In [1]:

```
#Enter each of the matrices A, B, and C as Numpy matrix.

#lod np
import numpy as np

#array matrices
A = np.array([[1, -2, 3, 7],[2, 1, 1, 4],[-3, 2, -2, 10]])

B = np.array([[0, 1, -3, -2],[10, -1, 2, -3],[5, 1, -1, 4]])

C = np.array([[4, 0, -2, 3],[3, 6, 9, 7],[2, 2, 5, 1],[9, 4, 6, -2]])

print("matrix A:")
print(A)

print("\nmatrix B:")
print(B)

print("\nmatrix C:")
print(C)
```

```
matrix A:
[[ 1 -2  3  7]
 [ 2  1  1  4]
 [-3  2 -2 10]]

matrix B:
[[ 0  1 -3 -2]
 [10 -1  2 -3]
 [ 5  1 -1  4]]

matrix C:
[[ 4  0 -2  3]
 [ 3  6  9  7]
 [ 2  2  5  1]
 [ 9  4  6 -2]]
```

In [2]:

```
#what are the dimensions of each of the matrices A, B, and C? Display these using Pyth

#lod np
import numpy as np

#array matrices
A = np.array([[1, -2, 3, 7],[2, 1, 1, 4],[-3, 2, -2, 10]])

B = np.array([[0, 1, -3, -2],[10, -1, 2, -3],[5, 1, -1, 4]])

C = np.array([[4, 0, -2, 3],[3, 6, 9, 7],[2, 2, 5, 1],[9, 4, 6, -2]])

print("dimensions of matrix A:", A.shape)
print("dimensions of matrix B:", B.shape)
print("dimensions of matrix C:", C.shape)
```

```
dimensions of matrix A: (3, 4)
dimensions of matrix B: (3, 4)
dimensions of matrix C: (4, 4)
```

1. Determine if each of the following matrix operations is defined. If it is defined, calculate the result using Python. If it is not defined, explain why it is not defined.

In [4]:

```
#(a) A + B = Matrices can be added since they have the same 3, 4 dimensions

try:
    A_plus_B = A + B
    print("A + B =\n", A_plus_B, 'The matrix is defined')
except ValueError:
    print("A + B is not defined because A and B have different dimensions.")
```

```
A + B =
 [[ 1 -1  0  5]
 [12  0  3  1]
 [ 2  3 -3 14]] The matrix is defined
```

In [6]:

```
#(b) AB = for matrix multiplication to be defines, number of columns in A must equal r

try:
    A_mul_B = np.matmul(A, B)
    print("\nAB =\n", A_mul_B, 'The matrix is defined')
except ValueError:
    print("\nAB is not defined because the number of columns in A doesn't match the nu
```

```
AB is not defined because the number of columns in A doesn't match the number of rows
in B.
```

In [8]:

```
#(c) AC = for matrix multiplication to be defines, number of columns in A must equal r

try:
    A_mul_C = np.matmul(A, C)
    print("\nAC =\n", A_mul_C, 'The matrix is defined')
except ValueError:
    print("\nAC is not defined because the number of columns in A doesn't match the nu
```

```
AC =
 [[ 67  22  37 -22]
 [ 49  24  34   6]
 [ 80  48  74 -17]] The matrix is defined
```

In [10]:

```
#(d)C^T = transpose of matrix C is defined

C_transpose = np.transpose(C)
print("\nC^T =\n", C_transpose,'The matrix is defined')
```

```
C^T =
 [[ 4  3  2  9]
 [ 0  6  2  4]
 [-2  9  5  6]
 [ 3  7  1 -2]] The matrix is defined
```

