

Viva2 Question

1. The use of any Large Language Models (LLMs) such as ChatGPT, Bard, Claude, etc., is strictly prohibited.
2. Participation in question-answer communities is not permitted, including but not limited to StackOverflow, Quora, etc.
3. The utilization of any JAR files is expressly forbidden.
4. Only official documentation is permitted for use.

Be advised, any breach of these rules will be deemed as academic misconduct, resulting in the entire group receiving a zero mark for Viva2.

Question 1

1. Modify your `addition`, `subtraction`, and `multiplication` methods to accept 2-dimensional arrays of sizes 2X2, 3X3, 4X4, 5X5, 6X6, and 7X7, performing the required matrix operations. Hardcoding is strictly prohibited. For example:

```
public static double[][] addition(double[][] a, double[][] b) {
    if (a.length == 2 && b.length == 2) {
        double[][] c = new double[2][2];

        // Hardcoding 1
        c[0][0] = c[0][0] + b[0][0]
        c[0][1] = c[0][1] + b[0][1]
        c[1][0] = c[1][0] + b[1][0]
        c[1][1] = c[1][1] + b[1][1]

        // Hardcoding 2
        for (int i = 0; i < 2; i++) {
            for (int j = 0; j < 2; j++) {
                c[i][j] = a[i][j] + b[i][j];
            }
        }
        // ...
    }
}
```

Marks will be deducted for hardcoding. Use `for-loop` instead.

2. Ensure that `addition`, `subtraction`, and `multiplication` accept two `double[][]` 2-dimensional arrays and return a `double[][]`. The use of `int[][]` is not allowed. It should return `null` if both arrays have different dimensions.
3. For `determinants`, it should accept a 3X3 matrix array, `double[][][]`, and return an array of determinants, `double[]`.

4. For `inverse`, it should handle 2-dimensional matrices of sizes 2X2, 3X3, 4X4, 5X5, 6X6 and 7X7, calculating the inverse. It should take a parameter `double[][]` and output a `double[][]`. Return `null` if the matrix has no inverse. Hardcoding is not allowed.

Question 2

Create a new method named `validPalindrome(String string)`. This method should check if the given string is a valid palindrome. A phrase is considered a valid palindrome if, after converting all uppercase letters to lowercase letters and removing all non-alphanumeric characters, it reads the same forward and backward. Alphanumeric characters include letters and numbers. You are not allowed to use the previously mentioned `removeSpecialCharacters()` and `isPalindrome()` methods. Also, you are not allowed to use `.replaceAll()` or `replace()`, that is, methods related to the concept of regex. Additionally, `StringBuilder()` is not allowed here. You might consider using the `Character` class for implementing `validPalindrome()`. The `Character` class contains

- `Character.isLowerCase()` to check if a given character is lowercase.
- `Character.isLetterOrDigit()` to check if a given character is either a letter or a digit.

Question 3

1. Modify your code to handle negative values as both lower and upper limits.
2. Ensure the lower limit is smaller than the upper limit; otherwise, return `null`.
3. If both the lower and upper limits are smaller than 2, return `null`.
4. If the lower limit is negative while the upper limit is a positive value greater than or equal to 2, the code should handle it appropriately.