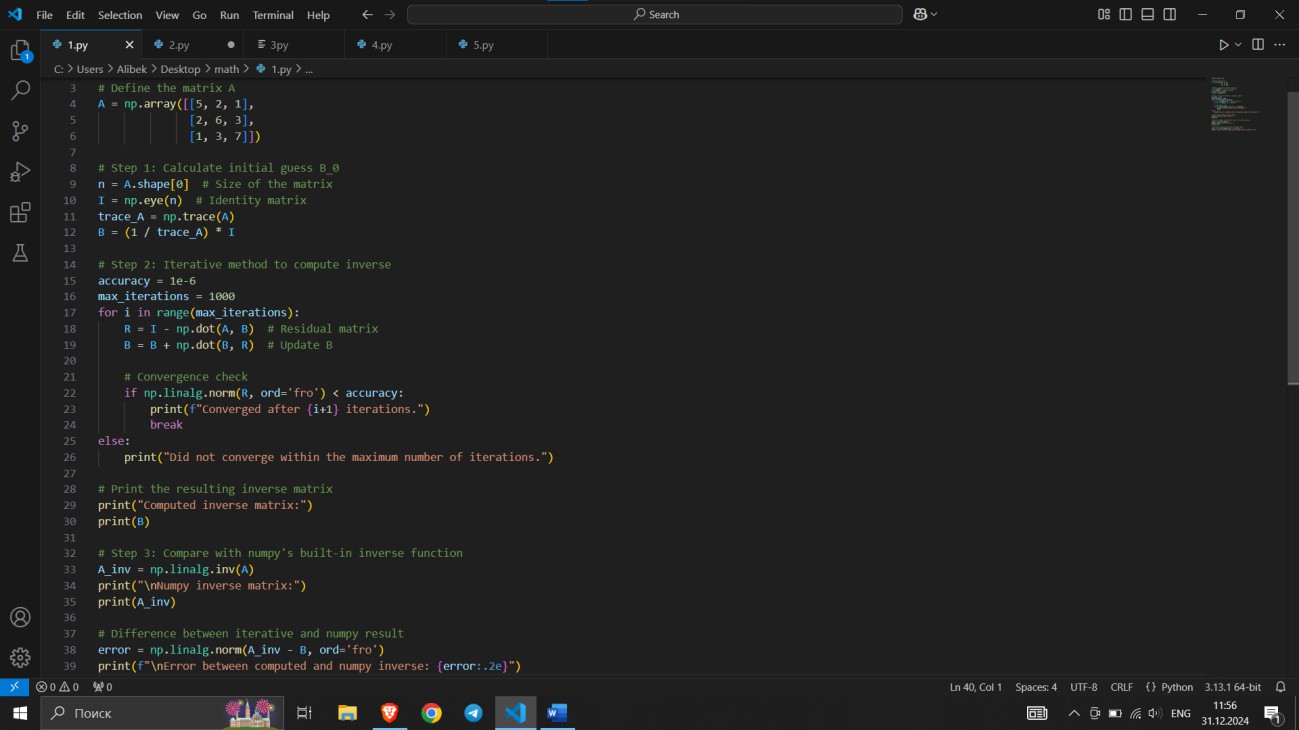
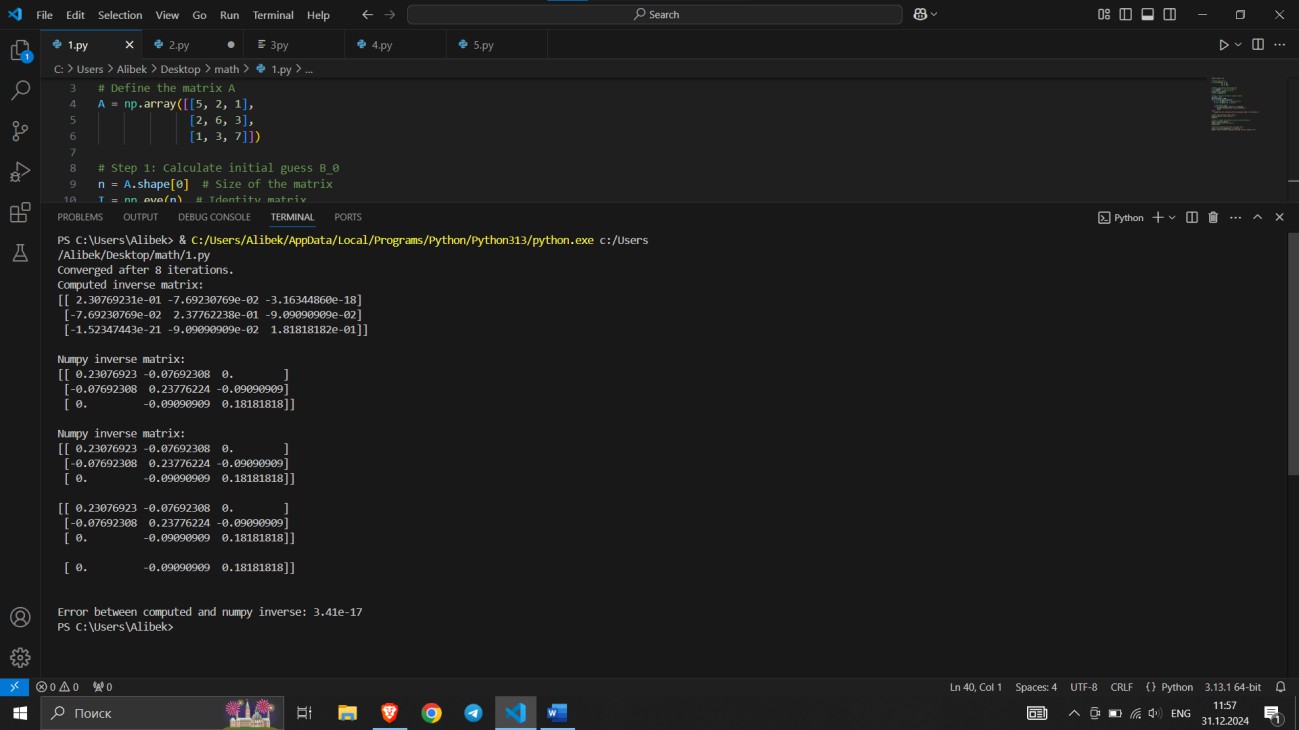
