

LAB ASSIGNMENT 1

MACHINE LEARNING

TEAM MEMBERS

NITHIN SAGAR - AIE22136

ADISH VAIBHAV – AIE22102

SUSHANTH – AIE22127

SET 2

QUESTION 1)

Pseudocode and Description

1.The function (vowel) counts the number of vowels (which includes both uppercase and lowercase) in the given input string by the user

- a) Define a function (vowel) which takes an input string as an argument
- b) Initialize a string variable 'vowels' which includes all the vowels (aeiou) both with lowercase and uppercase vowels
- c) Checking and count the number of vowels in the given input string by iteratively going through each character in the string and checking if that character belongs to the 'vowels' declared string
- d) Return the count of vowels

2.The function (consonant) counts the number of consonants (the alphabets that are not vowels) in the given input string by the user

- a) Define a function (consonant) which takes an input string as an argument
- b) Count the number of consonants in the given input string by iteratively going through each character in the string and checking if that character belongs to consonant and not a vowel
- c) Return the count of consonants.

3.In the main function, the user is asked to enter a string of characters. Then, the functions vowel and consonant are called and these returns the count of vowels and consonant

- a) Define the function main
- b) Ask the user to give the input (enter a string of alphabets)
- c) Read the input given by the user
- d) Call the function vowel with the given input string and store the vowels count
- e) Call the function consonant with the given input string and store the consonants count
- f) Print the vowels and consonants count

4.At the end main function is called

QUESTION 2)

Pseudocode and Description

1. Define a function (matrix) to input a matrix by the user

- a. Read the number of rows and columns for the given matrix by the user
- b. Initialize an empty list called mat to store the matrix //usage of the list
- c. Use nested for loops to iterate through each component in the matrix, asking the user to enter each component
- d. Append each row of the matrix to mat(list)
- e. Return resulting matrix mat

2. Define a function (multiply) which takes two matrices (mat_A and mat_B) as arguments and returns their product as a matrix

- a. First check if the number of columns in the matrix (mat_A) is equal to the number of rows in the matrix (mat_B). Return an error message if its not the case
- b. Then initialize an empty result matrix with dimensions as (number of rows in mat_A) x (number of columns in mat_B) //usage of the matrix
- c. Use nested for loop to perform matrix multiplication, updating each component of the result matrix based on the multiplication of corresponding components in matrices (mat_A and mat_B)
- d. Return resulting product matrix

3. Define the main function.

- a. Ask the user to enter the matrix A.
- b. Call the function (matrix) to obtain matrix mat_A
- c. Ask the user to enter the matrix B.
- d. Call the function (matrix) to obtain matrix mat_B
- e. Call the function (multiply) with mat_A and mat_B and store the result (multiplication of corresponding components in each matrix)
- f. If the result is an error message, print the error Or else, print the product matrix

4. Call the main function

QUESTION 3)

Pseudocode and Description

1. Define a function (get_list) to give a input of list of integers from the user
 - a. Ask the user for the total number of elements in the list
 - b. Initialize an empty list called list_input to store the integers
 - c. Use a for loop to iterate through each element in the list, asking the user to enter an integer for each iteration
 - d. Append each integer to list_input
 - e. Return resulting list list_input
2. Define a function (common_elements) that takes the two lists (lista and listb) as arguments and this returns the number of common components between this two lists
 - a. Convert lists (lista and listb) to sets (seta and setb)
 - b. Intersection of seta and setb to get common elements (common_ele)
 - c. Return length of common_ele
3. Define the main function.
 - a. Ask the user to enter the list A
 - b. Call the function (get_list) to get lista
 - c. ask the user to enter the list B
 - d. Call the function (get_list) to get listb
 - e. Call the function (common_elements) with lista and listb and store the result (intersection of both the lists)
 - f. Print the number of common elements between the two lists
4. Call the main function

QUESTION 4)

Pseudocode and Description

1. Define a function (matrix) to input a matrix by the user
 - a. Read the number of rows and columns for the given matrix by the user
 - b. Initialize an empty list called mat to store the matrix //usage of the list
 - c. Use nested for loops to iterate through each component in the matrix, asking the user to enter each component
 - d. Append each row of the matrix to mat(list)
 - e. Return resulting matrix mat
2. Define a function (transpose) that takes a matrix mat as an argument and returns the transpose of the matrix
 - a. Use a list comprehension method to generate the transpose_matrix by swapping rows and columns of the matrix given by the user
 - b. Return the transpose_matrix
3. Define a function (display) that takes a matrix mat and a message as arguments and prints the matrix with the specified message according to the function call
 - a. Print the message accordingly
 - b. Iterate through each row in the matrix and print it iteratively
4. Define the main function.
 - a. Ask the user to enter the matrix A
 - b. Call the function (matrix) to obtain input_matrix
 - c. Call the function (transpose) with input_matrix to get transpose_matrix.
 - d. Using the display function, display the transpose_matrix with the message "Transpose of Matrix A"
5. Call the main function

