NAME-SUTIRTHA SAMANTA

SEC-A ROLL-64

ENROLLMENT NO-12023006015045

SUBJECT- DSDA LAB4 ASSIGNMENT

1. Write a program to enter a string. Calculate the length of the string. Find the substring country. Count the occurences of each word in the given sentence. If the String as input is India is my motherland. I love my country. Capital of India is New Delhi.

input\_string = "India is my motherland. I love my country. Capital of India is New Delhi." length\_of\_string = len(input\_string)

print("Length of the string:", length\_of\_string) substring = "country"

index\_of\_substring = input\_string.find(substring) if index\_of\_substring != -1:

print(f"The substring '{substring}' is found at index:", index\_of\_substring) # Indent this line else:

print(f"The substring '{substring}' is not found in the input string.") # Indent this line as well import re

from collections import Counter

words = re.findall(r'\b\w+\b', input\_string.lower()) word\_count = Counter(words)

print("Word occurrences:", word\_count)

 Length of the string: 73

The substring 'country' is found at index: 34

Word occurrences: Counter({'india': 2, 'is': 2, 'my': 2, 'motherland': 1, 'i': 1, 'love': 1, 'country': 1, 'capital': 1, 'of': 1, 'new':

1. Write a program that accepts a comma separated sequence of words as input and prints the

words in a comma-separated sequence after sorting them alphabetically. Suppose the following input is supplied to the program: without,hello,bag,world Then, the output should be: bag,hello,without,world

Double-click (or enter) to edit

input\_sequence = "without,hello,bag,world" print("Input sequence:", input\_sequence)

words = input\_sequence.split(',') words.sort()

sorted\_sequence = ','.join(words)

print("Sorted sequence:", sorted\_sequence)

 Input sequence: without,hello,bag,world Sorted sequence: bag,hello,without,world

# Write a program that accepts sequence of lines as input and prints the lines after making all

characters in the sentence capitalized. Suppose the following input is supplied to the program: Hello world Practice makes perfect Then, the output should be: HELLO WORLD PRACTICE MAKES PERFECT

# Initialize an empty list to store the input lines lines = []

print("Enter lines of text (enter an empty line to finish):") while True:

line = input() # Indent this line if line:

lines.append(line) # Indent this line else:

break # Indent this line

uppercased\_lines = [line.upper() for line in lines] print("Output:")

for line in uppercased\_lines:

print(line) # Indent this line

 Enter lines of text (enter an empty line to finish): vampire

1. Write a program that accepts a sequence of whitespace separated words as input and prints the words after removing all duplicate words and sorting them alphanumerically. Suppose the following input is supplied to the program: hello world and practice makes perfect and hello world again Then, the output should be: again and hello makes perfect practice world

input\_string = input("Enter a sequence of whitespace-separated words: ") words = input\_string.split()

unique\_words = set(words)

sorted\_words = sorted(unique\_words)

sorted\_sequence = ' '.join(sorted\_words) print("Output:", sorted\_sequence)

 Enter a sequence of whitespace-separated words: I love India Output: I India love

# Write a program that accepts a sentence and calculate the number of letters and digits.

Suppose the following input is supplied to the program: hello world! 123 Then, the output should be: LETTERS 10 DIGITS 3

sentence = input("Enter a sentence: ") letters\_count = 0

digits\_count = 0

for char in sentence:

if char.isalpha(): # Indent this line

letters\_count += 1 # Indent this line elif char.isdigit(): # Indent this line

digits\_count += 1 # Indent this line print(f"LETTERS {letters\_count}")

print(f"DIGITS {digits\_count}")

 Enter a sentence: India is my nation LETTERS 15

DIGITS 0

# Double-click (or enter) to edit

1. Write a program which accepts a string as input to print “Yes” if the string “yes” or ”YES” or “Yes”, otherwise print “No”.

input\_string = input("Enter a string: ") if input\_string in ["yes", "YES", "Yes"]:

print("Yes") # Indented to indicate it's part of the 'if' block else:

print("No") # Indented to indicate it's part of the 'else' block

 Enter a string: I am a Student No

# Write a program which accepts a sequence of words separated by whitespace as input to print

the words composed of digits only. Example: If the following words is given as input to the program: 2 cats and 3 dogs. Then, the output of the program should be: [‘2’ , ‘3’] In case of input data being supplied to the question, it should be assumed to be a console input.

input\_string = input("Enter a sequence of words: ") words = input\_string.split()

digits\_only = [word for word in words if word.isdigit()] print(digits\_only)