In [12]:

```
#4. Illustrate the following property of matrix transpositions using these matrices: (

#load np
import numpy as np

#matrices A and B
A = np.array([[1, -2, 3, 7], [2, 1, 1, 4], [-3, 2, -2, 10]])
B = np.array([[0, 1, -3, -2], [10, -1, 2, -3], [5, 1, -1, 4]])
```

```
#(A + B)^T
transpose_of_sum = np.transpose(A + B)
print("(A + B)^T =\n", transpose_of_sum)

#A^T + B^T
sum_of_transposes = np.transpose(A) + np.transpose(B)
print("\nA^T + B^T =\n", sum_of_transposes)

#equal?
are_equal = np.array_equal(transpose_of_sum, sum_of_transposes)
print("\nare they equal?", are_equal)
```

```
(A + B)^T =
 [[ 1 12  2]
 [-1  0  3]
 [ 0  3 -3]
 [ 5  1 14]]

A^T + B^T =
 [[ 1 12  2]
 [-1  0  3]
 [ 0  3 -3]
 [ 5  1 14]]

are they equal? True
```

In [13]:

```
#5.Illustrate the following property of matrix transposition using these matrices (AC)

#load np
import numpy as np

#matrices A and C
A = np.array([[1, -2, 3, 7], [2, 1, 1, 4], [-3, 2, -2, 10]])
C = np.array([[4, 0, -2, 3], [3, 6, 9, 7], [2, 2, 5, 1], [9, 4, 6, -2]])

#(AC)^T
transpose_of_product = np.transpose(np.dot(A, C))
print("(AC)^T =\n", transpose_of_product)

#C^T A^T
product_of_transposes = np.dot(np.transpose(C), np.transpose(A))
print("\nC^T A^T =\n", product_of_transposes)

#equal?
are_equal = np.array_equal(transpose_of_product, product_of_transposes)
print("\nAre they equal?", are_equal)
```

```
(AC)^T =
 [[ 67  49  80]
 [ 22  24  48]
 [ 37  34  74]
 [-22   6 -17]]

C^T A^T =
 [[ 67  49  80]
 [ 22  24  48]
 [ 37  34  74]
 [-22   6 -17]]

Are they equal? True
```

In [14]:

```
#Fine C-1 using Python code

#load np
import numpy as np

#matrix C
C = np.array([[4, 0, -2, 3], [3, 6, 9, 7], [2, 2, 5, 1], [9, 4, 6, -2]])

#inverse of C
C_inverse = np.linalg.inv(C)
print("C^(-1) =\n", C_inverse)
```

```
C^(-1) =
 [[ 0.14814815 -0.07407407  0.14814815  0.03703704]
 [-0.3         0.35       -1.05        0.25      ]
 [ 0.02962963 -0.11481481  0.52962963 -0.09259259]
 [ 0.15555556  0.02222222  0.15555556 -0.11111111]]
```

In [19]:

```
import cv2
import matplotlib.pyplot as plt

image_bgr = cv2.imread('week7_image.jpg', cv2.IMREAD_COLOR)

image_rgb = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2RGB)


plt.imshow(image_rgb)
plt.axis('off')
plt.show()
```

