**Linear Algebra Hackathon 1**

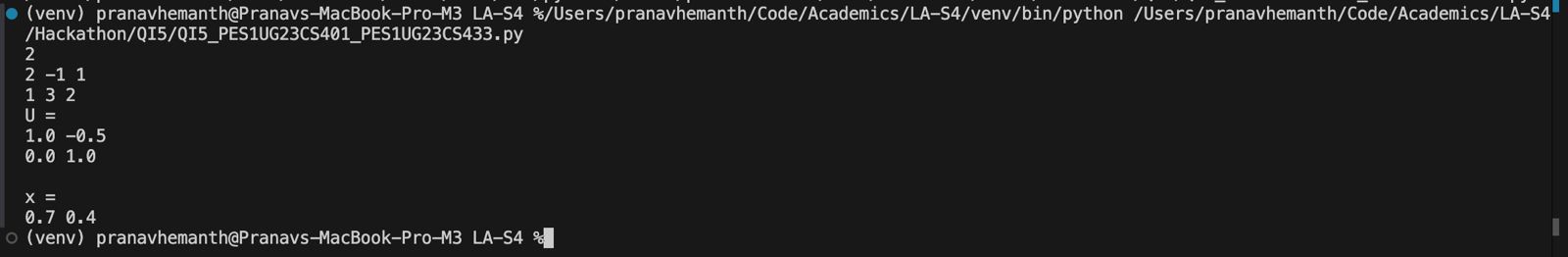
**PES University**

Name: Nishant Holla and Pranav Hemanth

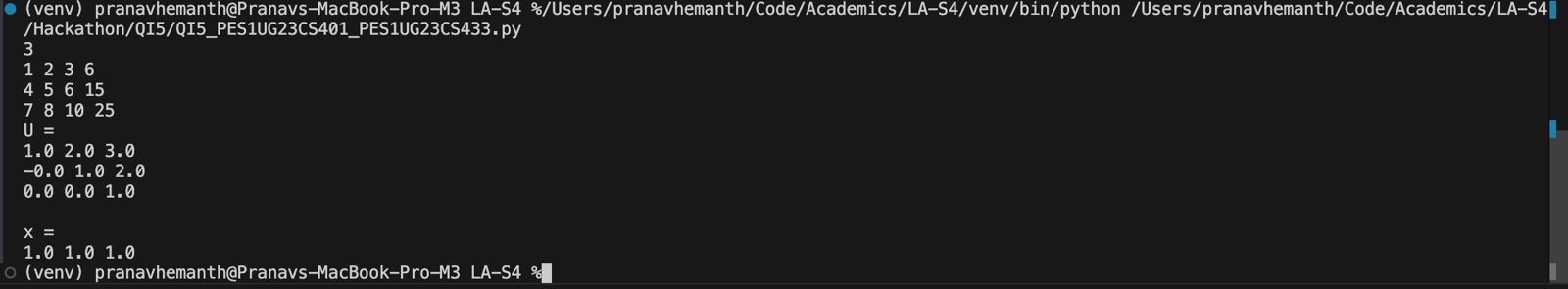
SRN: PES1UG23CS401 and PES1UG23CS433

**Implementation Question 5**

Test Case 1:



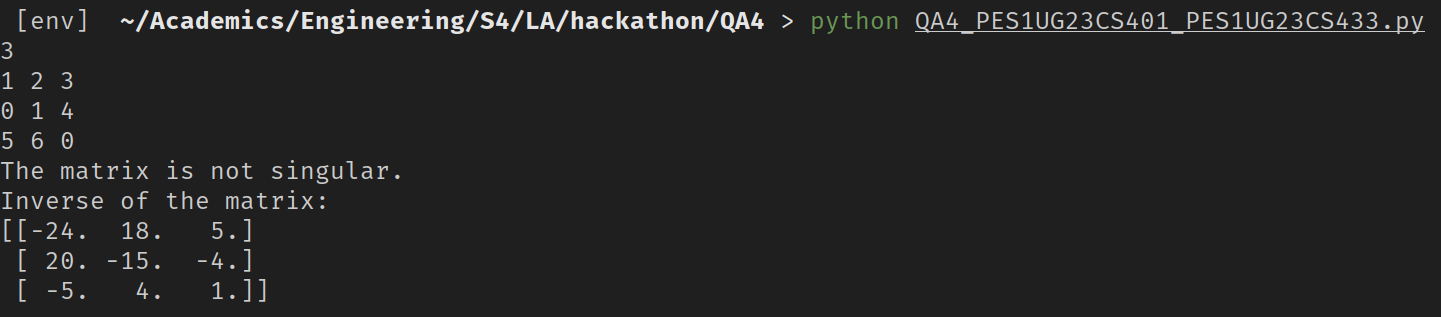
Test Case 2:



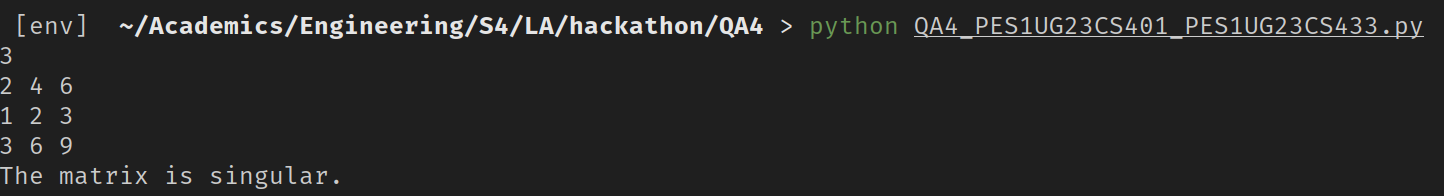
* First we are given a square matrix of size NxN and the column of the B matrix. So we can use the **np.hstack** function in numpy to reshape the NxN matrix to NxN+1 augmented matrix.
* We then **normalize** the row and then ensure all values below pivot are 0. Doing this for all rows gives us the Upper triangular matrix **U**
* We initialize the Vector **X** with 0s using **np.zeros** for solution vector
* Once we have **U** we do backward substitution to get the values of vector **X**

**Application Question 4**

Test Case 1:



Test Case 2:



* The input matrix is determined to be a square matrix if its **shape[0]** is equal to its **shape[1]** else the matrix is considered to be rectangular.
* If the matrix is a square matrix then **np.linalg.inv** function is used to calculate the inverse of the square matrix.
* If the matrix is a rectangular matrix then Moore-Penrose pseudo-inverse is calculated using **np.linalg.pinv** function.
* If the functions raise a **np.linalg.LinAlgError** then the input matrix is **singular** else it is **not singular** and inverse exists.