Name: Sumanasekara W.K.G.G.

Index: 190610E

In [ ]:

```
import sympy
import numpy as np
from numpy import linalg
import cv2 as cv
import matplotlib.pyplot as plt
```

In [ ]:

```
for i in range(1, 6):
    print(i, ":", i**2)
```

```
1 : 1
2 : 4
3 : 9
4 : 16
5 : 25
```

In [ ]:

```
for i in range(1, 6):
    if not sympy.isprime(i):
        print(i, ":", i**2)
```

```
1 : 1
4 : 16
```

In [ ]:

```
squares = [i**2 for i in range(1, 6)]
squares
```

Out[ ]:

```
[1, 4, 9, 16, 25]
```

In [ ]:

```
Unprime_squares = [i**2 for i in range(1, 6) if not sympy.isprime(i)]
Unprime_squares
```

Out[ ]:

```
[1, 16]
```

In [ ]:

```
A = np.array([[1, 2], [3, 4], [5, 6]])
B = np.array([[7, 8, 9, 1], [1, 2, 3, 4]])
D = np.array([[3, 2], [5, 4], [3, 1]])
```

In [ ]:

```
# Matrix multiplication
C = np.matmul(A, B)
C
```

Out[ ]:

```
array([[ 9, 12, 15,  9],
       [25, 32, 39, 19],
       [41, 52, 63, 29]])
```

In [ ]:

```
# Element wise multiplication
E = np.multiply(A, D)
E
```

Out[ ]:
```
array([[ 3,  4],
       [15, 16],
       [15,  6]])
```

In [ ]:
```
# random matrix using numpy
mat = np.random.randint(10, size=(5, 7))
mat
```

Out[ ]:
```
array([[4, 8, 3, 4, 1, 5, 4],
       [0, 5, 8, 5, 4, 0, 7],
       [8, 1, 8, 5, 3, 9, 1],
       [8, 9, 4, 2, 8, 9, 9],
       [7, 1, 0, 7, 1, 4, 3]])
```

In [ ]:
```
# row 2 to 4 and column 1
mat1 = mat[1:4, 0:1]
print("Matrix =", mat1)
print("Shape =", mat1.shape)
```

```
matrix = [[0]
 [8]
 [8]]
Shape = (3, 1)
```

In [ ]:
```
# row 2 to 4 and first two columns
mat2 = mat[1:4, 0:2]
print("Matrix =", mat2)
print("Shape =", mat2.shape)
```

```
Matrix = [[0 5]
 [8 1]
 [8 9]]
Shape = (3, 2)
```

In [ ]:
```
# Broadcasting
A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
B = np.array([[1, 5, 9]])

# example 1 - addition of matrix with a row matrix
C = A+B
C
```

Out[ ]:
```
array([[ 2,  7, 12],
       [ 5, 10, 15],
       [ 8, 13, 18]])
```

In [ ]:
```
# example 2 - multiplication a matrix by a row matrix
D = A*B
D
```

Out[ ]:
```
array([[ 1, 10, 27],
       [ 4, 25, 54],
       [ 7, 40, 81]])
```

In [ ]:
```
# example 3 - addding a constant the matrix
```

```
E = A+10
E
```

Out[ ]:
```
array([[11, 12, 13],
       [14, 15, 16],
       [17, 18, 19]])
```

In [ ]:
```
m, c = 2 , -4
N = 10
x = np.linspace(0 , N-1, N).reshape (N, 1)
sigma = 10
y = m*x + c + np.random.normal (0, sigma, (N, 1))

x = np.append(np.ones((N, 1)),x, axis=1)
x
```

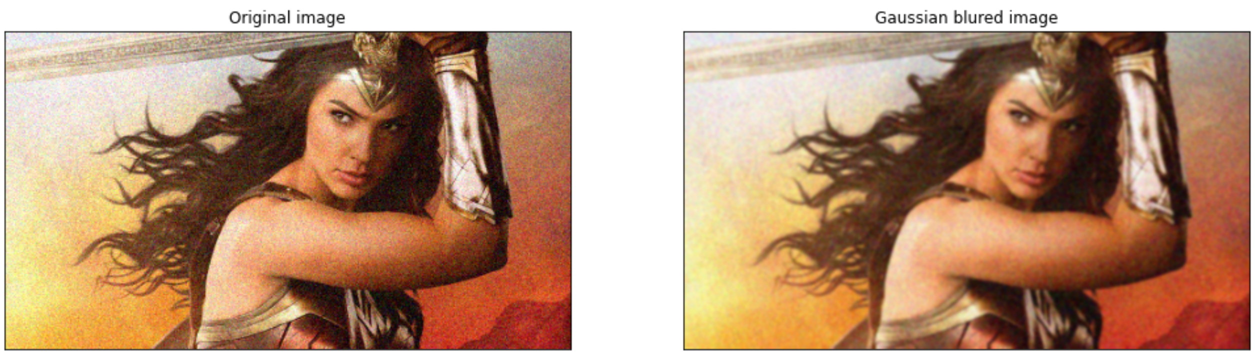
Out[ ]:
```
array([[1., 0.],
       [1., 1.],
       [1., 2.],
       [1., 3.],
       [1., 4.],
       [1., 5.],
       [1., 6.],
       [1., 7.],
       [1., 8.],
       [1., 9.]])
```

