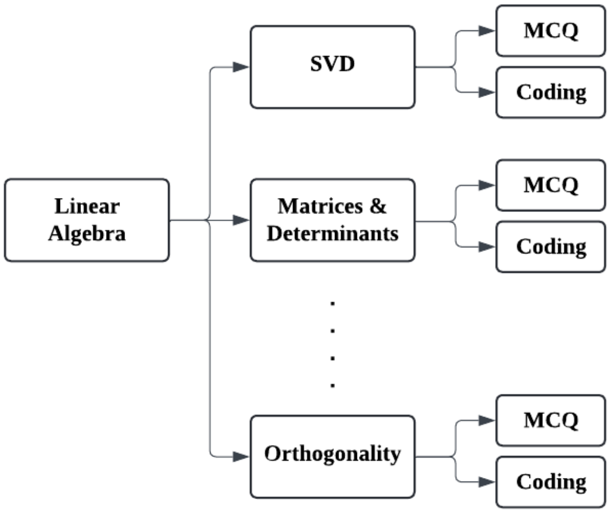
Instructions for Faculty Associates regarding Data preparation

- A google drive link will be shared to the Faculty Associates which contains a folder named '**Linear Algebra**'.

- Inside this folder, different folders will be created for each subtopic under Linear Algebra (such as SVD, Matrices & Determinants etc).

- *FAs are requested to create questions based on both MCQs and Coding under each subtopic*.

- Each subtopic folder is organized into two other folders named as MCQ & Coding as shown in the picture below.



Sample folder structure of the database

Instructions for MCQ based questions:

Inside the MCQ folder, there will be two things:

1) A Google spreadsheet named in the format *{Subtopic}_MCQ*
2) A folder named *Images*

The structure of the spreadsheet inside the MCQ folder is as follows,

Note: Naming formats are CASE-SENSITIVE, So, FAs are requested to strictly follow the naming conventions.

Question Number – Enter a unique number for each question.

Faculty ID – A unique ID representing each Faculty Associate will be provided which has to be used in naming files while uploading.

Question Text – Enter the question description in proper **LATEX** format that includes both the text & equations/special symbols (if any).

Image link in any question – If the question includes an image, upload it to the **'Question Images'** folder within the 'Images' folder. Then, paste the link to the image in the corresponding cell of the spreadsheet. If there is no image, enter **'NA'** in the cell.

Naming Format for Image while uploading – {Faculty ID}_Q{Question Number}

Difficulty tag – Select the difficulty level of the questions (Easy / Medium / Hard).

Question type – Select whether the question is single or multiple-choice.

Options – This field will contain 4 other columns each representing four different options for each question. Enter the options strictly in **LATEX** format with proper syntax for equations, special characters. If the options are in image format, upload the images into the folder named '**Option Images**' inside the Images folder. Then paste the link of each image to the corresponding option cells of the spreadsheet.

Naming Format for Image – *{Faculty ID}_{Question Number}_O{Option Number}*

Correct Answers – Enter the correct option out of the 4 available options.

Format for single choice MCQs – '**1**' if option-1 is correct/ '**2**' if option-2 is correct/ '**3**' if option-3 is correct/ '**4**' if option-4 is correct .

Format for multi-choice MCQs – Enter the numbers corresponding to the correct options separated by comma (1,2 if options 1 and 2 are correct).

Instructions for Coding based questions:

- For each coding question, FAs are required to create two text files.

- One containing the entire code of the respective question with proper syntax and indentations. This file is termed as Reference solution.

- Another one containing the template of the code that will be given to the student during the test. It is referred to as Student solution.

Inside the Coding folder, there will be three things:

1) A Google spreadsheet named in the format *{Subtopic}_Coding*
2) A folder named *Code Files*
3) Another folder named *Images*

The structure of the spreadsheet inside the Coding folder is as follows,

Note: Naming formats are CASE-SENSITIVE, So, FAs are requested to strictly follow the naming conventions.

Question Number – Enter a unique number for each question.

Faculty ID – A unique ID representing each Faculty Associate will be provided which will be used in naming files while uploading.

Question Text – Enter the question description in proper **LATEX** format that includes both the text & equations/special symbols (if any).