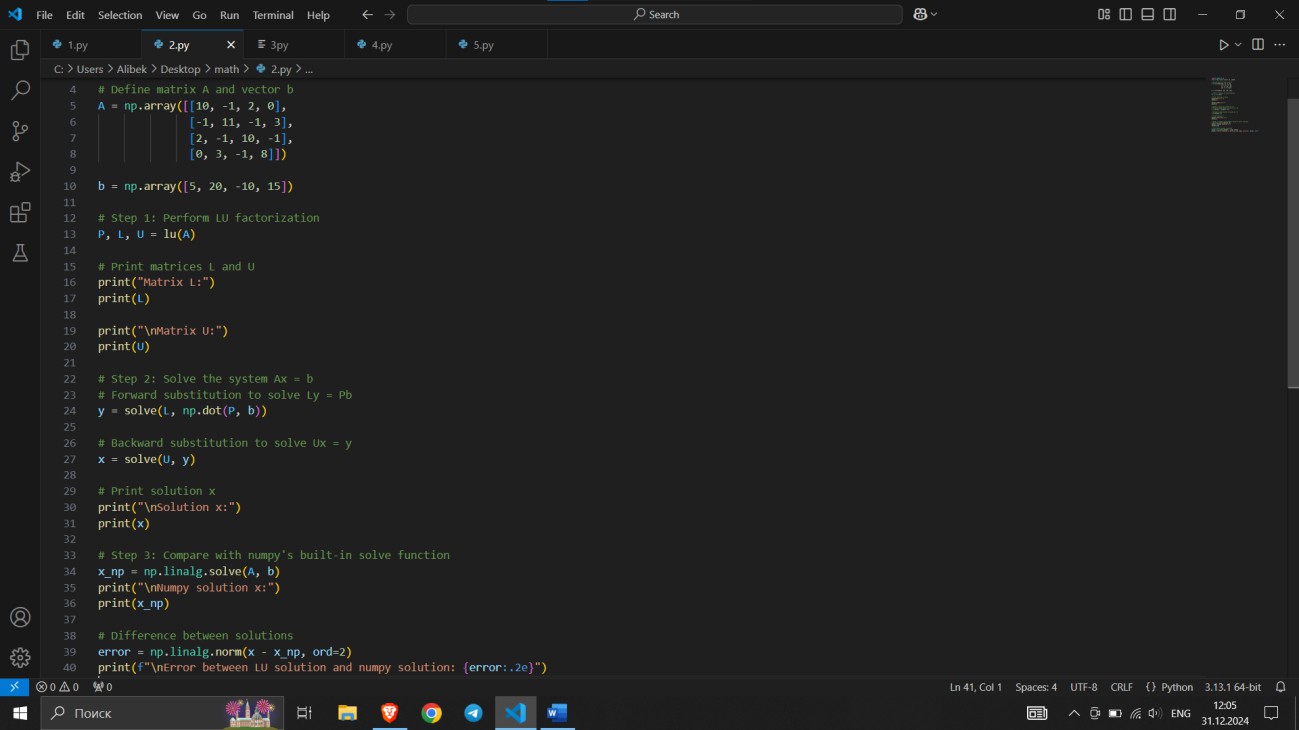
Task 1

