1)

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

seen = set()

pairs\_count = 0

for number in lst:

complement = target\_sum - number

if complement in seen:

pairs\_count += 1

seen.add(number)

return pairs\_count

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

target\_sum = 10

print("Number of pairs with sum equal to", target\_sum, "is:", count\_pairs\_with\_sum(lst, target\_sum))

2) def count\_pairs\_with\_sum(lst, target\_sum):

lst.sort()

left = 0

right = len(lst) - 1

pairs\_count = 0

while left < right:

current\_sum = lst[left] + lst[right]

if current\_sum == target\_sum:

pairs\_count += 1

left += 1

right -= 1

elif current\_sum < target\_sum:

left += 1

else:

right -= 1

return pairs\_count

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

target\_sum = 10

print("Number of pairs with sum equal to", target\_sum, "is:", count\_pairs\_with\_sum(lst, target\_sum))

3) def matrix\_multiply(A, B):

n = len(A)

result = [[0] \* n for \_ in range(n)]

for i in range(n):

for j in range(n):

result[i][j] = sum(A[i][k] \* B[k][j] for k in range(n))

return result

def matrix\_power(A, m):

if m < 1:

raise ValueError("The exponent m must be a positive integer.")

n = len(A)

result = [[1 if i == j else 0 for j in range(n)] for i in range(n)]

base = A

while m > 0:

if m % 2 == 1:

result = matrix\_multiply(result, base)

base = matrix\_multiply(base, base)

m //= 2

return result

def get\_matrix\_input():

try:

n = int(input("Enter the size of the square matrix (n): "))

matrix = []

print("Enter the matrix elements row by row, separated by spaces:")

for i in range(n):

row = list(map(float, input().split()))

if len(row) != n:

raise ValueError(f"Each row must have exactly {n} elements.")

matrix.append(row)

return matrix

except ValueError as e:

print(f"Error: {e}")

return None

def get\_integer\_input(prompt):

while True:

try:

value = int(input(prompt))

if value <= 0:

raise ValueError("The integer must be positive.")

return value

except ValueError as e:

print(f"Error: {e}. Please enter a positive integer.")

print("Matrix Exponentiation Program")

matrix = get\_matrix\_input()

if matrix is not None:

exponent = get\_integer\_input("Enter the positive integer m: ")

try:

result = matrix\_power(matrix, exponent)

print("Matrix A^m is:")

for row in result:

print(" ".join(map(str, row)))

except ValueError as e:

print(f"Error: {e}")

4)

def count\_highest\_occurring\_char(input\_string):

char\_count = {}

for char in input\_string:

if char.isalpha():

char = char.lower()

if char in char\_count:

char\_count[char] += 1

else:

char\_count[char] = 1

if not char\_count:

return None, 0

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

highest\_count = char\_count[highest\_char]

return highest\_char, highest\_count

input\_string = input("Enter a string: ")

highest\_char, highest\_count = count\_highest\_occurring\_char(input\_string)

if highest\_char:

print(f"The highest occurring character is '{highest\_char}' with {highest\_count} occurrences.")

else:

print("No alphabetic characters found in the input.")