This program takes the 1:many mappings of ICD10 codes and reduces them to 1:1 mappings by using the ICD9 target that was most frequently used in 2015. I do this in a series of steps:

1. Read the data from the GEM txt files using pandas and create a dataframe (a table).
2. Create the pertinent designations for HW questions for part I. I create a column with a Boolean value for source ICD codes with “No Map”. I then remove those which are “True” (i.e., no mappings) from the main GEM table
3. Create a "counts" table. This gives me 2 columns: source icd codes, and counts (# of mappings) for that source code.
4. Find those with only unique values (count==1) and create a table with unique (1-to-1) mappings
5. Create a table with just 1:many mappings (counts>1). (Note, this table shows the ICD source code, and the # of mappings. It is does not show the mapping of the source code to each target code individually.)
6. Sum the occurrences of 1:1 & 1:many mappings to answer the HW questions
7. Merge the 1:many counts table (step 5) with the original I10->9 GEM dataframe to create a table of just the 1:many mappings.
8. Import the ICD 9 frequency table, and merge that with the table of ICD 10 1:many mappings (step 7). This creates a table that lists each mapping and the frequency that the target code (ICD9) was used in 2015. If an ICD 9 code was used most frequently, that could be used as a way to reduce the 1:many mappings down to 1:1.
9. By merging the two tables, one could select the ICD9 target that was used most frequently for a given ICD 10 source code (via the Dataframe.max() method and grouping by ICD 10) to reduce the 1:many mappings (i.e., “forcing” a 1:1 mapping).

After running our program, we were able to reduce the 1:many mappings from 17751 down to 7712 1:1 mappings based on the most frequently selected ICD9 codes in 2015.