Data Engineering Coding Challenge

Problem 1

Parse fixed width file

- Generate a fixed width file using the provided spec (offset provided in the spec file represents the length of each field).
- Implement a parser that can parse the fixed width file and generate a delimited file, like CSV for example.
- DO NOT use python libraries like pandas for parsing. You can use the standard library to write out a csv file (If you feel like)
- Deliver source via github or bitbucket
- Bonus points if you deliver a docker container (Dockerfile) that can be used to run the code (too lazy to install stuff that you might use)
- Pay attention to encoding

Solution Steps: Code file - (parse.py)

Step 1: Generate a Fixed Width File

First, we need to create a fixed width file based on a given specification.
Let's assume we have a specification for the fixed width file and a function to generate a fixed width file based on the above data and specification.

Step 2: Implement a Parser

Next, we write a parser to read the fixed width file and convert it to a CSV file.

Step 3: Packaging in a Docker Container

And now we'll provide a Dockerfile to run the code.

```
# Use an official Python runtime as a parent image
FROM python:3.9-slim

# Set the working directory in the container
WORKDIR /app

# Copy the current directory contents into the container at /app
COPY . /app

# Run the script
CMD ["python", "your_script.py"]
```

Problem 2

Data processing

- Generate a csv file containing first_name, last_name, address, date_of_birth
- Process the csv file to anonymise the data
- Columns to anonymise are first_name, last_name and address
- You might be thinking that is silly
- Now make this work on 2GB csv file (should be doable on a laptop)
- Demonstrate that the same can work on bigger dataset
- Hint - You would need some distributed computing platform

Solution steps: Code files (data_processing.py, spark_processing.py)

1. Configure Intellij IDEA to use the Virtual Environment.
2. Open Intellij IDEA and create a new project or open your existing project.
3. Configure the Project Interpreter:
   - Go to 'File > Project Structure' (or 'Intellij IDEA > Preferences' on macOS).
   - Select 'project > Project Interpreter'.
   - Click the gear icon and select 'Add…'.
   - Choose 'Existing environment'.
   - Navigate to the Python interpreter in your virtual environment, typically my_project/venv/bin/python.
4. Synchronize the Project:
   - Right-click on your project directory in the Project Explorer.
   - Select Synchronize 'project_directory'.
5. Create and Run Python Scripts in IntelliJ IDEA

a. **Create a Python Script for Generating and Anonymizing the CSV File**:

- Right-click on your project directory and select New > Python File.
- Name the file **data_processing.py**.

(we enter the code here)

b. Create a Python Script for PySpark Processing:

- Right-click on your project directory and select New > Python File.
- Name the file **spark_processing.py**.

(we enter the code here)

c. **Run the Scripts in IntelliJ IDEA**:

- Right-click on data_processing.py and select Run 'data_processing' to generate and anonymize the CSV file.

- Right-click on spark_processing.py and select Run 'spark_processing' to process the CSV file with PySpark.

Troubleshooting Steps

- **Check Python Interpreter in IntelliJ IDEA**:
  - ➢ Ensure IntelliJ IDEA is using the correct Python interpreter.
  - ➢ Go to File > Settings > Project > Project Interpreter and ensure it points to the virtual environment.
- **Ensure PySpark is Installed in the Correct Environment**:
  - ➢ Verify that PySpark is installed in the virtual environment being used by IntelliJ IDEA.
- **Invalidate Caches/Restart IntelliJ IDEA**:
  - ➢ Go to File > Invalidate Caches / Restart and select Invalidate and Restart.

Output files:
Problem 1 output file name: output.csv
Problem 2 output file name: sample.csv, anonymized_sample.csv and anonymized_sample_spark.csv