Instructions

Write a Ruby program to first assemble a single set of records by parsing data from 3 different file formats and then display these records sorted in 3 different ways.

PURPOSE

We hope that this exercise will allow us to evaluate your skills as a developer. The qualities that we pay special attention to are:

- Simplicity/elegance of design
- Adherence to good software engineering principles
- Maintainability (clean, easy to understand code)
- Effective use of the standard library
- Use of unit tests

INPUT

A record consists of the following 5 fields: last name, first name, gender, date of birth and favorite color. You will be given 3 files, each containing records stored in a different format.

- The pipe delimited file lists each record as follows:
  LastName | FirstName | MiddleInitial | Gender | FavoriteColor | DateOfBirth
- The comma delimited file looks like this:
  LastName, FirstName, Gender, FavoriteColor, DateOfBirth
- The space delimited file looks like this:
  LastName FirstName MiddleInitial Gender DateOfBirth FavoriteColor

You may assume that the delimiters (commas, pipes and spaces) do not appear anywhere in the data values themselves. Write a Ruby program to read in records from these files and combine them into a single set of records.

OUTPUT

- Create and display 3 different views of the recordset (see sample-output.txt):
  - Output 1 – sorted by gender (females before males) then by last name ascending.
  - Output 2 – sorted by birth date, ascending.
  - Output 3 – sorted by last name, descending.
- Ensure that fields are displayed in the following order: last name, first name, gender, date of birth, favorite color.
- Display dates in the format M/D/YYYY.

PACKAGING

- Please package the code in a zip or tar archive when you send it to us.
- Tell us: 1) which script or Rake task to run in order to produce the desired output from your program and 2) how to run the tests.
- Specify what version of Ruby you're using.
- You may use the core and the standard library, but no gems, to implement your solution. If you create any Rake tasks, please specify the version of Rake required. You may use any testing libraries for unit tests (such as RSpec, MiniTest, Test::Unit, etc.) – please specify which gems and versions are required.

Solutions as well as questions and comments should be sent to us at
jhanks@bybaxter.com