Write Python code that prints and implements the following menu:

1 - open address book
2 - add entry
3 - remove entry
4 - store address book
5 - view address book by name, alphabetically
6 - view address book by email, alphabetically
7 - search email addresses
8 - search names
9 - search names and emails
10 – quit

Menu Meanings

1. User will be prompted to enter a filename in its entirety (they would type addresses.csv, not just addresses). The program then opens it, reads in all entries for later usage. The contents of this file replace the previous address book; any unsaved changes would be lost.
2. User will be prompted for a name, then an email address, and the pair of info will be added to the table. A name might have multiple email addresses associated with it, but an email address can only have one name associated with it. (What will this mean when you choose your data representation?)
3. User is asked for an email address to remove; the email/name pair is then removed from the address book if present. Program then either prints "it was removed" or "that was not found in the address book."
4. User will be prompted for a filename to store the current address book in CSV format; the program then performs this action.
5. Program finds and lists all names alphabetically with the associated emails in two columns: names in first column, emails in second column. Both columns must be left-aligned, and each column should be just wide enough to hold the longest name or email address. If a name has multiple email addresses, they must all be listed in adjacent rows (but the ordering of those rows doesn't matter).
6. Program finds and lists all emails alphabetically with the associated names in two columns: emails in first column, names in second column. Both columns must be left-aligned, and each column should be just wide enough to hold the longest name or email address.
7. Ask the user for an email address. Either print out the associated name, or print "no entry found."
8. Ask the user for a name. Either print out all associated emails, or print "no entries found."

9. Ask the user for a string. First print out emails-and-names, a blank line, then names-and-emails, where the name or email matches.
10. Program quits. Any recent changes to the address book that weren't stored will be lost.

Requirements

- The menu should be repeated and reprinted each time the user makes a choice, until they quit.
- You must write a separate function each for options #1 through #9. You are encouraged to create any other functions you find useful.
- You must use a dictionary to store the address book while your program is running.
- You must store your files in the "CSV" formatting described below. This will allow your program to interact with other CSV-capable programs, as well as make grading feasible.
- You may not use global variables – if you need to communicate values between functions, you will have to use your parameters/arguments to do so.
- Initially, the program opens with an empty address book.

CSV Format

Comma-Separated Value (CSV) files are an old "standard" for files. Each line contains text values separated by commas. The first row should contain column-descriptions instead of a specific row of data. Furthermore, every single text value will be surrounded by quotes.

Example:

```
"Olympian","number of medals","sport"
"Michael Phelps","22","swimming"
"Jim Thorpe","2", "pentathalon,decathalon"
"Mia Hamm","3","soccer"
```

Notice the comma inside Jim Thorpe's sport. This is why we require quotes around the entry. If we furthermore wanted quotes inside an item, we could escape them with backslashes. The above format is how we must format our files. CSV isn't quite a standard – some versions use tabs, periods, or spaces instead of commas. Some don't require quotations around items. But we must use the above formatting for our assignment for consistency's sake. Whenever you have entries stored in your program's current address book, of course no quotations should be present – not in a name, and not in an email; they are only present in the .csv files that are storing the information