Assignment: AWS Lambda Layer for Matrix Transposition

- **Overview**

  This assignment focuses on the creation of a Python function capable of transposing a 2-dimensional matrix of numbers, packaging this function as a reusable AWS Lambda Layer, and then utilizing this layer in a separate AWS Lambda function. This task will demonstrate the reusability of code in cloud environments, specifically within AWS, and introduce the concept of Lambda Layers as a means to share code, libraries, and other dependencies across multiple Lambda functions.

- **Objectives**
  - Develop a Python function to transpose a 2D matrix.
  - Package this function into an AWS Lambda Layer for reuse.
  - Create a new AWS Lambda function that utilizes this layer to perform matrix transposition.

- **Requirements**
  - Basic knowledge of Python and working with lists.
  - AWS account with access to AWS Lambda and IAM.
  - Familiarity with AWS Lambda, including creating functions and layers.
  - Understanding of how to package Python code for AWS Lambda.

- **Part 1: Developing the Transpose Function**

- Task 1.1: Implement the Transpose Function
  - Instructions:
    - Write a Python function named `transpose_matrix` that accepts a 2D list (matrix) of numbers and returns its transpose.
    - The transpose of a matrix is achieved by flipping a matrix over its diagonal, switching the row and column indices of the matrix.
    - Test your function locally with various matrices to ensure accuracy.

- Task 1.2: Prepare the Function for AWS Lambda Layer
  - Instructions:
    - Create a Python file named `matrix_utils.py` and include your `transpose_matrix` function within this file.
    - Ensure that all dependencies, if any, are included or noted for later setup in the AWS environment.

- **Part 2: Creating and Deploying a Lambda Layer**

- Task 2.1: Package the Lambda Layer

  - Instructions:
    - Prepare your `matrix_utils.py` for packaging into a ZIP file. This file should be placed in a Python directory structure according to the runtime you plan to use, e.g., `python/lib/python3.8/site-packages/`.

- Zip the folder structure, ensuring the path to your `matrix_utils.py` mimics Python's package structure.

- Task 2.2: Create the Lambda Layer
    - Instructions:
        - In the AWS Management Console, navigate to Lambda and choose "Layers" from the sidebar.
        - Click "Create layer," name your layer (e.g., `matrix-transpose-layer`), and upload the ZIP file you prepared.
        - Select the compatible runtime for your Lambda function (matching the Python version used).
        - Note the ARN of the created layer upon completion.

- **Part 3: Utilizing the Lambda Layer in a Function**

- Task 3.1: Create a New Lambda Function
    - Instructions:
        - Create a new Lambda function (e.g., `TransposeMatrixFunction`) from the AWS Management Console.
        - Choose the same Python runtime as used for the Lambda Layer.
        - In the function code, import `matrix_utils` and write a handler that invokes `transpose_matrix` with a matrix provided in the event data.

- Task 3.2: Add the Lambda Layer to Your Function
    - Instructions:
        - In the Lambda function configuration, under "Layers," choose "Add a layer."
        - Select "Custom layers" and choose the layer you created previously.
        - Add the layer to your function.

- Task 3.3: Test Your Lambda Function
    - Instructions:
        - Configure a test event in the Lambda console with a sample 2D matrix as the input.
        - Invoke your Lambda function with the test event and verify that the returned value is the transpose of the input matrix.
        - Test with various sizes and content in the matrices to ensure robustness.

- **Deliverables**
    - The `matrix_utils.py` file contains the `transpose_matrix` function.
    - Documentation of the process and commands used to create and deploy the Lambda Layer.

- The code for the AWS Lambda function that utilizes the layer to transpose matrices.
- Screenshots or logs demonstrating successful tests of the Lambda function with different matrices.