Name : Vakeesan.K

Index no. : 190643G

1. Write a program to display the squares of numbers form 1 to 5 as seen below.

In [ ]:

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

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

2. Alter the code in 14 to print the square only of non-prime numbers. Use the isprime function in the sympy package for testing if a number is a prime.

In [ ]:

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

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

3. Use a comprehension to do 14.

In [ ]:

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

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

4. Use a comprehension to do 11.

In [ ]:

```
sqr_2=[i**2 for i in range(1,6) if not sympy.isprime(i)]
print(sqr_2)
```

```
[1, 16]
```

5. Using NumPy carry out the following computations.

In [ ]:

```
import numpy as np
A=np.array([[1,2],[3,4],[5,6]])
B=np.array([[7,8,9,1],[1,2,3,4]])
print(np.dot(A,B))
print(A @ B)
print(np.matmul(A,B))
```

```
[[ 9 12 15  9]
 [25 32 39 19]
 [41 52 63 29]]
[[ 9 12 15  9]
 [25 32 39 19]
 [41 52 63 29]]
[[ 9 12 15  9]
 [25 32 39 19]
 [41 52 63 29]]
```

In [ ]:

```
A=np.array([[1,2],[3,4],[5,6]])
B=np.array([[3,2],[5,4],[3,1]])
print(A * B)
```

```
[[ 3  4]
 [15 16]
 [15  6]]
```

6. Generate a 5×7 array of random integers in the interval [0,10] and extract the sub array consisting rows 2 to 4 and columns 1 and first two columns. What is the size of the resulting array?

In [ ]:

```
rand_array=np.random.randint(11, size=(5, 7))
print(rand_array)
arr=rand_array[1:4,0:2]
print(arr)
np.size(arr)
arr.shape
```

```
[[ 4  5  4  1  8  3  4]
 [ 1  0  4  3  9  0  8]
 [ 4  7 10 10  0  6  9]
 [ 1  0  2 10  6  3  9]
 [ 2  8  4  6  3  3  8]]
[[1 0]
 [4 7]
 [1 0]]
```

Out[ ]:

```
(3, 2)
```

7. Show three examples of broadcasting.

In [ ]:

```
x = np.array([12,24,36])
y = np.array([45 , 55])

print(np.reshape(x,(3,1))*y)

z = np.array([[12, 22, 33],[45,55, 66]])
print(x+z)

print(x*z)
```

```
[[ 540  660]
 [1080 1320]
 [1620 1980]]
[[ 24  46  69]
 [ 57  79 102]]
[[ 144  528 1188]
 [ 540 1320 2376]]
```

8. Consider the following code snippet.

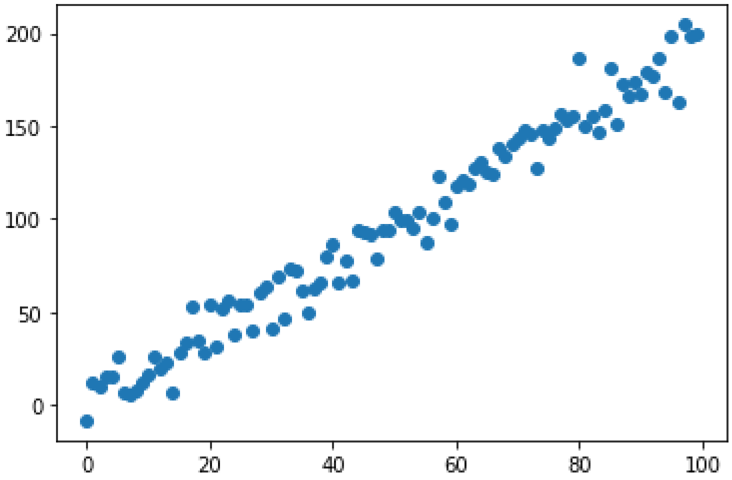
In [ ]:

```
import matplotlib.pyplot as plt
from numpy import linalg
m, c = 2 , -4
N = 100
x = np . linspace (0 , N-1, N) . reshape (N, 1 )
sigma = 10
y = m*x + c + np . random. normal (0 , sigma , (N, 1 ) )
plt.scatter(x,y)

X=np.append(np.ones([N,1]),x,axis = 1)

w = linalg.inv(X.T @ X) @ X.T @ y
print(w)
```

```
[[-0.55834633]
 [ 1.94994059]]
```



