```
% Taha Akhlaq MATLAB Assignment 4: Getting Funcy

clc; % clear command window



% Question 1
calculateDotProduct = @(vectorOne, vectorTwo) vectorOne' * vectorTwo;


% Question 2
function [isOrthonormal] = checkIfMatrixIsOrthonormal(matrixToCheck)
    calculateDotProduct = @(vectorOne, vectorTwo) vectorOne' * vectorTwo;

    numericalPrecisionThreshold = 1000 * eps;

    % Check if each column vector has a norm of 1
    for columnIndex = 1:size(matrixToCheck, 2)
        columnVector = matrixToCheck(:, columnIndex);
        columnNorm = norm(columnVector);
        if abs(columnNorm - 1) > numericalPrecisionThreshold
            isOrthonormal = false;
            return;
        end
    end

    % Check if all column vectors are orthogonal to each other
    for firstColumnIndex = 1:size(matrixToCheck, 2)
        for secondColumnIndex = firstColumnIndex + 1:size(matrixToCheck, 2)
            dotProductValue = calculateDotProduct(matrixToCheck(:,
firstColumnIndex), matrixToCheck(:, secondColumnIndex));
            if abs(dotProductValue) > numericalPrecisionThreshold
                isOrthonormal = false;
                return;
            end
        end
    end

    % If all checks passed, the matrix is orthonormal
    isOrthonormal = true;
end


% Question 3

function orthonormalizedMatrix =
performGramSchmidtOrthonormalization(inputMatrix)

    calculateDotProduct = @(vectorOne, vectorTwo) vectorOne' * vectorTwo;

    % Check if the input matrix is already orthonormal
    if checkIfMatrixIsOrthonormal(inputMatrix)
```

```
        orthonormalizedMatrix = inputMatrix;
        return;
    end

    % Store the orthonormalized vectors
    numberOfRows = size(inputMatrix, 1);
    numberOfColumns = size(inputMatrix, 2);
    orthonormalizedMatrix = zeros(numberOfRows, numberOfColumns);

    % Apply Gram-Schmidt
    for currentColumnIndex = 1:numberOfColumns
        % Take the current column vector from the input matrix
        modifiedVector = inputMatrix(:, currentColumnIndex);

        % Subtract the projections onto the computed orthonormal vectors
        for previousColumnIndex = 1:currentColumnIndex - 1
            projectionCoefficient =
calculateDotProduct(orthonormalizedMatrix(:, previousColumnIndex),
modifiedVector);
            projectionVector = projectionCoefficient *
orthonormalizedMatrix(:, previousColumnIndex);
            modifiedVector = modifiedVector - projectionVector;
        end

        % Normalize the resulting vector
        orthonormalizedMatrix(:, currentColumnIndex) = modifiedVector /
norm(modifiedVector);
    end
end


% Question 4

% Generate a 4x4 matrix with random complex numbers
randomComplexMatrix = randi(15, 4, 4) + 1j * randi(15, 4, 4);

% Display the original randomly generated matrix
disp('Original Random Complex Matrix:');
disp(randomComplexMatrix);

% Step 2: Apply Gram-Schmidt Orthonormalization
orthonormalizedOutputMatrix =
performGramSchmidtOrthonormalization(randomComplexMatrix);

% Display the resulting orthonormalized matrix
disp('Orthonormalized Matrix After Gram-Schmidt Process:');
disp(orthonormalizedOutputMatrix);

% Verify if the output matrix is truly orthonormal
isOrthonormalResult =
checkIfMatrixIsOrthonormal(orthonormalizedOutputMatrix);
disp('Is the matrix orthonormal? (1 = Yes, 0 = No)');
disp(isOrthonormalResult);
```

```
% Output:

Original Random Complex Matrix:
   2.0000 + 1.0000i   3.0000 + 8.0000i  13.0000 + 8.0000i   9.0000 +12.0000i
   3.0000 +13.0000i   8.0000 + 8.0000i   9.0000 +14.0000i  13.0000 + 9.0000i
  11.0000 +10.0000i   3.0000 +13.0000i  14.0000 + 9.0000i  14.0000 + 4.0000i
   8.0000 +15.0000i   1.0000 + 4.0000i  11.0000 +13.0000i  15.0000 +10.0000i

Orthonormalized Matrix After Gram-Schmidt Process:
   0.0760 + 0.0380i   0.1463 + 0.5288i   0.7217 + 0.3956i  -0.0307 + 0.1160i
   0.1140 + 0.4938i   0.5497 + 0.0530i  -0.2328 + 0.2461i   0.3003 - 0.4829i
   0.4179 + 0.3799i  -0.1461 + 0.4807i  -0.1803 - 0.4009i  -0.4832 - 0.0443i
   0.3039 + 0.5698i  -0.1411 - 0.3487i   0.1095 + 0.0518i   0.2680 + 0.5958i

Is the matrix orthonormal? (1 = Yes, 0 = No)
   1
```

Published with MATLAB® R2024b