SACHIN.R

AIE-E

Functions Used:

1)no\_vow\_conso:

The no\_vow\_conso function counts the total number of vowels and consonants int the string inputted by the user. In this function we have list consisting of all the vowels. We have both its uppercase and lowercase form in the list. We are then traversing the string inputted by the user and then checking the string for vowels. If the vowels are found then increement the count\_vow by one or else increement the count\_conso by one. Then at last we are printing the total number of vowels and consonants.

2)compare\_lists:

The compare\_lists function comapare two lists and then counts the total number of same number in both the lists. We do this using two for loops having the range starting from 0 to the length of the lists L1 and L2 respectively. If the same number is found in both the lists and if the if condition becomes true, then count becomes count variable becomes count +1. Then at last the count variable becomes printed by using the print function.

3)matrix\_multiplication:

The matrix Multiplication function that I have can be used for performing the operation of matrix multiplication without importing the numpy library. First we have created a variable called the “result” in which we have initialized the result variable with the zeroes to ensure that the matrix has no other values. Then we have Three “for” loops in which first for loop iterates equal to the number of rows in the in the first matrix. Second for loop iterates to the number of columns in matrix B. Number of iterations is determined by the number of columns in matrix\_b or, equivalently, the number of rows in matrix\_a and this happens in the third for loop, where the in the third for loop we get the resultant matrix.

4) get\_user\_input\_matrix function():

This function is used for getting the Matrix input without using numpy. In this we have created an empty matrix where in we are appending the row and column elements respectively. Then we are intialzing the rows of second to columns of first matrix such that there is no possibility for a wrong multiplication.

5)Mat\_Transpose:

This function is used for finding the transpose of the matrix that we are giving.We are first initializing the the matrix with zeroes to ensure that no other values already exist in the matrix. Then we are adding two for loops. Through those two for loops we are interchanging the row wise elements to column wise elements and column wise element s to row wise elements respectively. Then we are printing the following elements using a for loop.

Pseudo Code:

1. Function no\_vow\_conso()

Random\_String 🡨 USERINPUT

Vowel🡨[a,e,i,o,u,A,E,I,O,U]

Count\_conso🡨0

Count\_vowel🡨0

FOR i 🡨 random\_String

IF i🡨 vowel

Count\_vowel🡨Count\_vowel+1

ELSE

Count\_conso🡨Count\_conso+1

PRINT “the number of vowels “

PRINT “the number of cosonants”

1. FUNCTION compare\_lists(L1,L2)

Count🡨0

FOR i🡨0 to len(L1)

FOR j🡨 0 to len(L2)

IF L1[i] ==L2[j]

Count🡨Count+1

PRINT “the count of elements”

L1🡨[1,23,45,778,902,678]

L2🡨[2,23,778,902,344,298]

1. FUNCTION matrix\_multiplication(matrix\_a,matrix\_b)

result🡨0🡨FOR \_ 🡨 range(len(matrix\_b[0])) FOR\_ i🡨range(len(matrix\_a))

FOR i🡨len(matrix\_a)

FOR j🡨len(matrix\_b[0])

FOR k🡨len(matrix\_b)

Return matrix

result[i][j]🡨result[i][j]+matrix\_a[i][k]+matrix\_b[k][j]

FUNCTION get\_user\_input(rows,columns)

Matrix🡨[]

FOR i🡨range(rows)

rows🡨[]

FOR j🡨range(columns)

element🡨float(element)

Append element to row

Append row to matrix

Return matrix

row\_first🡨int(row\_matrix)

column\_first🡨int(column\_matrix)

user\_matrix\_first🡨get\_user\_input\_matrix(row\_first,column\_first)

rows\_second🡨column\_first

column\_second🡸int(column\_second)

user\_matrix\_second🡨get\_user\_input\_matrix(row\_second,column\_second)

result\_matrix🡨matrix\_multiplication(user\_first\_matrix,user\_matrix\_second)

PRINT🡸”Result Matrix”

FOR row🡨in result\_matrix

PRINT row

4)FUNCTION Mat\_Transpose(Matrix\_a)

Transpose\_Matrix🡨0 FOR row🡨0 to len(Matrix\_a) FOR column 🡨0 to Matrix\_a[0]

FOR i🡨 0 to len(Matrix\_a)

FOR j🡨0 to len(Matrix\_b)

Transpose\_Matrix[j][i]🡸Tranpose\_Matrix[i][j]  
Matrix\_a🡨[[1,2,3],[4,5,6]]

resultant\_Transpose\_Matrix🡨 Mat\_Transpose(Matrix\_a)

FOR🡨row in resultant\_Transpose\_Matrix

PRINT row