Linear Algebra Homework

In this homework, we will be manipulating a colored image
The jpg file used have the same directory as the notebook

Libraries used :

- We will be using the **Pillow** library to manipulate the .jpg file (only to open the file, in this case)
- We will be using the **NumPy** library to convert our image to an array and operate on it using matrix operations and what we have learned about this library during the bootcamp

Let's start by opening our image using Pillow and converting to an array using NumPy asarray function

The image used is in the RGB format, we except to get a large array containing [R , G , B] data with integers from 0 to 256

In [117…

```
import numpy as np
from PIL import Image

file = Image.open('buildings.jpg')
image = asarray(file)
print(image)
```

```
[[[192 221 253]
  [192 221 253]
  [192 221 253]
  ...
  [193 222 252]
  [193 222 252]
  [193 222 252]]

 [[192 221 253]
  [192 221 253]
  [192 221 253]
  ...
  [193 222 252]
  [193 222 252]
  [194 223 253]]

 [[192 221 253]
  [192 221 253]
  [192 221 253]
  ...
  [193 222 252]
  [193 222 252]
  [194 223 253]]

 ...

 [[ 39  40  44]
  [ 39  40  44]
  [ 40  41  45]
  ...
  [ 61  43  41]
  [ 65  47  45]
  [ 70  52  50]]
```

```
[[ 39  40  44]
 [ 39  40  44]
 [ 40  41  45]
 ...
 [ 53  43  42]
 [ 58  46  46]
 [ 62  50  50]]

[[ 39  40  44]
 [ 39  40  44]
 [ 39  40  44]
 ...
 [ 53  47  47]
 [ 59  50  51]
 [ 62  53  54]]]
```

Now that we have loaded our image as an array, we can operate on it and answer what's asked for in the homework using NumPy functions

1. Transpose the array
2. Check if the matrix is one or many of the special types we have seen (diagonal, triagonal...).
3. Choose a column vector and calculate both norms that we've seen.

In [112…

```python
# 1. Transpose

transposed_image = image.T
print(transposed_image)
```

```
[[[192 192 192 ...  39  39  39]
  [192 192 192 ...  39  39  39]
  [192 192 192 ...  40  40  39]
  ...
  [193 193 193 ...  61  53  53]
  [193 193 193 ...  65  58  59]
  [193 194 194 ...  70  62  62]]

 [[221 221 221 ...  40  40  40]
  [221 221 221 ...  40  40  40]
  [221 221 221 ...  41  41  40]
  ...
  [222 222 222 ...  43  43  47]
  [222 222 222 ...  47  46  50]
  [222 223 223 ...  52  50  53]]

 [[253 253 253 ...  44  44  44]
  [253 253 253 ...  44  44  44]
  [253 253 253 ...  45  45  44]
  ...
  [252 252 252 ...  41  42  47]
  [252 252 252 ...  45  46  51]
  [252 253 253 ...  50  50  54]]]
```

In [113…

```python
# 2.Type of the matrix
# Check if it is triangular upper,lower or diagonal

tril_image = np.tril(image)
triu_image = np.triu(image)

print('the array is triangular lower :', np.array_equal(image,tril_image))

print('the array is triangular upper :', np.array_equal(image,tril_image))
```

```
print('the array is diagonal :', np.array_equal(image,tril_image) or np.array_equal(image
```

```
the array is triangular lower : False
the array is triangular upper : False
the array is diagonal : False
```

In [116…

```
# 3. Let's choose a random column and calculate its norm1 and 2 :

random_index_1 = np.random.randint(image.shape[1])
random_index_2 = np.random.randint(image.shape[2])

column= image[:, random_index_1, random_index_2]

print('We picked the column [:,{},{}]= {} '.format(random_index_1, random_index_2, column


from numpy.linalg import norm
norm_1 = norm(column, 1)
norm_2 = norm(column, 2)

print('\nThe norms of the column :\n Norm 1 = {}\n Norm 2 = {}'.format(norm_1,norm_2))
```

```
We picked the column [:,1546,1]= [226 226 226 ... 118 109 102]

The norms of the column :
 Norm 1 = 545544.0
 Norm 2 = 10332.318616844914
```