Image link in any question – If the question contains an image, you must upload the image into the folder named as **'Images'** folder. Then paste the link of that image into the corresponding cell in the spreadsheet. If there is no image, enter **'NA'** in the cell.

Naming Format for Image while uploading – *{Faculty ID}_Q{Question Number}*

Difficulty tag – Select the difficulty level of the questions (Easy / Medium / Hard).

Language – Select the coding language of the question (MATLAB / Python).

Question type – Select the type of coding question (Full function / Fill in the blank).

Student Solution – This field will contain the Google drive link of text file of the Student solution corresponding to each question. It must be uploaded to folder named '**Student Template'** inside '**Code Files**' folder.

Naming Format for Text file – *{Faculty ID}_S_Q{Question Number}*

Reference Solution – This field will contain the Google drive link of text file of the Reference solution corresponding to each question. It must be uploaded to folder named '**Reference Template'** inside '**Code Files**' folder.

Naming Format for Text file – *{Faculty ID}_R_Q{Question Number}*

Test Cases – Enter the test case for each question. The minimum number of tests cases are 1 and can go till 5.

Format for filling Test Cases – Enter the arguments that must be passed to the function inside the parenthesis ( ). If there are multiple arguments that must be passed, separate them by comma inside the parenthesis.

Example of a “Fill in the Blank” Coding question:

Consider the question, *“Complete the following function that computes the determinant of a square matrix without using any built-in MATLAB functions for determinant calculation.”*

For the above question, the FAs should create two templates as mentioned before.

Reference Solution:

This text file should contain the entire code corresponding to the question which will be used to validate student’s response.

```matlab
function d = my_determinant(A)
    % Get the size of the input matrix
    [n, ~] = size(A);

    % Base case for 1x1 matrix
    if n == 1
        d = A(1,1);
        return;
    end

    % Base case for 2x2 matrix
    if n == 2
        d = A(1,1) * A(2,2) - A(1,2) * A(2,1);
        return;
    end

    % Initialize determinant
    d = 0;


    for j = 1:n
        % Get the minor of A
        minor = A(2:end, [1:j-1, j+1:end]);
        minor_det = my_determinant(minor);
        % Add the cofactor to the determinant
        d = d + ((-1)^(1+j)) * A(1,j) * minor_det;
    end
end
```

The above code should be saved in .txt extension and should be uploaded to the corresponding folder in the google drive.

If your Faculty ID is 21AIE001 and the question number is 5, then the naming format for the reference solution file will be “***21AIE001_R_Q5.txt***”.

Student Solution:

This text file should contain code template with some missing parts which has to be filled by student in the exam.

Wherever you are adding missing parts in the code, at that point you need to include “**%E8**” near it as shown below.

```
function d = my_determinant(A)
    % Get the size of the input matrix
    [n, ~] = %E8

    % Base case for 1x1 matrix
    if n == 1
        d = A(1,1);
        return;
    end

    % Base case for 2x2 matrix
    if n == 2
        d = %E8
        return;
    end

    % Initialize determinant
    d = 0;


    for j = 1:n
        % Get the minor of A
        minor = A(2:end, [1:j-1, j+1:end]);
        minor_det = my_determinant(minor);
        % Add the cofactor to the determinant
        d = d + %E8
    end
end
```

If your Faculty ID is 21AIE001 and the question number is 5, then the naming format for the Student solution file will be “***21AIE001_S_Q5.txt***”.

Test cases format will be – ([1,2; 3,4]) or ([1,2,4; 5,2,1; 3,7,9])

Example of a “Full function” Coding question:

Consider the question, *“Write a function that computes the Greatest Common Divisor (GCD) of two integers without using any built-in MATLAB functions for GCD calculation.”*

For the above question, the FAs should create two templates as mentioned before.

Reference Solution:

```
function d = my_gcd(a, b)
    % Ensure a and b are positive
    a = abs(a);
    b = abs(b);

    % Euclidean algorithm to find GCD
    while b ~= 0
        temp = b;
        b = mod(a, b);
        a = temp;
    end
    d = a;
end
```

Student Solution:

```
function d = my_gcd(a, b)
      %E8
end
```

Follow the above-mentioned named formats.

Test cases format – (6,2) / (4,5)

Please try to avoid keeping blank spaces before and after the comma while entering multiple arguments in test cases.