ASSIGNMENT

By : Nagesh Bhagelli

[1MS23CS118]

#1. Write a Python program to create a 3x3 matrix using a nested list.

matrix\_3x3 = [

    [9, 8, 7],

    [6, 5, 4],

    [3, 2, 1]

]

# Printing the matrix

for row in matrix\_3x3:

    print(row)

#2. How would you calculate the percentage of a student if the total marks are 500 and the student scored 375?

total\_marks = 500

marks\_scored = 375

percentage = (marks\_scored / total\_marks) \* 100

print(f"Percentage of the student is {percentage}%")

#3. Write a Python program that prints the highest and lowest number from a list.

num\_list = [155, 202, 8, 189, 314, 64, 132]

# Using built-in functions

highest = max(num\_list)

lowest = min(num\_list)

print(f"Highest number: {highest}")

print(f"Lowest number: {lowest}")

#Manually finding the maximum and minimum using for loop and if condition

# Assume the first number is both highest and lowest

highest = num\_list[0]

lowest = num\_list[0]

# Loop through the list to find highest and lowest

for num in num\_list:

    if num > highest:

        highest = num

    if num < lowest:

        lowest = num

print(f"Highest number: {highest}")

print(f"Lowest number: {lowest}")

#4. Create a 4x4 identity matrix using NumPy.

import numpy as np

identity\_matrix = np.identity(4)

print(identity\_matrix)

#5.Write a Python function to calculate the percentage for a given marks list and total marks.

#Hint: percentage = (sum of marks / total marks) \* 100

def calculate\_percentage(marks\_list, total\_marks):

    total\_scored = sum(marks\_list)

    percentage = (total\_scored / total\_marks) \* 100

    return percentage

marks = [90, 95, 70, 85, 80]

total = 500  # Total marks possible

result = calculate\_percentage(marks, total)

print(f"Percentage of the student is {result}%")

#6. How do you add two matrices of the same size using NumPy? Provide an example with 2x2 matrices.

import numpy as np

m1 = np.array([[10, 20],[30, 40]])

m2 = np.array([[50, 60], [70, 80]])

# Adding the two matrices

result = m1 + m2

print(result)

#7. Write a Python program to find the highest and lowest number in each matrix (2D list).

import numpy as np

matrix = np.array([

    [33, 81, 12],

    [49, 66, 90],

    [75, 24, 58]

])

highest = np.max(matrix)

lowest = np.min(matrix)

print(f"Highest number in matrix: {highest}")

print(f"Lowest number in matrix: {lowest}")

#8. How would you calculate the average of a given list of numbers?

num\_list = [10, 20, 30, 40, 50]

# Calculate average

average = sum(num\_list) / len(num\_list)

print(f"Average: {average}")

#9. Write a Python function that accepts a matrix and returns the sum of all its elements

def sum\_of\_matrix(matrix):

    total\_sum = sum(sum(row) for row in matrix)

    return total\_sum

matrix = [

    [1, 2, 3],

    [4, 5, 6],

    [7, 8, 9]

]

result = sum\_of\_matrix(matrix)

print(f"The sum of all elements in the matrix is {result}")

#10. Write a program to calculate the difference between the highest and lowest values in each list.

matrix = [

    [3, 8, 1],

    [1, 6, 9],

    [7, 2, 5]

]

# Calculate the difference between the highest and lowest value in each list

difference = [max(row) - min(row) for row in matrix]

# Using a for loop to print the differences

for i in range(len(difference)):

    print(f"Difference between highest and lowest values in list {i + 1}: {difference[i]}")

# 11. Write a Python function that takes a list of numbers and returns the percentage of numbers greater than the average.

def percentage\_greater\_than\_average(numbers):

    average = sum(numbers) / len(numbers)

    greater\_than\_avg = sum(1 for num in numbers if num > average)

    return (greater\_than\_avg / len(numbers)) \* 100

numbers = [10, 20, 30, 40, 50, 60, 70]

result = percentage\_greater\_than\_average(numbers)

print(f"Percentage of numbers greater than the average: {result}%")