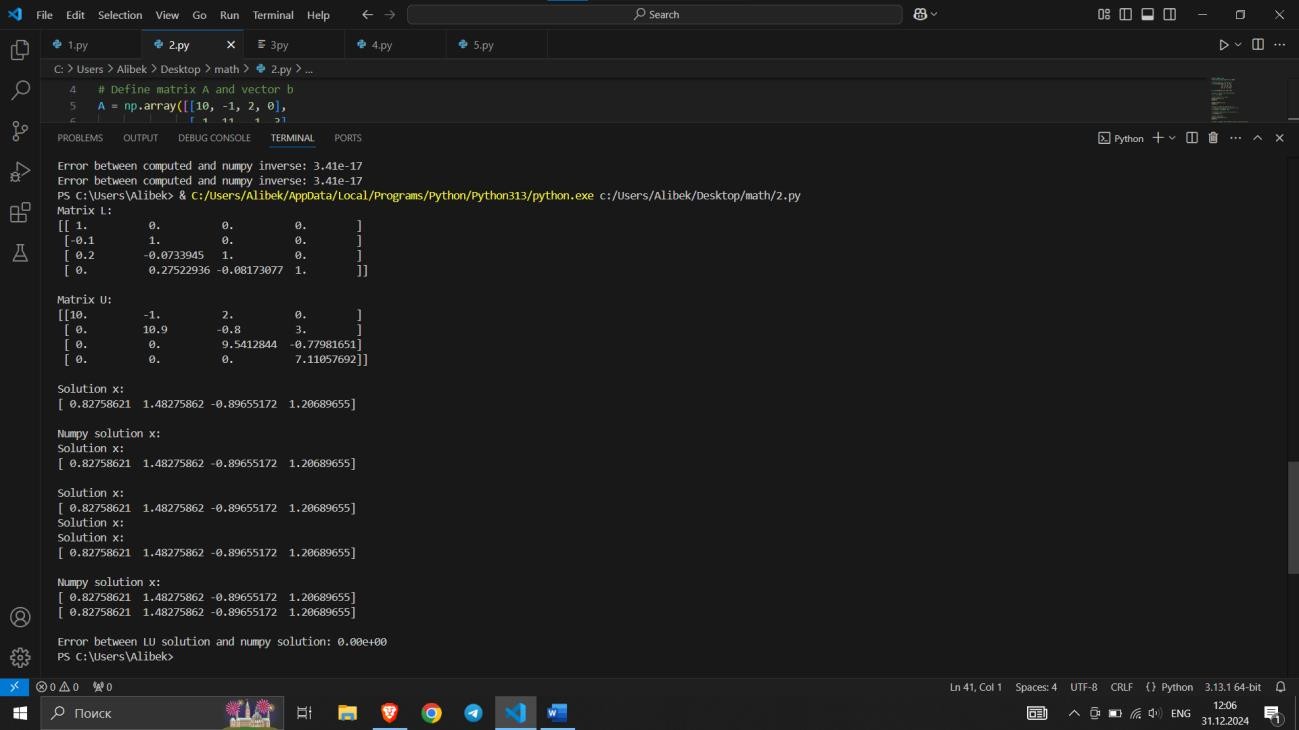




This Python script implements the iterative method for matrix inversion as described. It also compares the result with the numpy linalg.inv function to verify the accuracy of the computed inverse.