In [18]:

```
pip install opencv-python
```

```
Collecting opencv-python
  Obtaining dependency information for opencv-python from https://files.pythonhosted.
org/packages/38/d2/3e8c13ffc37ca5ebc6f382b242b44acb43eb489042e1728407ac3904e72f/openc
v_python-4.8.1.78-cp37-abi3-win_amd64.whl.metadata
  Downloading opencv_python-4.8.1.78-cp37-abi3-win_amd64.whl.metadata (20 kB)
Requirement already satisfied: numpy>=1.21.2 in c:\users\lexiw\anaconda3\lib\site-pac
kages (from opencv-python) (1.24.3)
Downloading opencv_python-4.8.1.78-cp37-abi3-win_amd64.whl (38.1 MB)
   ---------------------------------------- 0.0/38.1 MB ? eta -:--:--
   ---------------------------------------- 0.1/38.1 MB 1.1 MB/s eta 0:00:35
    --------------------------------------- 0.7/38.1 MB 7.0 MB/s eta 0:00:06
   - -------------------------------------- 1.8/38.1 MB 14.3 MB/s eta 0:00:03
   --- ------------------------------------ 2.9/38.1 MB 15.6 MB/s eta 0:00:03
   --- ------------------------------------ 3.8/38.1 MB 17.1 MB/s eta 0:00:03
   ----- ---------------------------------- 4.8/38.1 MB 19.2 MB/s eta 0:00:02
   ------ --------------------------------- 6.1/38.1 MB 20.3 MB/s eta 0:00:02
   ------- -------------------------------- 7.3/38.1 MB 21.1 MB/s eta 0:00:02
   -------- ------------------------------- 8.0/38.1 MB 20.5 MB/s eta 0:00:02
   --------- ------------------------------ 9.4/38.1 MB 21.3 MB/s eta 0:00:02
   ----------- ---------------------------- 10.6/38.1 MB 26.2 MB/s eta 0:00:02
   ------------ --------------------------- 11.4/38.1 MB 25.2 MB/s eta 0:00:02
   ------------- -------------------------- 12.6/38.1 MB 26.2 MB/s eta 0:00:01
   -------------- ------------------------- 13.6/38.1 MB 27.3 MB/s eta 0:00:01
   --------------- ------------------------ 14.7/38.1 MB 27.3 MB/s eta 0:00:01
   ---------------- ----------------------- 15.8/38.1 MB 26.2 MB/s eta 0:00:01
   ----------------- ---------------------- 17.0/38.1 MB 27.3 MB/s eta 0:00:01
   ------------------ --------------------- 17.9/38.1 MB 27.3 MB/s eta 0:00:01
   -------------------- ------------------- 19.1/38.1 MB 27.3 MB/s eta 0:00:01
   --------------------- ------------------ 20.4/38.1 MB 27.3 MB/s eta 0:00:01
   ---------------------- ----------------- 21.8/38.1 MB 28.5 MB/s eta 0:00:01
   ------------------------ --------------- 23.3/38.1 MB 27.3 MB/s eta 0:00:01
   ------------------------- -------------- 24.3/38.1 MB 28.5 MB/s eta 0:00:01
   -------------------------- ------------- 25.3/38.1 MB 28.5 MB/s eta 0:00:01
   --------------------------- ------------ 26.4/38.1 MB 28.5 MB/s eta 0:00:01
   ---------------------------- ----------- 27.3/38.1 MB 27.3 MB/s eta 0:00:01
   ----------------------------- ---------- 28.5/38.1 MB 27.3 MB/s eta 0:00:01
   ------------------------------- -------- 29.6/38.1 MB 26.2 MB/s eta 0:00:01
   -------------------------------- ------- 30.5/38.1 MB 25.2 MB/s eta 0:00:01
   --------------------------------- ------ 31.6/38.1 MB 25.2 MB/s eta 0:00:01
   ---------------------------------- ----- 32.4/38.1 MB 24.2 MB/s eta 0:00:01
   ----------------------------------- ---- 33.5/38.1 MB 23.4 MB/s eta 0:00:01
   ----------------------------------- ---- 34.2/38.1 MB 23.4 MB/s eta 0:00:01
   ------------------------------------- -- 35.6/38.1 MB 23.4 MB/s eta 0:00:01
   -------------------------------------- - 36.7/38.1 MB 22.6 MB/s eta 0:00:01
   ---------------------------------------  37.7/38.1 MB 24.2 MB/s eta 0:00:01
   ---------------------------------------  38.1/38.1 MB 24.2 MB/s eta 0:00:01
   ---------------------------------------  38.1/38.1 MB 24.2 MB/s eta 0:00:01
   ---------------------------------------- 38.1/38.1 MB 19.2 MB/s eta 0:00:00
Installing collected packages: opencv-python
Successfully installed opencv-python-4.8.1.78
Note: you may need to restart the kernel to use updated packages.
```

In [21]:

```python
#Find the resolution of this image. Hint: OpenCV imports the image as a Numpy array, w

height, width, _ = image_rgb.shape
print(f"resolution: {width} x {height}")
```

```
resolution: 600 x 453
```

In checking the array dimensions, you should see that three numbers are displayed. What is this third number, and why is it there?

-The third number is the number of channels in the image. It is there to represent the color information for each pixel in the image.