# 12. Write a Python function that multiplies two matrices using NumPy.

import numpy as np

def multiply\_matrices(matrix1, matrix2):

    return np.dot(matrix1, matrix2)

matrix1 = np.array([[1, 2], [3, 4]])

matrix2 = np.array([[5, 6], [7, 8]])

result = multiply\_matrices(matrix1, matrix2)

print("Result of matrix multiplication:")

print(result)

# 13. Simulate a high-low game where the player guesses whether the next number will be higher or lower. Track the score.

import random

def high\_low\_game():

    score = 0

    current\_number = random.randint(1, 100)

    while True:

        print(f"Current number: {current\_number}")

        guess = input("Will the next number be higher (h) or lower (l)? ").lower()

        next\_number = random.randint(1, 100)

        print(f"Next number: {next\_number}")

        if (guess == 'h' and next\_number > current\_number) or (guess == 'l' and next\_number < current\_number):

            score += 1

            print("You guessed correctly!")

        else:

            print("You guessed incorrectly.")

            break

        current\_number = next\_number

    print(f"Your final score is: {score}")

# Start the game

high\_low\_game()

# 14. Write a Python program to calculate the percentage increase between two numbers (old and new).

def percentage\_increase(old, new):

    return ((new - old) / old) \* 100

old\_value = 50

new\_value = 75

result = percentage\_increase(old\_value, new\_value)

print(f"The percentage increase is: {result}%")

# 15. Write a Python function that finds the second highest and second lowest number in a list.

def second\_highest\_lowest(numbers):

    unique\_numbers = sorted(set(numbers))

    if len(unique\_numbers) < 2:

        return None, None

    return unique\_numbers[-2], unique\_numbers[1]

numbers = [10, 20, 20, 40, 30]

second\_highest, second\_lowest = second\_highest\_lowest(numbers)

print(f"Second highest: {second\_highest}\nSecond lowest: {second\_lowest}")

# 16. Write Python code to compute the sum of the diagonal elements of a square matrix.

import numpy as np

def sum\_of\_diagonal(matrix):

    return np.trace(matrix)

matrix = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])

result = sum\_of\_diagonal(matrix)

print(f"The sum of the diagonal elements of a square matrix is: {result}")

# 17. Write a Python function to find the most frequent element in a list of numbers.

from collections import Counter

def most\_frequent(numbers):

    count = Counter(numbers)

    return count.most\_common(1)[0][0]

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

result = most\_frequent(numbers)

print(f"The most frequent element is: {result}")

# 18. Create a program to simulate a "high-low" number guessing game. The program generates a random number, and the user must guess whether the next number will be higher or lower.

import random

def high\_low\_game():

    score = 0

    current\_number = random.randint(1, 100)

    while True:

        print(f"Current number: {current\_number}")

        guess = input("Will the next number be higher (h) or lower (l)? ").lower()

        next\_number = random.randint(1, 100)

        print(f"Next number: {next\_number}")

        if (guess == 'h' and next\_number > current\_number) or (guess == 'l' and next\_number < current\_number):

            score += 1

            print("You guessed correctly!")

        else:

            print("You guessed incorrectly.")

            break

        current\_number = next\_number

    print(f"Your final score is: {score}")

# Start the game

high\_low\_game()

# 19. Write a Python function to calculate the cumulative sum of a list of numbers (for percentage calculation).

def cumulative\_sum(numbers):

    result = []

    total = 0

    for num in numbers:

        total += num

        result.append(total)

    return result

numbers = [10, 20, 30, 40]

result = cumulative\_sum(numbers)

print(f"The cumulative sum is: {result}")

# 20. Write a Python program that calculates the percentage of each element in a matrix relative to the total sum of all elements.

import numpy as np

def matrix\_percentages(matrix):

    total\_sum = np.sum(matrix)

    return (matrix / total\_sum) \* 100

matrix = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])

result = matrix\_percentages(matrix)

print("Percentage of each element in the matrix relative to the total sum:")

print(result)

**\*\*\*\*\***