Q1 . Write a Python function that multiplies a matrix by a scalar and returns the result.

def scalar\_multiply(matrix, scalar):

    result = []

    for row in matrix:

        new\_row = []

        for num in row:

            new\_row.append(num \* scalar)

        result.append(new\_row)

    return result

Q2. Write a Python function that takes the dot product of a matrix and a vector. return -1 if the matrix could not be dotted with the vector

def matrix\_dot\_vector(a: list[list[int | float]], b: list[int | float]) -> list[int | float]:

    if len(a[0]) != len(b):

        raise ValueError("Matrix column count must match vector size.")

    result = []

    for i in range(len(a)):

        dot\_product = sum(a[i][j] \* b[j] for j in range(len(b)))

        result.append(dot\_product)

    return result

Q3 . Write a Python function that computes the transpose of a given matrix.

def transpose\_matrix(a):

    result = []

    for j in range(len(a[0])):

        row = []

        for i in range(len(a)):

            row.append(a[i][j])

        result.append(row)

    return result

Q4. Write a Python function that reshapes a given matrix into a specified shape. if it cant be reshaped return back an empty list **[ ]**

import numpy as np

def reshape\_matrix(a: list[list[int | float]], new\_shape: tuple[int, int]) -> list[list[int | float]]:

    flat = [num for row in a for num in row]

    if len(flat) != new\_shape[0] \* new\_shape[1]:

        return []

    return np.array(flat).reshape(new\_shape).tolist()

Q5 . Write a Python function that calculates the mean of a matrix either by row or by column, based on a given mode. The function should take a matrix (list of lists) and a mode ('row' or 'column') as input and return a list of means according to the specified mode.

def calculate\_matrix\_mean(matrix: list[list[float]], mode: str) -> list[float]:

    if mode == 'row':

        return [sum(row) / len(row) for row in matrix]

    elif mode == 'column':

        return [sum(matrix[i][j] for i in range(len(matrix))) / len(matrix) for j in range(len(matrix[0]))]

    else:

        return []