When you generate a random \( m \times n \) matrix, it would have rank as big as possible, which means the minimum of \( m \) and \( n \). Try this by using \( \text{rank}(\text{rand}(4, 7)) \), \( \text{rank}(\text{rand}(8, 4)) \) etc.

You can generate \( m \times n \) matrices with smaller rank by the following method. Assume you would like to get an \( 4 \times 7 \) matrix of rank 2. Then use \( A = \text{rand}(4, 2) \ast \text{rand}(2, 7) \). Try it and check that indeed rank equals 2.

Problem 1. a) Generate random \( 6 \times 8 \), \( 7 \times 5 \), \( 10 \times 10 \) matrices (do not print them!) and check that their ranks are as expected.
b) Generate a random \( 6 \times 8 \) matrix of rank 3. Check that its rank is 3.
Ask matlab for a basis for its nullspace. The matlab command \( \text{null}(A) \) produces a matrix whose columns form a basis for the nullspace of \( A \).
Use Matlab commands and the theory to check that you indeed got a basis for the nullspace of \( A \).
First use matrix multiplication to check whether vectors you got are in the nullspace. Then check that they are linearly independent. Then refer to a theorem from the textbook which implies that these vectors span the nullspace.

Problem 2. Solve the following linear systems with complex coefficients by using two methods \( \text{rref}([A \ b]) \) and \( x = A\backslash b \) and explain your answers.

\[
(1 + i)x_1 + (2 - i)x_2 + 3ix_3 = 7 - 5i \\
2x_1 + (1 - i)x_3 = 4i \\
x_1 + 4ix_2 + (1 + 3i)x_3 = -5 + 7i
\]

and

\[
(1 + i)x_1 + (-3i)x_2 + x_3 = 5 - 4i \\
(3 - i)x_1 - (6 + i)x_2 + 3x_3 = 1 + 2i
\]