The attached utf-8 encoded text file contains the tags associated with an online biomedical scientific article formatted as follows. Each Scientific article is represented by a line in the file delimited by carriage return.

[Adolescent] [Adult] [Antidotes] [Child] [Child, Preschool] [Cholesterol] [Chromatography, High Pressure Liquid] [Chromatography, Ion Exchange] [Diet] [Female] [Humans] [Infant] [Male] [Phenylalanine] [Phenylketonurias] [Reference Values] [Retrospective Studies] [Tyrosine] [Ubiquinone] \n

[Diabetes Complications] [Diabetes Mellitus, Type 1] [Diabetic Angiopathies] [Female] [Humans] [Male] [Sex Characteristics] \n

[Anti-Inflammatory Agents, Non-Steroidal] [Benzeneacetamides] [Chemical Phenomena] [Chemistry] [Chromatography, High Pressure Liquid] [Hydroxamic Acids] [Magnetic Resonance Spectroscopy] \n

[Alcaligenes] [Aminobenzoates] [Chromogenic Compounds] [Colorimetry] [Enzyme Inhibitors] [Escherichia coli] [Hydrolysis] [Kinetics] [Nitrobenzoates] [Penicillin Amidase] [Phenylacetates] [Reference Standards] [Spectrophotometry] \n

Write a program that, using this file as input, produces a list of pairs of tags which appear TOGETHER in *any order and position* in at least ***fifty*** different Scientific articles. For example, in the above sample, [Female] and [Humans] appear together twice, but every other pair appears only once. Your program should output the pair list to stdout in the same form as the input (eg tag 1, tag 2\n).

You **MAY** return an approximate solution, i.e. lists which appear at least ***fifty*** times with high probability, as long as you explain why this tradeoff improves the performance of the algorithm.

Please include, either in comments or in a separate file, a brief description of the run-time and space complexity of your algorithm.

Your solution should preferably be implemented in Python. Please include compilation/runtime instructions with your code.