# Matrix Inversion

You can invert a matrix by creating the matrix . This means you stack right next to the identity matrix. Then perform Gaussian Elimination on just like you were solving a system of equations. You will end up with the matrix . Sounds easy. But matrix inversion is highly unstable numerically. You will investigate in this lab.

1. The Hilbert matrix is notoriously unstable. (Read about it online; note that our definition uses “+1” because python and math are different sometimes.)
2. Implement matrix inversion and check it on some small Hilbert matrices. (Check by verifying . This will *not* be exact so account for rounding errors).
3. Make a plot of matrix size vs. error. Error is defined as the Frobenius norm of the matrix so . (It’s trivial to do in numpy so look it up!)
4. Implement GEPP (Gaussian elimination with partial pivoting) and make a similar plot to part 2. You should see the errors decrease with GEPP.
5. Finally, plot the error in the inherent np.linalg.inv function and compare it to yours.

You can seen an example of GEPP [here](https://web.mit.edu/10.001/Web/Course_Notes/GaussElimPivoting.html). Note they do NOT make the diagonal all 1’s like we do – but you can keep doing it our way. The main idea is the pivoting.