In [ ]:
```
(np.linalg.inv(x.T @ x)) @ x.T @ y
```

Out[ ]:
```
array([[-9.1611146 ],
       [ 3.33249295]])
```

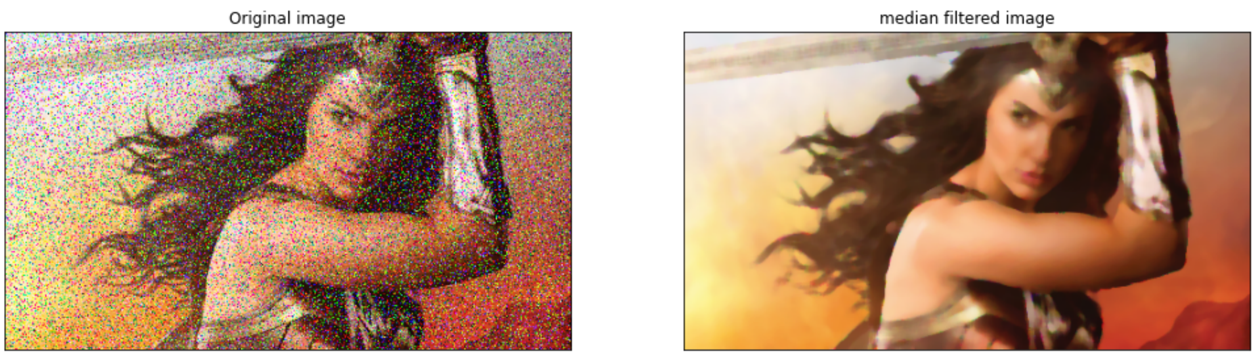
In [ ]:
```
img = cv.imread("gal_gaussian.png")
blur = cv.GaussianBlur(img, (5, 5), 0)

fig, ax = plt.subplots(1, 2, figsize = (17, 10))
ax[0].imshow(img[...,::-1])
ax[1].imshow(blur[...,::-1])
ax[0].get_xaxis().set_visible(False)
ax[0].get_yaxis().set_visible(False)
ax[0].set_title("Original image")
ax[1].get_xaxis().set_visible(False)
ax[1].get_yaxis().set_visible(False)
ax[1].set_title("Gaussian blured image")
plt.show()
```



In [ ]:
```
img = cv.imread("gal_sandp.png")
blur = cv.medianBlur(img, 7)

fig, ax = plt.subplots(1, 2, figsize = (17, 10))
ax[0].imshow(img[...,::-1])
ax[1].imshow(blur[...,::-1])
ax[0].get_xaxis().set_visible(False)
ax[0].get_yaxis().set_visible(False)
ax[0].set_title("Original image")
ax[1].get_xaxis().set_visible(False)
ax[1].get_yaxis().set_visible(False)
ax[1].set_title("median filtered image")
plt.show()
```