 Enter a sequence of words: India []

# Please write a program which count and print the numbers of each character in a string input by console. Example:

If the following string is given as input to the program: abcdefgabc Then, the output of the program should be: a,2 c,2 b,2 e,1 d,1 g,1 f,1

from collections import Counter

input\_string = input("Enter a string: ") char\_count = Counter(input\_string)

for char in sorted(char\_count):

print(f"{char},{char\_count[char]}") # Indented to be part of the 'for' loop

 Enter a string: Sutirtha S,1

a,1

h,1

i,1

r,1

t,2

u,1

 Code  Text

# Write a program that accepts a string

I. 1.reverses it. II. 2.checks whether it is a palindrome. III. 3.checks whether it ends with a specific substring. IV. 4.capitalize the first letter of each word in a string V. 5.check if a string is anagram of another string VI. 6.remove vowels from string VII. 7.find length of the longest word in a sentence

input\_string = input("Enter a string: ") reversed\_string = input\_string[::-1]

print("Reversed string:", reversed\_string)

is\_palindrome = input\_string == reversed\_string

print("Is the string a palindrome?", is\_palindrome)

substring = input("Enter a substring to check if the string ends with it: ") ends\_with\_substring = input\_string.endswith(substring)

print("Does the string end with the substring?", ends\_with\_substring) capitalized\_string = input\_string.title()

print("Capitalized string:", capitalized\_string)

another\_string = input("Enter another string to check for anagram: ") is\_anagram = sorted(input\_string) == sorted(another\_string)

print("Is the string an anagram of the other string?", is\_anagram) vowels = "aeiouAEIOU"

no\_vowels\_string = ''.join(char for char in input\_string if char not in vowels) print("String without vowels:", no\_vowels\_string)

sentence = input("Enter a sentence to find the length of the longest word: ") words = sentence.split()

longest\_word\_length = max(len(word) for word in words)

print("Length of the longest word:", longest\_word\_length)

 Enter a string: Country Reversed string: yrtnuoC

Is the string a palindrome? False

Enter a substring to check if the string ends with it: Cons Does the string end with the substring? False

Capitalized string: Country

Enter another string to check for anagram: Palace

Is the string an anagram of the other string? False String without vowels: Cntry

Enter a sentence to find the length of the longest word: Crown Length of the longest word: 5

# Write Python program to find the sum of all odd numbers in a 2D array.

def sum\_odd\_numbers(arr):

"""

This function takes a 2D array as input and returns the sum of all odd numbers in it. """

total = 0

for row in arr:

for num in row: # Indentation is crucial here! if num % 2 != 0:

total += num return total

# Example usage:

arr = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

result = sum\_odd\_numbers(arr)

print("Sum of odd numbers:", result) # Output: 25 Sum of odd numbers: 25

# Write a Python program to print transpose of matrix.

def transpose\_matrix(matrix):

"""

This function takes a matrix (represented as a list of lists) and prints its transpose. """

transposed = []

for i in range(len(matrix[0])): # Iterate over columns new\_row = []

for row in matrix:

new\_row.append(row[i])

transposed.append(new\_row)

# Print the transposed matrix for row in transposed:

print(row)

# Example usage: matrix = [

[1, 2, 3],

[4, 5, 6],

[7, 8, 9]

]

transpose\_matrix(matrix)

 [1, 4, 7]

[2, 5, 8]

[3, 6, 9]

# Write a Python program to check whether a given matrix is sparse or not.

def is\_sparse(matrix):

"""

This function checks whether a given matrix is sparse or not.

A matrix is considered sparse if the number of zero elements is greater than half of the total elements. """

rows = len(matrix)

cols = len(matrix[0])

total\_elements = rows \* cols zero\_count = 0

for row in matrix:

for element in row:

if element == 0:

zero\_count += 1

return zero\_count > (total\_elements / 2) # Example usage:

matrix1 = [ [1, 0, 0],

[0, 0, 5],

[0, 9, 0]

]

matrix2 = [ [1, 2, 3],

[4, 5, 6],

[7, 8, 9]

]

print("Is matrix1 sparse?", is\_sparse(matrix1)) # Output: True print("Is matrix2 sparse?", is\_sparse(matrix2)) # Output: False

 Is matrix1 sparse? True Is matrix2 sparse? False

# Write a Python program to count the prime numbers in an array.

def count\_primes(arr):

"""

