Goals

1. Write code to read resources from a file into memory.
2. Write functions to efficiently insert, find, edit and delete from the set of resources.

Problem

You are given a file called people.csv which is formatted like this, but with 1,000,000+ rows:

id,firstname,lastname,email,age
1,John,Smith,JSmith@wwken.com,30
2,John,Jones,jonesy1234@wwken.com,45
3,Mary,Jones,mjrules@wwken.com,29
4,Jane,Doe,janedoe@wwken.com,40

Using whichever language you'd like, write the following functions:

read_from_file(filename)
This function should iterate over all rows in filename and hold each person resource in memory for future operations.

find_by_id(id)
This function should return the person with the specified id or false if no person is found.

find_by_first_name(name)
This function should return an array of all the people whose first name exactly matches the name argument.

find_by_name_prefix(prefix)
This function should return an array of all people whose first or last name starts with the prefix argument.

find_by_email(email)
This function should return a single person whose email matches the argument or false if none is found.

find_where_age_between(min, max)
This function should return an array of all people whose ages are between the min and max ages given

insert(), edit(), delete()
These functions should provide the ability to add, edit or delete a single person from the in memory collection of people. There's no need to write anything to the people.txt file with these functions. You may choose what arguments and return values make sense here.

Assumptions

- Assume you have infinite memory available unless otherwise stated.
- Assume that you only need to read from a file once and it will be the first function called.

- Assume that once the file is read into memory it may be discarded. There's no need to update the file when inserts, edits or deletes are made. They only need to affect future find calls.
- Assume that ids are unique incremental integers, first and last names may be repeated, emails are unique and case insensitive, and ages are integers in the range of 0-100