1. Let $A \in \mathbb{R}^{n \times n}$ be any tri-diagonal matrix with non-zero entries on the tri-diagonal. Prove that rank of A is not less than $n-1$.
2. Given $Ax = b$, prove that if y satisfies $Ay = b + \delta b$, then
$$\frac{\|y-x\|_2}{\|x\|_2} \leq \frac{\sigma_1}{\sigma_n} \frac{\|\delta b\|_2}{\|b\|_2}$$
where σ_1 is the largest eigenvalue and σ_n is the smallest eigenvalue.
3. Consider solving the 2×2 linear system using Cramer's rule:
$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$
Cramer's rule:
   - Compute $d = a_{11}a_{22} - a_{12}a_{21}$, $d_1 = b_1 a_{22} - b_2 a_{12}$ and $d_2 = a_{11}b_2 - a_{21}b_1$.
   - Obtain $x_1 = \frac{d_1}{d}$ and $x_2 = \frac{d_2}{d}$.

   Show by means of an example that Cramer's rule is not backward stable.

   Hint: To construct such an example, you need to input a matrix A and b and obtain x using Cramer's rule on finite precision machine (with machine precision say $\mu = 10^{-16}$). Once you obtain x, compute the residual $r = Ax - b$ and show that this is large even if $\mu \to 0$.
4. Let $B \in \mathbb{R}^{n \times n}$ be an upper bidiagonal matrix, i.e., B_{ij} is non-zero only along the diagonal and along the first super-diagonal. Derive an algorithm to compute $\kappa_\infty(B) = \|B\|_\infty \left\|B^{-1}\right\|_\infty$ *exactly*, i.e., the algorithm should not involve any iterations. Further, the computational cost of the algorithm should be $\mathcal{O}(n)$.
5. Let A be a nonsingular symmetric matrix and has the factorization $A = LDM^T$, where L, M are lower triangular matrices and D is a diagonal matrix. Prove that $L = M$.
6. Prove that if λ is an eigenvalue of $A + \rho uu^T$, then $(A - \lambda I)^{-1} u$ is its eigenvector.
7. Let $A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ and $b = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$. Begining with the vector $x = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ perform the conjugate gradient algorithm to obtain the solution. Clearly, highlight the iterates x_k's, residuals r_k's and the descent directions p_k's.
8. It is often of interest to solve *constrained least squares problems*, where the solution x must satisfy a linear constraint, say $Cx = d$, in addition to minimizing $\|Ax - b\|_2$, where $A \in \mathbb{R}^{m \times n}$, $C \in \mathbb{R}^{p \times n}$ and full-rank. We further assume that $p \leq n$ (guarantees a solution exists) and $n \leq m + p$ (so that the system is not underdetermined). Devise an appropriate algorithm to obtain x. Of course, as with all algorithms, it should be accurate (well-conditioned), stable and optimal complexity.