This function takes an array as input and returns the count of prime numbers in it. """

count = 0

for num in arr:

if num <= 1:

continue

is\_prime = True

for i in range(2, int(num\*\*0.5) + 1):

if num % i == 0:

is\_prime = False break

if is\_prime:

count += 1 return count

# Example usage:

arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

prime\_count = count\_primes(arr)

print("Number of prime numbers:", prime\_count) # Output: 4  Number of prime numbers: 4

# Write a Python program to find second highest element of an array.

def find\_second\_highest(arr):

"""

This function finds the second highest element in an array. """

if len(arr) < 2:

return None # Not enough elements for a second highest

highest = second\_highest = float('-inf') for num in arr:

if num > highest:

second\_highest = highest highest = num

elif num > second\_highest and num != highest:

second\_highest = num

return second\_highest if second\_highest != float('-inf') else None # Example usage:

arr1 = [10, 5, 20, 8, 15]

arr2 = [1, 1, 1]

arr3 = [5]

print("Second highest in arr1:", find\_second\_highest(arr1)) # Output: 15

print("Second highest in arr2:", find\_second\_highest(arr2)) # Output: None print("Second highest in arr3:", find\_second\_highest(arr3)) # Output: None

# Write a Python program which counts the non-zero elements in an integer array.

def count\_non\_zero(arr):

"""

This function counts the non-zero elements in an integer array. """

count = 0

for num in arr:

if num != 0:

count += 1 return count

# Example usage:

arr = [1, 0, 2, 0, 3, 0, 0, 4]

non\_zero\_count = count\_non\_zero(arr)

print("Number of non-zero elements:", non\_zero\_count) # Output: 4  Number of non-zero elements: 4

# Write a Python program to merge two float arrays.

import numpy as np

def merge\_float\_arrays(arr1, arr2):

"""

This function merges two float arrays using NumPy and returns the merged array. """

merged\_array = np.concatenate((arr1, arr2)) return merged\_array

# Example usage:

arr1 = np.array([1.2, 3.4, 5.6])

arr2 = np.array([7.8, 9.0])

merged\_array = merge\_float\_arrays(arr1, arr2)

print("Merged array:", merged\_array) # Output: [1.2 3.4 5.6 7.8 9. ]

 Merged array: [1.2 3.4 5.6 7.8 9. ]

# Write a Program program where elements of two integer arrays get added index wise and get stored into a third array.

def add\_arrays\_indexwise(arr1, arr2):

"""

This function adds elements of two integer arrays index-wise and stores the results in a third array. """

result = []

for i in range(min(len(arr1), len(arr2))): # Iterate up to the length of the shorter array result.append(arr1[i] + arr2[i])

return result

# Example usage:

arr1 = [1, 2, 3, 4]

arr2 = [5, 6, 7, 8]

result\_array = add\_arrays\_indexwise(arr1, arr2)

print("Result array:", result\_array) # Output: [6, 8, 10, 12]

 Result array: [6, 8, 10, 12]

# Write a Python program to multiply two matrices.

def multiply\_matrices(matrix1, matrix2):

"""

This function multiplies two matrices and returns the resulting matrix. """

result = [[0 for \_ in range(len(matrix2[0]))] for \_ in range(len(matrix1))] for i in range(len(matrix1)):

for j in range(len(matrix2[0])):

for k in range(len(matrix2)):

result[i][j] += matrix1[i][k] \* matrix2[k][j] return result

# Example usage:

matrix1 = [[1, 2], [3, 4]]

matrix2 = [[5, 6], [7, 8]]

result\_matrix = multiply\_matrices(matrix1, matrix2) # Print the result matrix

for row in result\_matrix:

print(row)

 [19, 22]

[43, 50]

# Write a Python program to subtract two matrices.

def subtract\_matrices(matrix1, matrix2):

"""

This function subtracts two matrices and returns the resulting matrix. """

result = [[0 for \_ in range(len(matrix1[0]))] for \_ in range(len(matrix1))] for i in range(len(matrix1)):

for j in range(len(matrix1[0])):

result[i][j] = matrix1[i][j] - matrix2[i][j] return result

# Example usage:

matrix1 = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

matrix2 = [[9, 8, 7], [6, 5, 4], [3, 2, 1]]

result\_matrix = subtract\_matrices(matrix1, matrix2) # Print the result matrix

for row in result\_matrix: print(row)

 [-8, -6, -4]

[-2, 0, 2]

[4, 6, 8]