Instructions: You must **not** communicate with anyone else using electronic media. Use the Python programming environment to attempt this assessment. You are required to upload your solutions as a Python file (.ipynb) on Moodle under the assignment section once you have completed the exam. The file should be named using your **UID**, for example, UID1234567.ipynb. You will be allowed a maximum of five attempts to upload your submissions and the latest submission will be considered as your final answer. Ensure the file contains all the necessary code as well as all written explanations. Ensure that the code contains all steps leading to the final answer, including calculations, functions, and any other relevant details. Merely copying answers provided by an online AI service, like gpt or Gemini, may not fetch you any credit. In case of concerns, the instructor reserves the right to interview you while evaluating your submissions to authenticate your proposed solutions.

[Max. time = 2 hours | Max. score = 10]

NOTE: You should **not** use a `counter` explicitly anywhere in your program or your solution to identify the communities. You must resolve the problem by developing the suggested model - no other method will be accepted.

==========

Creating communities based on voting preferences

Question: Consider a population of $N = 100$ citizens with multiple affiliations and personas that are publicly known. Citizens are identified by their respective numerical IDs that range from 1 to N. **Your task is to develop a strategy to create disjoint communities of people with similar types based on their individual voting preferences using concepts from linear algebra as suggested below.** The exercise begins with a mega poll wherein every individual elects one person from the entire population based on their preferred types through a secret ballot. Individuals cannot vote for themselves and can only vote for one individual only. It is possible that an individual may abstain from voting (i.e., abstaining is also considered a vote). In order to access the results of the mega poll, you are instructed to access the `numpy array` in the file provided to you on Moodle with filename ⟨`xy`⟩`.ipynb` under the category *Feedback files*. The first column in the array refers to the `voter ID` and the second column refers to their preferred option (in terms of their ID), i.e. a certain row of this matrix may read $(33, 21)$ to imply that voter with $ID = 33$ has voted for a person with $ID = 21$. The model, suggested here, groups any voter and the voter's pick (expressed in the ballot) as members of the same community because it considers both these citizens to be aligned along similar types (values and personas).

Design your model to create the aforementioned communities by solving the following sub-tasks.

1. Create an appropriate digital encoding of the voting activity. Consider encodings with either binary or ternary or quaternary digits, whichever you think is most appropriate to develop the model. Clearly enumerate, using a python programming environment, the voting activity for every individual in the language of your chosen digital encoding. **[3]**
2. Gather the digitally encoded voting result for every individual in the form of a vector and tile these vectors as columns of a matrix M such that the columns range from 1 to N in accordance with the IDs of the citizens. **[0.5]**

In this model, for creating the disjoint communities based on voting preferences, we will ignore all instances of reciprocal votes from the voting results because these choices are regarded as a *mutually orchestrated fix* and do not represent actual synergies in types.

3 Find the reduced row echelon form of your matrix M. **[0.5]**

4 Extract all columns from M that correspond to non-reciprocal votes into a new reduced matrix M_R. **[2]**

5 Do the columns of M_R constitute a basis of $Col(M)$? Explain your answer. **[1]**

6 Now devise an appropriate matrix transformation to M_R so that the columns of the transformed matrix M_{R_T} can be mapped to the IDs of every individual and thereby capture the entire population. **[1]**

7 Find an appropriate vector space using M_{R_T} in order to readily identify the disjoint communities within the population. Now, enlist the communities by their IDs. **[2]**

□