

For today's lab you'll need a large data files, so start the lab by copying the directory
P:\Math-CS\810-051-fienup\common\lab6 to your P: space (or desktop or flash drive).

Part A: We'll redo part C from lab 5. Write a program that processes a file of customer records to generate mailing labels for only female customers living in Iowa (IA). The customer records are in the file customerData.txt. **Each customer record is on a single line with 12 fields separated by commas(',')**. The order of the fields on a line is: First Name, Middle Initial, Last Name, Street Address, City, State, Zip Code, Country, Email Address, Telephone Number, Gender, and Birthday.

The general approach you should take is to:

- read each line as a string,
- split the line into a list of strings for the fields,
- if the record satisfies the criteria listed above (female from Iowa), build the label by concatenating select elements from the list, and
- write the mailing label to the file (mailingLabels.txt)

Each mailing label should be formatted as below with 5 blank line separating each label:

Jane Smith
123 Main Street
Cedar Falls, IA 50613

After you have the program for part A working correctly, raise your hand and we'll check your work.

Part B: In part B, we'll process the same data file: customerData.txt, but your program should tally the number of customers from each state, and print the number from each state sorted from most to least. One way to do this is to use parallel lists: one to maintain the state initials and one to maintain the count of each state. For example, the following diagram shows 'OR' with a count of 54 occurrences, 'NY' with a count of 115 occurrences, etc.

```
statesList = [ 'OR', 'NY', 'ID', ....]  
statesCount = [ 54, 115, 17, ...]
```

These lists would initially be empty lists (statesList = [] and statesCount = []) with the general processing being:

- read each line as a string,
- split the line into a list of strings for the fields to get the state
- check to see if the state is already in the statesList. If it is, find the index of the state and use that index to increment the corresponding count in the statesCount list
- if the state is not in the statesList, then add it to the end of the statesList and add a 1 to the end of the statesCount list.
- After processing the whole file, make a list of lists that combines the statesList and statesCount such as:

```
combinedStatesList = [ [54, 'OR'], [115, 'NY'], [17, 'ID'], ....]
```

- The sort method can then be used to sort the list of lists
- Print the sorted state and number-of-occurrence pairs to the screen.

After you have the program for part B working correctly, raise your hand and we'll check your work.

When you are done with all parts, hand in the following: both program for part A and B

If you don't get all parts done in lab, don't worry about it. Try to have the lab completed by next lab.