Learning Opportunity 1-4

1-4-1 This exercise is very similar to LO 1-2-2. Read through all the steps before starting.

1. Return to LO 1-2-2 and consider the linear system described by the vector equation $Ax = b$, where A is an 8×8 coefficient matrix.
2. Build a QR decomposition of A.
3. Solve the system by using the inverse of a unitary matrix and backsolving.
4. Do the above *twice*, first approximately using `RDF`, then again exactly using `QQbar`. The employed commands will likely be identical. Comment on the difference between these two computations.

Full marks for two versions of the same solution, along with excellent commentary on how they are different.

1-4-2 This exercise will solidify your understanding of how to compute a QR decomposition from Householder reflectors. Read, study and understand the subsection of SCLA which informally describes an algorithm for obtaining a full QR decomposition with a sequence of Householder reflections. ("QR Decomposition via Householder Reflections")

1. Build a 6×6 matrix of random integers using `M = random_matrix(QQ, 6, algorithm="unimodular", upper_bound=9)` which will make a matrix with determinant 1, and hence will be nonsingular. Convert your matrix to `QQbar` with `A = M.change_ring(QQbar)`.
2. Now build all of the Householder reflections necessary to create R, there will be five of them.
3. Form the product of your Householder matrices in the correct order, and as efficiently as possible, to create the Q matrix of the decomposition.
4. You should have the R matrix as a by-product of building the Householder matrices. Verify that Q is unitary and that the decomposition does the job it should do.Hints: Be sure all your vectors and matrices are over `QQbar` and stay that way (and do not become symbolic matrices with `sqrt()` in them). The Sage matrix method `.change_ring()` can help with this, as well as providing `QQbar` to various constructors. You can use the `house()` function from previous worksheets. The Sage matrix method `.column()` could be useful, and a Python slice like `v[2:6]` could also be handy. Note that this exercise does not suggest building a totally general function to create a QR decomposition of any old matrix. Just find the correct five Householder matrices, one by one, and do the right things with them.

   Challenge: build a random 6×6 matrix of rank 4 and repeat the above procedure to see what happens with a singular matrix. Use `M = random_matrix(QQ, 6, algorithm="echelonizable", rank=4, upper_bou`

Full marks for building Q from the correct Householder matrices along with the relevant checks.