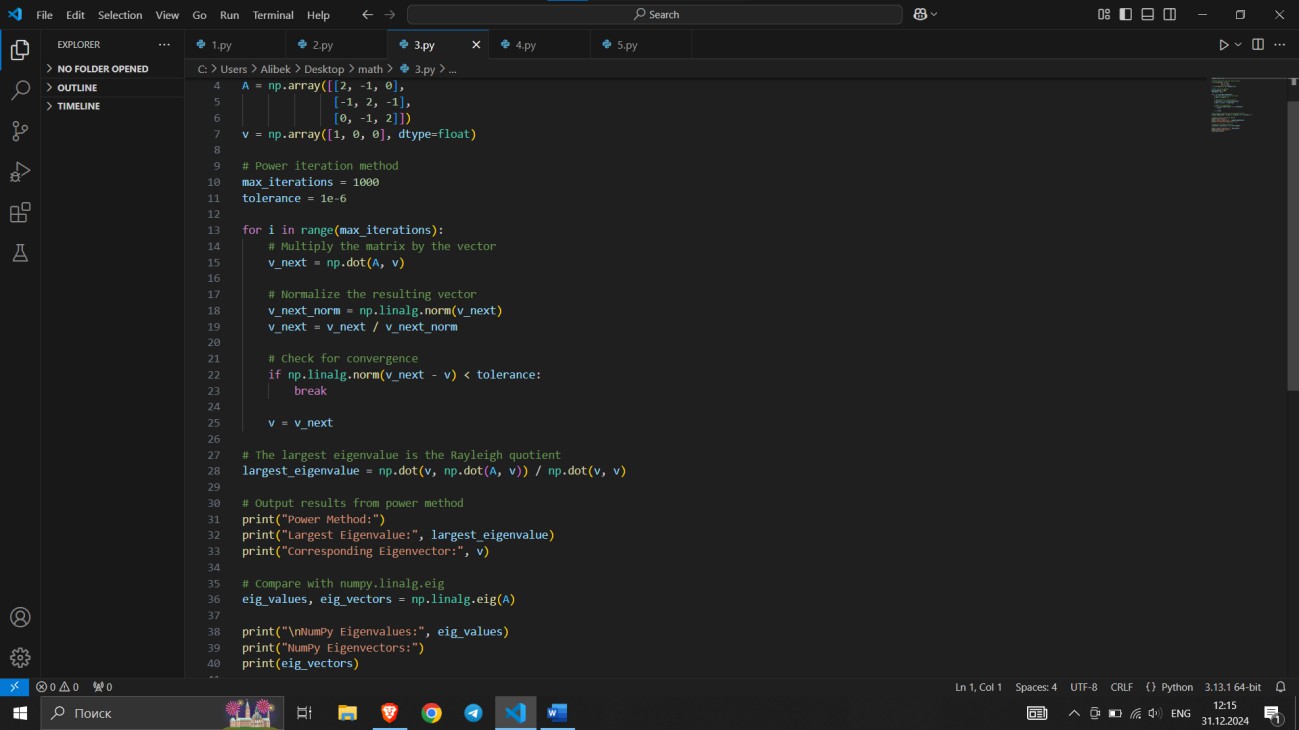
Task 2

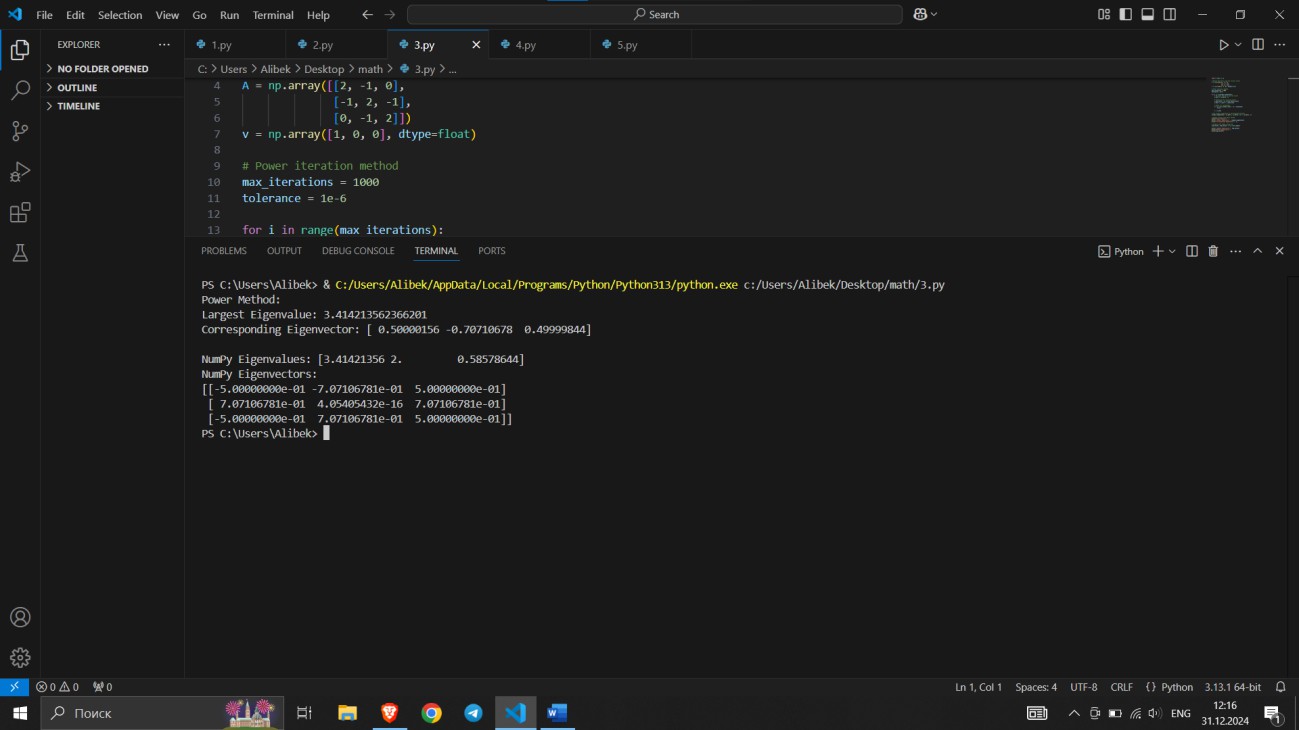




The LU factorization task is implemented and the system is solved using the factorized matrices. The solution is compared to numpy's direct solver to ensure accuracy.