10. Use Gaussian smoothing to filter the noise in the image gal_gaussian.png shown in Fig. 13.

In [ ]:

```
import cv2 as cv
im = cv.imread(r'C:\Python39\cv\exercices\lec1\image gal_gaussian.jpg')
assert im is not None


g_smooth = cv.GaussianBlur(im,(5,5),cv.BORDER_DEFAULT)


cv.namedWindow('Image',cv.WINDOW_AUTOSIZE)
cv.imshow("Image",im)
cv.waitKey(0)

cv.imshow("Image",g_smooth)
cv.waitKey(0)


cv.destroyAllWindows()
```

11. Use median filtering to filter the salt and pepper noise in the image gal_sandp.png shown in Fig. 2.

In [ ]:

```
import cv2 as cv
im_2 = cv.imread(r"C:\Python39\cv\exercices\lec1\image gal_saltp.jpg")
```

```
assert im_2 is not None

m_filter = cv.medianBlur(im_2,5)

cv.imshow("Image",im_2)
cv.waitKey(0)

cv.imshow("Image",m_filter)
cv.waitKey(0)
cv.destroyAllWindows()
```

12. Create a 40×60 grayscale image and change the color of the top-right quarter to 125.

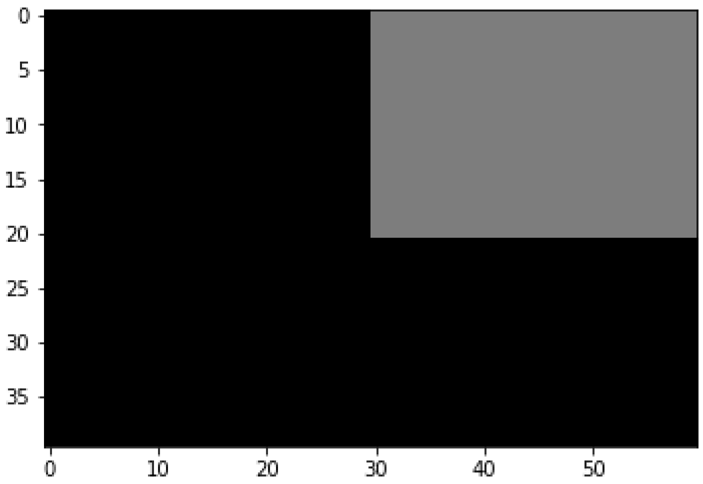
In [ ]:

```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt

im = np.zeros((40,60), dtype=np.uint8)
assert im is not None

im[0:21, 30:61] = 125

fig, ax = plt.subplots()
ax.imshow(im, cmap='gray', vmin=0, vmax=256)
plt.show()
"""cv.imshow("Image", im)
cv.waitKey(0)
cv.destroyAllWindows"""
```



Out[ ]:

```
'cv.imshow("Image", im)\ncv.waitKey(0)\ncv.destroyAllWindows'
```

13. Create a 40×60 color image and change the color of the bottom-left quarter to “Barbie Pink”4.

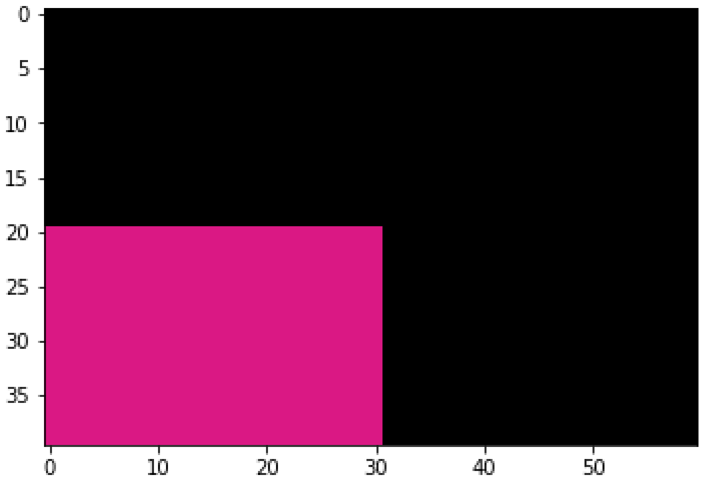
In [ ]:

```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt

im = np.zeros((40,60,3), dtype=np.uint8)
assert im is not None

im[20:41, 0:31] = [218, 24, 132]
```

```
fig, ax = plt.subplots()
ax.imshow(im, cmap='gray', vmin=0, vmax=256)
plt.show()
```



14. Increase the brightness of the image tom_dark.jpg shown in Fig. 3.

In [ ]:

```
import cv2 as cv
im = cv.imread(r"C:\Python39\cv\exercices\lec1\tom_dark.jpg")
assert im is not None

bright_tom = im + 128
cv.imshow('Image',im)
cv.waitKey(0)

cv.imshow("Image",bright_tom)
cv.waitKey(0)

cv.destroyAllWindows()
```

In [ ]:

```

```