Name : A.M.A.D. ADIKARI

Index No : 190021A

In [ ]:

```
# Question 01
# squares of numbers in the interval [1, 5]
for i in range(1, 6):
    print(i, ': ', i**2)
```

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

In [ ]:

```
# Question 02
# squares of non-prime numbers in the interval [1, 5]
import sympy
for i in range(1, 6):
    if not sympy.isprime(i):
        print(i, ': ', i**2)
```

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

In [ ]:

```
# Question 03
# Question 01 with comprehension
squares = [i**2 for i in range(1, 6)]
for count, value in enumerate(squares, 1):
    print(count, ': ', value)
```

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

In [ ]:

```
# Question 04
# Question 02 with comprehension
squares = [i for i in range(1, 6) if not sympy.isprime(i)]
for i in squares:
    print(i, ': ', i**2)
```

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

In [ ]:

```
# Question 05
# part (a)
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.matmul(A, B))
```

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

In [ ]:

```
# part (b)
A = np.array([[1, 2], [3, 4], [5, 6]])
B = np.array([[3, 2], [5, 4], [3, 1]])
print(np.multiply(A, B))
```

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

In [ ]:

```
# Question 06
random_num = np.random.randint(0,11,(5,7))  # 5 x 7 matrix
subarray = random_num[1:4, :2]              # 3 x 2 matrix

print('random array\n', random_num)
print('sub array\n', subarray)
print('The size of the sub array :', np.shape(subarray))
```

```
random array
 [[ 4  3  8  9  7  5  4]
 [ 3  5  0  1  2  6  1]
 [ 3  0  7  1 10 10  6]
 [ 9  1  9  1 10  1  2]
 [ 4  4 10  3  3  0  8]]
sub array
 [[3 5]
 [3 0]
 [9 1]]
The size of the sub array : (3, 2)
```

In [ ]:

```
# Question 07
A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]]) # 3 x 3 matrix
B = np.array([1])                               # 1 x 1 matrix
C = np.array([1, 2, 3])                         # 1 x 3 matrix
D = np.array([[4], [5], [6]])                   # 3 x 1 matrix

# example 1
print('Example 1\n', A + B)

#example 2
print('Example 2\n', A + C)

# example 3
A_reshaped = np.transpose(A)                    # 3 x 3 matrix
print('Example 3\n', A_reshaped + D)
```

```
Example 1
 [[ 2  3  4]
 [ 5  6  7]
 [ 8  9 10]]
Example 2
 [[ 2  4  6]
 [ 5  7  9]
 [ 8 10 12]]
Example 3
 [[ 5  8 11]
 [ 7 10 13]
 [ 9 12 15]]
```

In [ ]:

```
# Question 08
import numpy as np
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))

# part (a)
X = np.append(np.ones((N, 1)), x, axis = 1)
print('X\n', X)

# part (b)
XT = np.transpose(X)
Z = np.matmul(np.matmul(np.linalg.inv(np.matmul(XT, X)), XT), y)
print('Z\n', Z)
```

```
X
 [[1. 0.]
 [1. 1.]
 [1. 2.]
 [1. 3.]
 [1. 4.]
 [1. 5.]
 [1. 6.]
 [1. 7.]
 [1. 8.]
 [1. 9.]]
Z
 [[-7.72993843]
 [ 2.01608644]]
```

In [ ]:

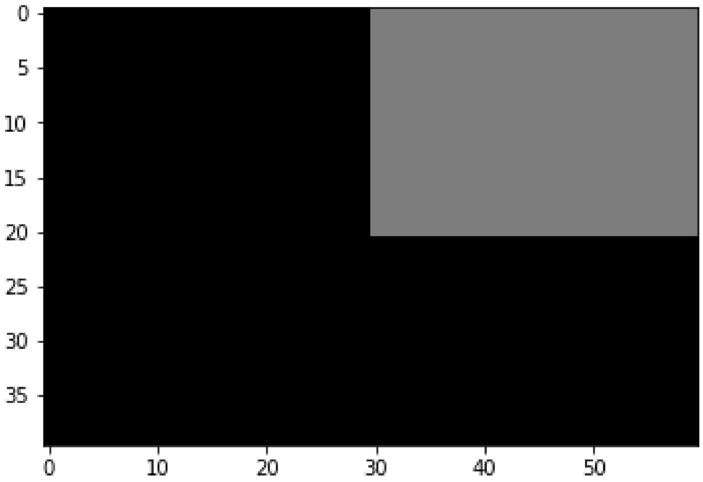
```
# Question 10
import cv2 as cv

img = cv.imread("gal_gaussian.png")

blur = cv.GaussianBlur(img, (5, 5), 0)

cv.namedWindow('Image', cv.WINDOW_AUTOSIZE)
cv.imshow('Image', img)
cv.waitKey(0)
cv.imshow('blurred Image', blur)
cv.waitKey(0)
cv.destroyAllWindows()
```

In [ ]:

```
# Question 11
image = cv.imread("gal_sandp.png")
blur = cv.medianBlur(image, 5)

cv.namedWindow("image", cv.WINDOW_AUTOSIZE)
cv.imshow("image", image)
cv.waitKey(0)
cv.imshow("blurred image",blur)
cv.waitKey(0)
cv.destroyAllWindows()
```

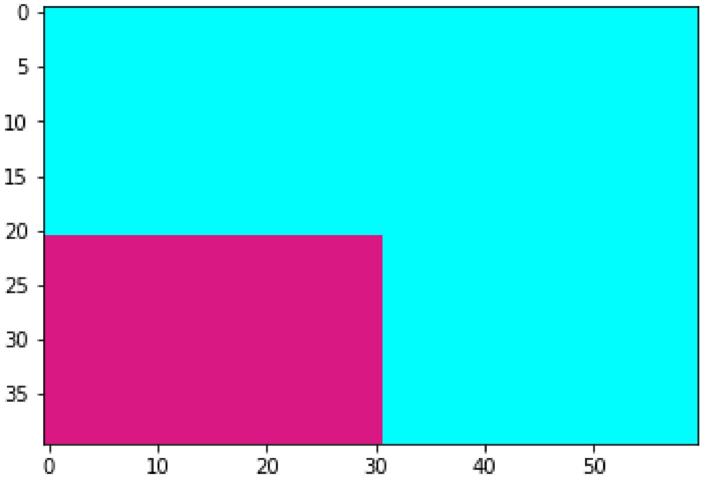
In [ ]:

```
# Question 12
image = np.zeros((40, 60), dtype=np.uint8)
image[0:21,30:] = 125

plt.imshow(image, cmap='gray', vmin=0, vmax=255)
plt.show()
```



In [ ]:

```
# Question 13
image = np.zeros( (40, 60, 3), dtype=np.uint8)
image[21:, :31] = (218, 24, 132)
image[:21,:] = (0, 255, 255)
image[21:, 31:61] = (0, 255, 255)

plt.imshow(image, cmap='pink', vmin=0, vmax=255)
plt.show()
```



In [ ]:

```
# Question 14
image = cv.imread("tom_dark.jpg")
bright_image = image + 25

cv.namedWindow("image", cv.WINDOW_AUTOSIZE)
cv.imshow("image", image)
cv.waitKey(0)
cv.imshow("bright image", bright_image)
cv.waitKey(0)
cv.destroyAllWindows()
```