**Pipeline Coding Problem**

We would like you to complete a small coding problem so we can gauge your problem solving, coding, and application design skills. You can assume that your application will be run as part of a data processing pipeline for several years, and that it should include error handling and logging to help with troubleshooting problems that may arise in the future. Also assume the application will need to be modified in the future by other developers on the team so writing clear, readable code is important.

The assignment is to create a command-line/console application that provides a main method that accepts input (CLI) parameters as needed based on the problem description below. The application will create the appropriate data structures and execute an algorithm for processing 1 or more input data files.

We will test the application with several different input files processed independently and together in a group.

**The Problem**

Users of the Corteva website can provide their name and email address in order to register for a free newsletter written by Corteva agronomists on Pioneer seed products. Corteva wants to share the list of newsletter subscribers across multiple parts of the business using their centralized data warehouse. Data must be loaded to the data warehouse via a RESTful API using a JSON format. The problem is to convert 1 or more CSV-formatted files containing the exported list of new subscribers into JSON format for import into the data warehouse. The output of the command-line driven application should be a single file containing a list of JSON objects. This application must be written using a modern, flexible scripting language such as Python, Ruby, or Perl. A sample input file named pipeline\_input\_user\_export-vx.csv is included with this problem statement.

A very basic JSON format for the output has been proposed by the data warehouse team and they have provided an example named user\_list\_ex.json and a schema description in user\_list\_schema.json. You may use this schema for your json output or you may modify it. If your application uses a modified schema, please provide a new schema description file to describe your schema.

Extra credit if you can wrap your solution in a dockerfile with your script as the entrypoint and provide us the docker command required to run your container with the input and output mounted in.

Links to how to install docker desktop on your machine can be found here:

<https://hub.docker.com/editions/community/docker-ce-desktop-windows><https://hub.docker.com/editions/community/docker-ce-desktop-mac>

**Programming Languages and Libraries**

Write the code using a modern scripting language such as Python, Ruby, or Perl. Python is our preferred language, but you should choose one you are comfortable with. You may use available open source modules.

**Submission**

We would like to have the script, a list of all module dependencies, and operating instructions provided at least 24 hours prior to your scheduled interview. If you can provide it earlier, then please do. To deliver the application source code and associated documentation, please upload it to a public Git repository such as GitHub, GitLab, or BitBucket. If that is not possible, please contact us to make alternative arrangements.

**Final Thoughts**

We expect this exercise will take a few hours to complete. If you are limited in the time you have available to complete the work, please let us know that with your problem submission and we will account for that in the results. If you have any questions about the problem or how to approach it, please feel free to contact us. If you are unable to finish this prior to the 24 hour cutoff, please submit what you have finished and an explanation of what is working.