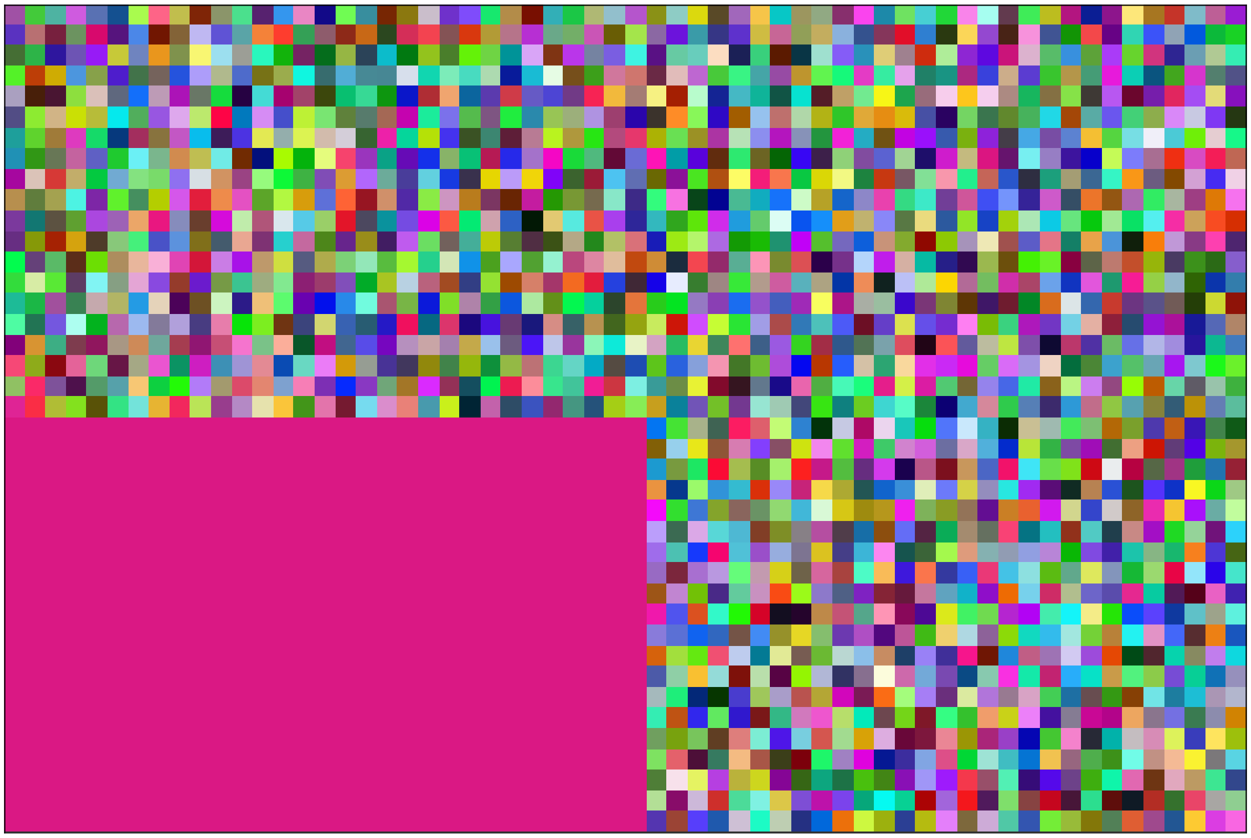


In [ ]:
```
my_img = np.zeros((40, 60), dtype=np.uint8)
my_img[0:21, 30:] = 125

fig, ax = plt.subplots(figsize = (17, 10))
ax.imshow(my_img, cmap="gray", vmin=0, vmax=255)
ax.get_xaxis().set_visible(False)
ax.get_yaxis().set_visible(False)
plt.show()
```

In [ ]:
```
my_img = np.random.randint(255, size=(40, 60, 3), dtype=np.uint8)
my_img[20:, :31] = [218, 24, 132]

fig, ax = plt.subplots(figsize = (17, 10))
ax.imshow(my_img)
ax.get_xaxis().set_visible(False)
ax.get_yaxis().set_visible(False)
plt.show()
```



In [ ]:
```
def changeBrightness(img, value):
    "function to chage the brightness"
    ## positive int for "value" increase the brightness
    ## negative int for "value" reduce the brightness
    hsv = cv.cvtColor(img, cv.COLOR_BGR2HSV)
    h, s, v = cv.split(hsv)

    limit = 255 - value
    v[v > limit] = 255
    v[v <= limit] += value

    final_hsv = cv.merge((h, s, v))
    img = cv.cvtColor(final_hsv, cv.COLOR_HSV2BGR)
    return img
```

In [ ]:
```
img = cv.imread("tom_dark.jpg")
bright = changeBrightness(img, 100)

fig, ax = plt.subplots(1, 2, figsize = (17, 10))
ax[0].imshow(img[...,::-1])
ax[1].imshow(bright[...,::-1])
ax[0].get_xaxis().set_visible(False)
ax[0].get_yaxis().set_visible(False)
ax[0].set_title("Original image")
ax[1].get_xaxis().set_visible(False)
ax[1].get_yaxis().set_visible(False)
ax[1].set_title("Image after increasing the brightness")
```

Out[ ]:
```
Text(0.5, 1.0, 'Image after increasing the brightness')
```

Original image

Image after increasing the brightness