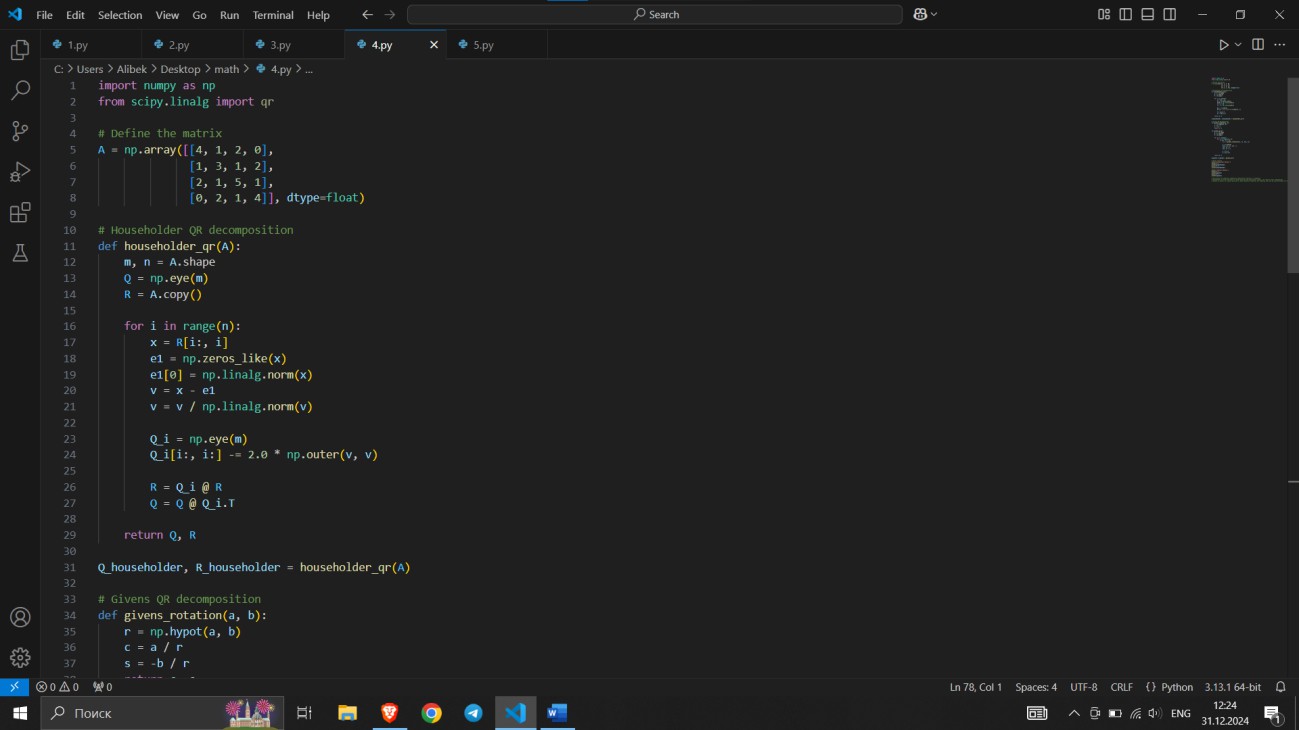
Task 3

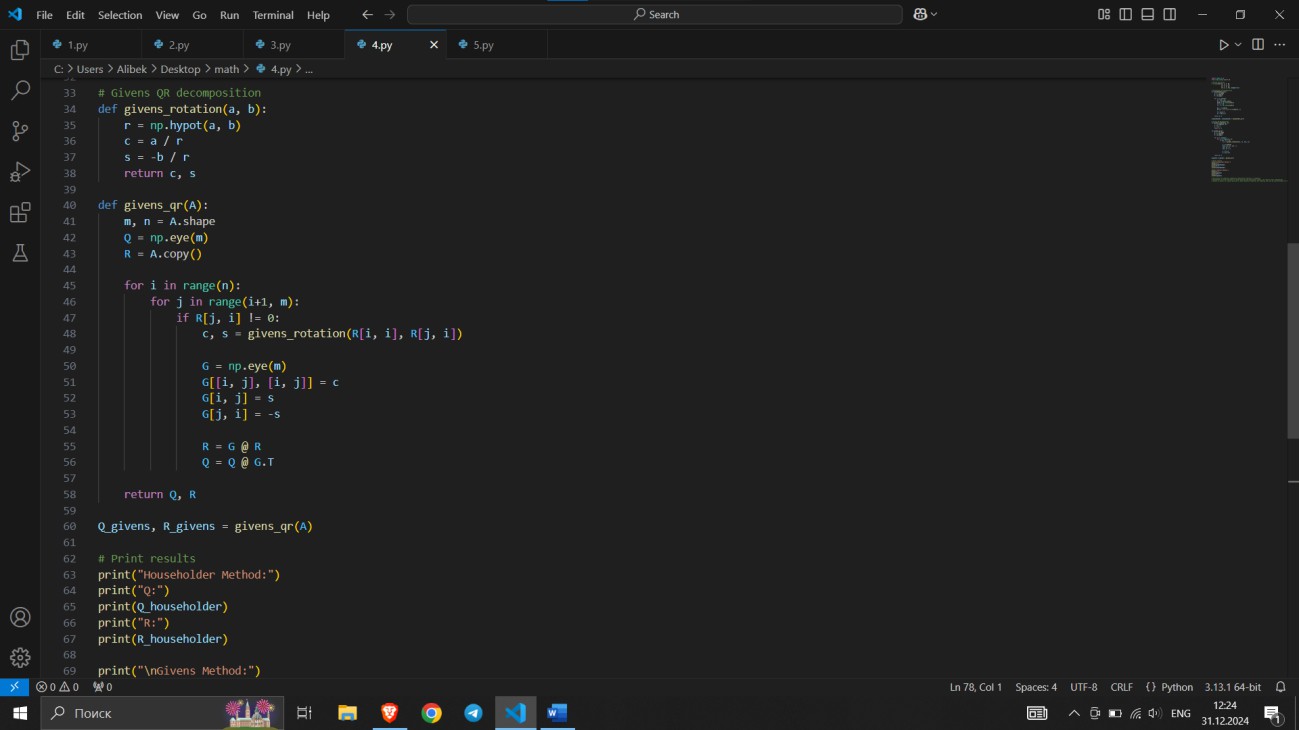


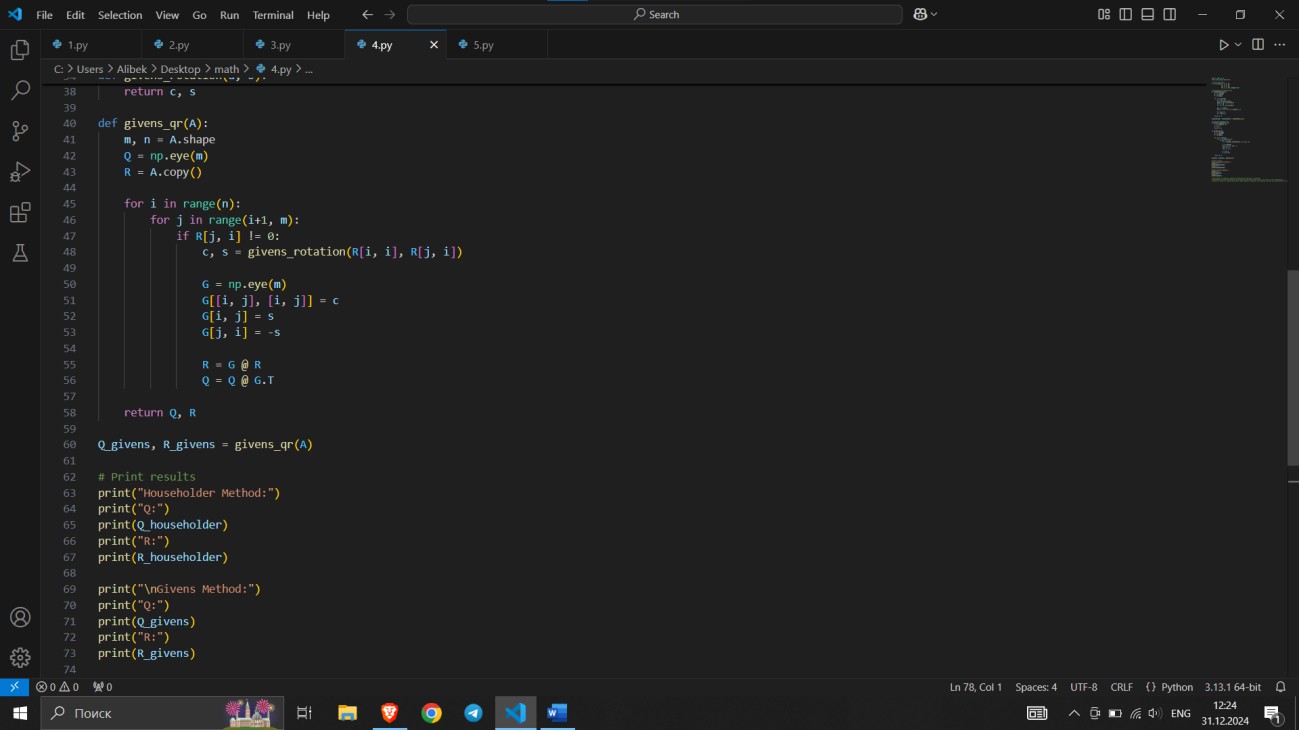


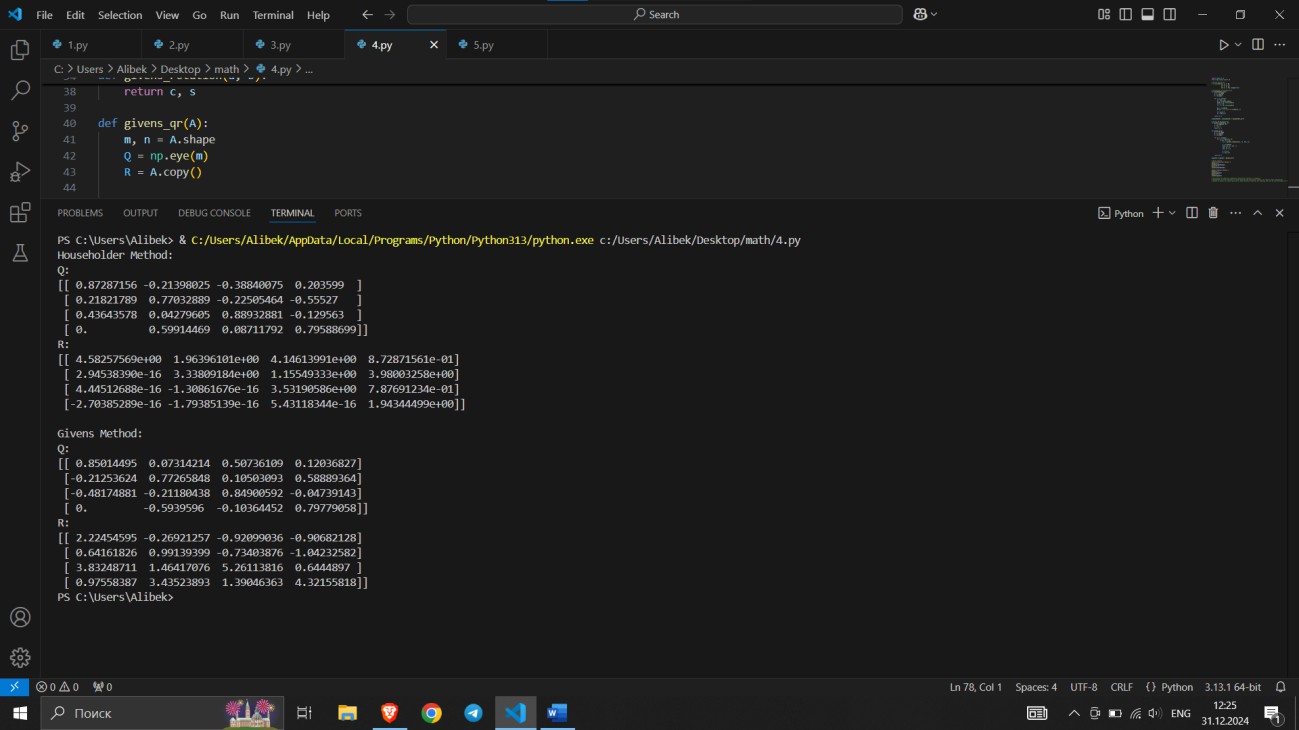
The code implements the power iteration method to find the largest eigenvalue and its corresponding eigenvector of the given matrix. It also compares the result with the output of numpy.linalg.eig.

