# In[1]:

def count\_pairs\_with\_sum(numbers, target\_sum):

count = 0

n = len(numbers)

for i in range(n):

for j in range(i + 1, n):

if numbers[i] + numbers[j] == target\_sum:

count += 1

return count

numbers = [2, 7, 4, 1, 3, 6]

target\_sum = 10

result = count\_pairs\_with\_sum(numbers, target\_sum)

print(f"Number of pairs with sum {target\_sum}: {result}")

# In[2]:

def calculate\_range(numbers):

if len(numbers) < 3:

return "Range determination not possible"

max\_value = max(numbers)

min\_value = min(numbers)

range\_value = max\_value - min\_value

return range\_value

input\_list = [5, 3, 8, 1, 0, 4]

result = calculate\_range(input\_list)

print(f"The range of the list {input\_list} is: {result}")

# In[ ]:

# In[3]:

import numpy as np

def matrix\_power(A, m):

# Check if A is a square matrix

if A.shape[0] != A.shape[1]:

return "Error: Input must be a square matrix."

# Check if m is a positive integer

if not isinstance(m, int) or m <= 0:

return "Error: m must be a positive integer."

# Initialize the result as the identity matrix of the same size as A

result = np.eye(A.shape[0])

# Multiply the matrix A, m times

for \_ in range(m):

result = np.dot(result, A)

return result

# Example usage

A = np.array([[1, 2], [3, 4]])

m = 3

result = matrix\_power(A, m)

print(result)

# In[4]:

def highest\_occurring\_character(input\_string):

# Create a dictionary to hold character counts

char\_count = {}

# Count occurrences of each alphabetic character

for char in input\_string:

if char.isalpha(): # Consider only alphabetic characters

char = char.lower() # Convert to lowercase for case insensitivity

if char in char\_count:

char\_count[char] += 1

else:

char\_count[char] = 1

# Determine the character with the highest occurrence

if not char\_count:

return "No alphabetic characters found."

max\_char = max(char\_count, key=char\_count.get)

max\_count = char\_count[max\_char]

return max\_char, max\_count

# Example usage

input\_string = "hippopotamus"

result = highest\_occurring\_character(input\_string)

print(f"The maximally occurring character is '{result[0]}' & occurrence count is {result[1]}.")

# In[ ]: