8 y_7 y_6 y_5 y_4 y_3 x_8 x_7

red1: & x_8 x_7 x_6 x_5 x_4 x_3 x_2 x_1
1 1 0 0 0 0 0 0
x_8 x_7 0 0 0 0 0 0

red1 & 0b11111100
>>6
0 0 0 0 0 0 x_8 x_7

red2 & 0b11111100
y_8 y_7 y_6 y_5 y_4 y_3 y_2 y_1
1 1 1 1 1 1 0 0
y_8 y_7 y_6 y_5 y_4 y_3 0 0

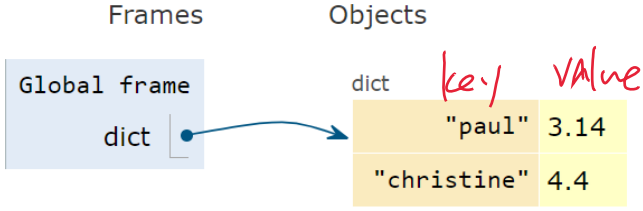
2. Dictionary

A dictionary gives a mapping from one category to another. For example, you can map

- name (str) to salary (float)
- zip code (int) to city name (str)
- weight (float) to name (str)
- Email (str) to gpa (float)

key to value

```
dict = {"paul":3.14, "christine": 4.40}
```



key: value

dictionary ops:
find, insert, delete

Student = { "A12345678" : ['Paul', 'Cao', 3.14, 'CSE'] }

Exercise: Please create a dictionary called salary and put in the following data:

```
'paul cao', 12334
'mia minnes', 43211
'rick ord', -5
```

salary = { 'paul cao': 12334, 'mia minnes': 43211, 'rick ord': -5 }

What is the outcome of the code snippet below?

```
zipcodes = {92122:"San Diego", 12345:"Schenectady", 92093:"UCSD"}
zipcodes["12345"]
```

zipcodes[12345] A

A) Schenectady
B) San Diego
C) UCSD
D) KeyError: The key '12345' is not found in the dictionary

What is the outcome of the code snippet below?

```
zipcodes = {92122:"San Diego", 12345:"Schenectady", 92093:"UCSD"}
zipcodes["UCSD"] #How about zipcodes[1]?
```

same result

dict[a key]

A) 92122
B) 12345
C) 92093
D) KeyError: The key 'UCSD' is not found in the dictionary

What is the outcome of the code snippet below?

```
zipcodes = {92122:"San Diego", 12345:"Schenectady", 92093:"UCSD"}
zipcodes[12345] = "paul city"
print(zipcodes[12345])
```

zipcodes →

92122	'SD'
12345	~~'S...y'~~ 'paul city'
92093	'UCSD'

dict[key] = value
insert if key isn't there
update if key exists

A) San Diego
B) Schenectady
C) UCSD
D) paul city
E) KeyError: The key 12345 is not found in the dictionary

What will be printed by the code snippet below?

```
zipcodes = {92122:"San Diego", 12345:"Schenectady", 92093:"UCSD"}
for item in zipcodes:
    print(item)
```

A)	B)	C)	D)
92122 12345 92093	San Diego **Schenectady** UCSD	92122:San Diego **12345:Schenectady** **92093:UCSD**	San Diego:92122 **Schenectady:12345** **UCSD:92093**

What is the result of the following expression?

```
zipcodes = {92122:"San Diego", 12345:"Schenectady", 92093:"UCSD"}
12345 in zipcodes
```

A) True

B) False

C) None

D) Error: in operator cannot be applied on dictionaries

Coding challenge

Write a function that takes in an image and returns the frequency of different pixel colors in the picture as a dictionary. For example, if you are given the following image,

```
[[(255, 255, 255), (255, 255, 255), (0, 0, 0), (0, 0, 0), (0, 0, 0)],
 [(255, 255, 255), (255, 255, 255), (0, 0, 0), (0, 0, 0), (0, 0, 0)],
 [(0, 0, 0), (0, 0, 0), (0, 0, 0), (0, 0, 0), (0, 0, 0)],
 [(0, 0, 0), (0, 0, 0), (0, 0, 0), (0, 0, 0), (0, 0, 0)],
 [(0, 0, 0), (0, 0, 0), (0, 0, 0), (0, 0, 0), (0, 0, 0)]]
```

your code should return

```
{(255, 255, 255): 4, (0, 0, 0):21}
```

```
def color_freq_count(img):
  # write code here
```