**Reduced Row Echelon Form (ייצוג מטריצה מדורגת מצומצמת)**

**Instructions:**

-This method returns a matrix that is the reduced row echelon form of the original matrix. Note that the original matrix must remain unchanged.

-To find the reduced row echelon form of a matrix, we do Gaussian elimination to get a matrix with the following properties:

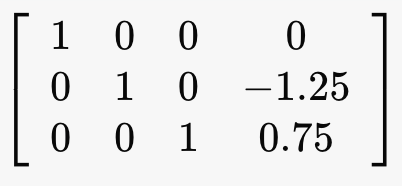
**1.** For each row (that is non-zero), the leading non-zero digit is 1.

**2.** For each leading digit 1, all cells of the matrix in the same column are zeroes.

**3.** The leading digit 1 of every non-zero row is to the right of the leading digit 1 of every row above it.

**4.** All rows of only zeroes are found at the bottom of the matrix.

-**For example:**

 becomes 

(Original Matrix) (Reduced Row Echelon Form matrix)

-For further reading and testing see the following links:

**RREF Matrix calculator:**

<https://www.emathhelp.net/calculators/linear-algebra/reduced-row-echelon-form-rref-calculator/?i=%5B%5B2%2C4%2C8%2C1%5D%2C%5B16%2C3%2C5%2C0%5D%2C%5B1%2C0%2C4%2C3%5D%5D&reduced=on>

**Wikipedia Reference:**

<https://en.wikipedia.org/wiki/Row_echelon_form#Reduced_row_echelon_form>

**Tip:**

-split up each of the Gaussian Elimination operations (row switch, multiplication by scalar, row subtraction, etc.) each into their own functions. This will make it much easier to debug.

**Notes:**

1. The matrices returned may have “negative zeroes” instead of regular zeroes (-0 instead of 0) in some entries. In C/C++, zero and negative zero are equivalent, so you can ignore this. This has to do with the way floating point numbers are represented under the hood. For further reading check this out: <https://en.wikipedia.org/wiki/Signed_zero>
2. These functions should be implemented in one of the .cpp files in your project, not in the header files.