Task 4



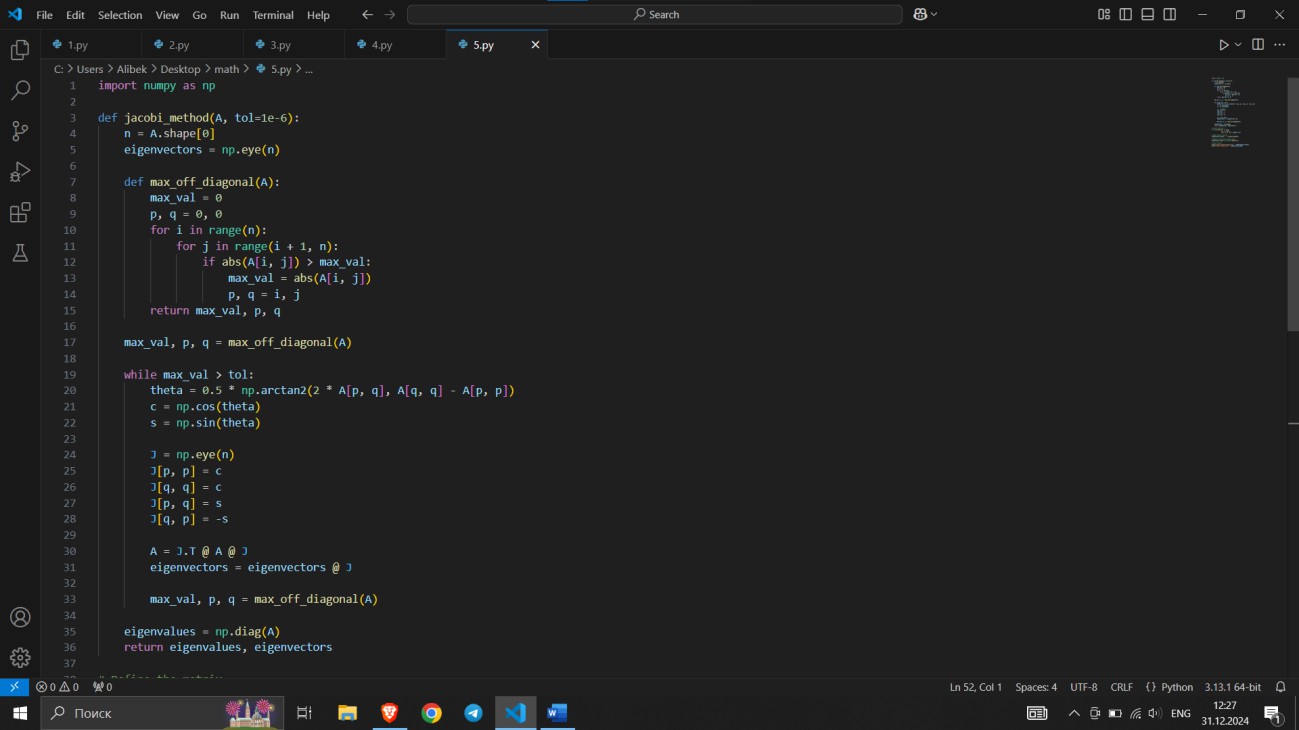


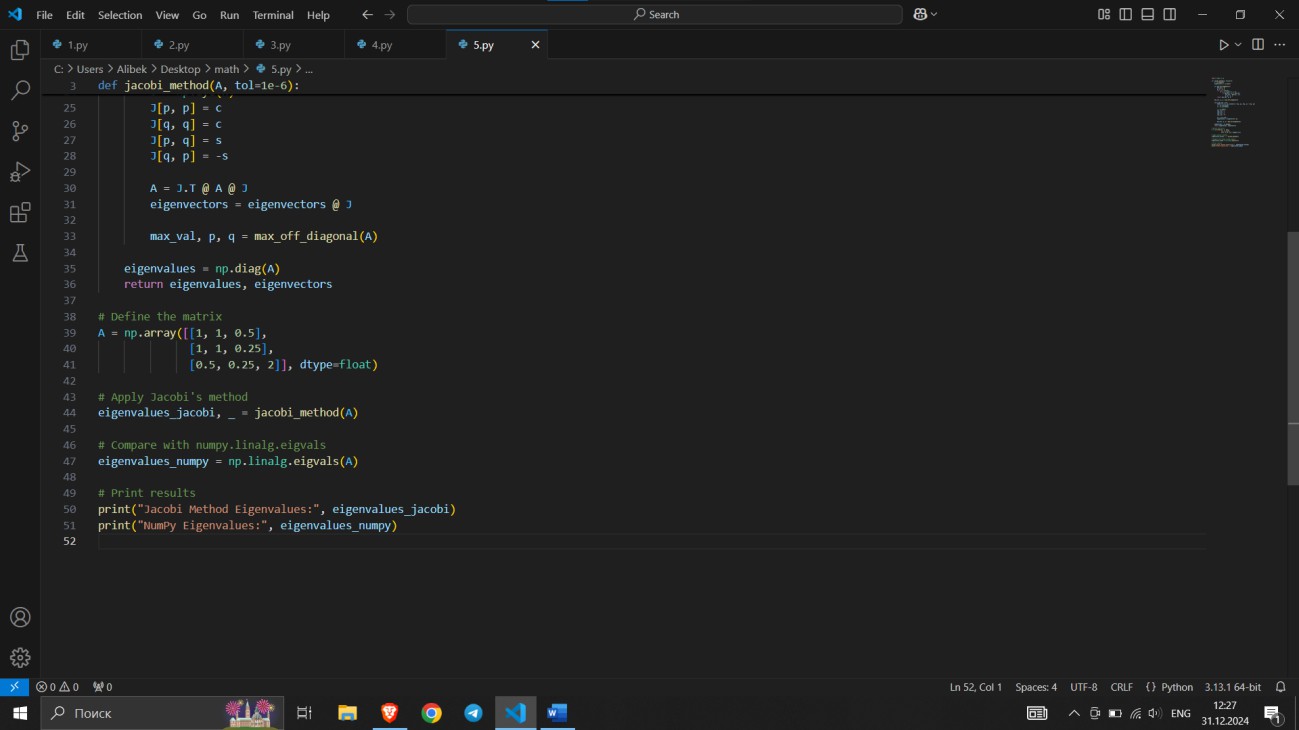


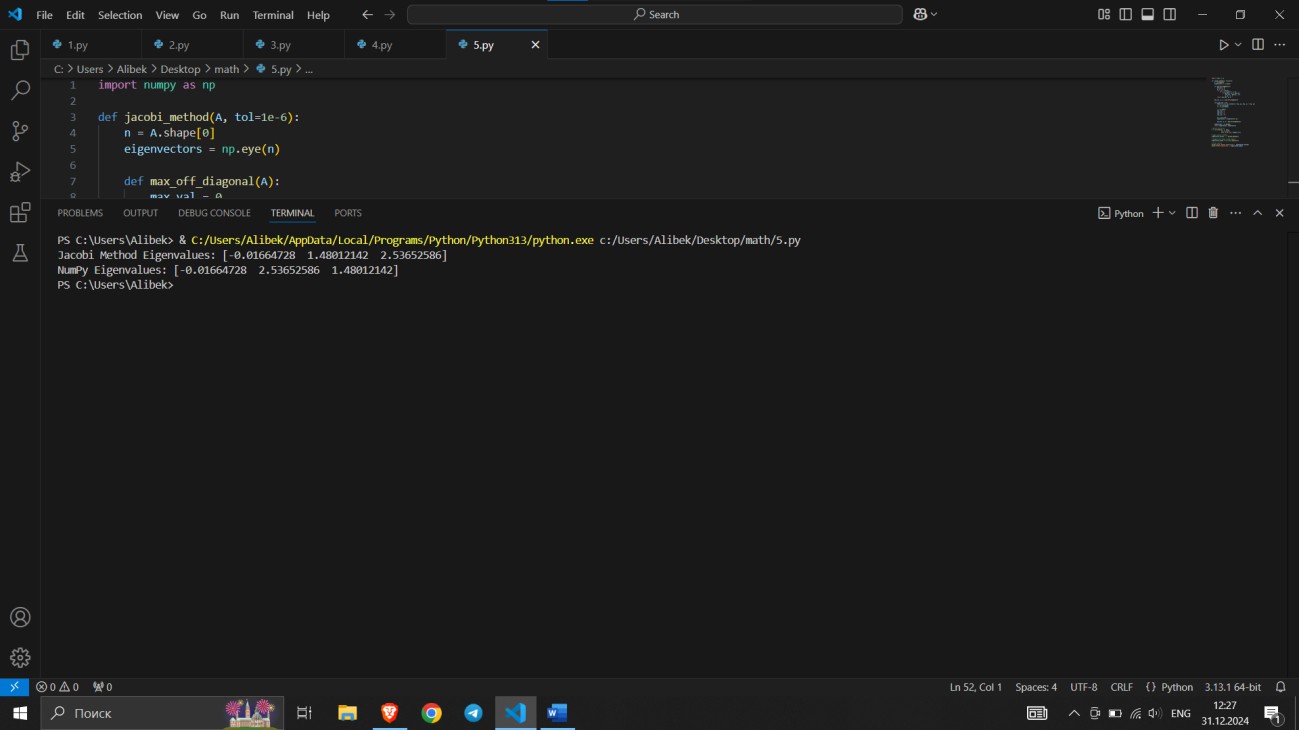


The code above implements QR decomposition using both the Householder and Givens methods, and outputs the Q and R matrices for the given matrix AAA. It also includes qualitative comments comparing the efficiency and numerical stability of the two methods.

Task 5







The code uses Jacobi's method to compute all eigenvalues of the given matrix, with accuracy set to 10−610^{-6}10−6. It also compares the result with numpy.linalg.eigvals.