Assignment 01

NAME : M. SUSHANTH

ROLL NUMBER : A1E22127

ASSIGNMENT REPORT

SET A: QUESTION 01

Pseudo Code

```
def find_pairs(input_list):
    pairs = []

    for i in range(len(input_list)):
        for j in range(i + 1, len(input_list)):
            if input_list[i] + input_list[j] == 10:
                pairs.append((input_list[i], input_list[j]))

        for k in range(j + 1, len(input_list)): # Move this line outside the second loop
            if input_list[i] + input_list[j] + input_list[k] == 10:
                pairs.append((input_list[i], input_list[j], input_list[k]))

    return pairs

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

result_pairs = find_pairs(user_List)

print(f"Q1) The pairs in the list {user_List} that form a sum of 10 are as follows:\n{result_pairs}")
```

code explanation :

The find_pairs function iterates through combinations of elements in the input list, checking for pairs and triplets whose sum is 10. The pairs and triplets are appended to the pairs list. The outer loop iterates over each element, the first inner loop finds pairs, and the second inner loop finds triplets. The final result is printed with the original list and corresponding pairs/triplets.

SET A: QUESTION 02

Pseudo Code:

```
function findingRange(input_list):
    if length of input_list < 3:
        print "ERROR: Range determination not possible."
        return

    min_value = first element of input_list
    max_value = first element of input_list

    for each value in input_list:
        if value < min_value:
            min_value = value
        else if value > max_value:
            max_value = value

    final_Range = max_value - min_value
    print "Q2) The range of the list is final_Range"

user_list = empty list
n = input("Enter the size of the list: ")
print "Enter the values inside the list:"
for a in range n:
    value = input(f"Enter the {a + 1}th element of the list: ")
    append value to user_list

print "Q2) The current list includes elements: user_list"
call findingRange with user_list
```

code explanation

The findingRange function determines the range of a given list by iterating through its elements and finding the minimum and maximum values. The initial check ensures the list has at least three elements for a meaningful range calculation. The final range is calculated as the difference between the maximum and minimum values. The user is prompted to input the size and elements of the list, and the result is printed.

SET A: QUESTION 03

Pseudo Code

```
# Function to determine the range of a list
function findingRange(input_list):
    if length of input_list < 3:
        print "ERROR: Range determination not possible."
        return

    min_value = first element of input_list
    max_value = first element of input_list

    for each value in input_list:
        if value < min_value:
            min_value = value
        else if value > max_value:
            max_value = value

    final_Range = max_value - min_value
    print "Q2) The range of the list is final_Range"

# Main program
user_list = empty list
n = input("Enter the size of the list: ")
print "Enter the values inside the list:"
for a in range n:
    value = input(f"Enter the {a + 1}th element of the list: ")
    append value to user_list

print "Q2) The current list includes elements: user_list"
call findingRange with user_list

# Matrix multiplication and power function
function matrix_mult(A, B):
    result = empty list
    for i in range length of A:
        rows = empty list
        for j in range length of B[0]:
            sum_value = 0
            for k in range length of B:
                sum_value += A[i][k] * B[k][j]
            append sum_value to rows
        append rows to result
    return result

function power_mult(A, m):
    result = copy of A
    for _ in range m - 1:
        result = matrix_mult(result, A)
    return result

function create_matrix(rows, columns):
    matrix = empty list
    print "Enter the values inside the matrix:"
    for _ in range rows:
        row = empty list
        for _ in range columns:
            value = input()
            append value to row
        append row to matrix
    return matrix

function print_matrix(matrix):
    for each row in matrix:
        print row

function Matrix():
    rows = input("Enter the rows of the matrix: ")
    columns = input("Enter the columns of the matrix: ")

    A = create_matrix(rows, columns)

    print "Matrix A:"
    print_matrix(A)

    m = input("Enter the power to be applied to the matrix: ")

    output = power_mult(A, m)

    print "Q3) Resulting matrix after power multiplication:"
    print_matrix(output)

call Matrix
```

code explanation

The findingRange function determines the range of a list by finding the minimum and maximum values, and prints the result. The main program prompts the user to input the size and elements of a list, then calls findingRange to display the range.

The Matrix function performs matrix multiplication with a power function. It prompts the user to input the dimensions and elements of a matrix, the power to apply, and prints the resulting matrix after power multiplication using the power_mult and matrix_mult functions.

SET A: QUESTION 04

Pseudo Code

```
- function MostOccuringLetter(input_word):  
-   highest_occurrence = 0  
-   most_occurred_letter = "  
-  
-   for i from 0 to length of input_word - 1:  
-       count = 1  
-  
-       for j from i + 1 to length of input_word - 1:  
-           if input_word[i] equals input_word[j]:  
-               count += 1  
-  
-           if count > highest_occurrence:  
-               highest_occurrence = count  
-               most_occurred_letter = input_word[i]  
-  
-   return most_occurred_letter, highest_occurrence  
-  
- user_input_word = input("Enter a word of your choice: ")  
-  
- result_letter, result_count = MostOccuringLetter(user_input_word)  
-  
- print "Q4) The most occurring letter is: result_letter"  
- print "Q4) result_letter occurred result_count times"
```

code explanation

The MostOccuringLetter function identifies the most occurring letter in a given word and returns both the letter and its count. The code iterates through each character in the word, counting occurrences by comparing with subsequent characters. The result, including the most occurred letter and its count, is then printed after taking user input.

1.