Challenge

- **Do not use Rails**
- Think about writing **readable and clean code**
- Channel Sandi Metz: think Object Oriented! Review POODR and check out LaunchSchool's nice online book as a start
- Use RSpec for **unit tests**
- **Think simple**! Think about responsibilities of classes and draw them out before jumping into code.
- Make a git repo and be sure to commit in steps so you can review your process.

Step 1 - Build a system to parse and sort a set of recipe records

- Create this a ruby **command line app**
- App takes as **input a file with a set of records** in one of three formats described below, and **outputs (to the screen) the set of records sorted in one of three ways**.

Input

Write a Ruby program to read in records from these files and combine them into a single set of records.

The input is 3 files, each containing records stored in a different format. Create

these files yourself, and make (and note) any assumptions if it makes solving your problem easier. A record consists of the following 4 fields:

- `name`
- `category`
- `cook_time`
- `servings`

an example of the format and what one record might look like:

```ruby
#The pipe-delimited file lists each record as follows:
name | category | cook_time | servings
apple pie | dessert | 90 minutes | 8

#The comma-delimited file looks like this:
name, category, cook_time, servings
apple pie,dessert,90 minutes,8

#The space-delimited file looks like this:
name category cook_time servings
'apple pie' dessert '90 minutes' 8
```

Output

Create and display 3 different views of the data you read in:

- **Output 1** – sorted by category, then by name, ascending.
- **Output 2** – sorted by cook_time, ascending.
- **Output 3** – sorted by servings, then cook_time, descending.

Step 2 - Build an API to access your system

Tests for this section are required as well. Your assignment is to build a standalone API using a ruby library like Grape or Cuba, with the following endpoints:

```ruby
POST /recipe #Post a single data line in any of the 3 formats suppor
GET /recipes #returns recipe records, sorted by category, then by na
GET /recipes/:category/name #returns recipe records for the given ca
GET /recipes/cook_time #returns all recipe records sorted by cook_ti
```

It's your choice how you render the output from these endpoints but make sure it's well structured data. These endpoints should return JSON.

Step 3 - Review and Refactor

Make sure you commit before you refactor, and that all your tests are passing. Take a stab at refactoring but we'll also go through and review/refactor together.