Spark Interview test instructions

In this test, you are asked to write a Spark Application that will handle some data manipulation: an input dataset is to be loaded and parsed, then some of the fields in this dataset have to be replaced using lookup information that is made available in a specific input file and finally, the resulting data is to be saved back to some text files.

input data format

The input data is read from /tmp/gaia-spark-test (`DATA_DIRECTORY`) and the application should read:
- an arbitrary number of files called filen.txt (with n any number)
- an additional file called ip-lookups.txt where the lookup reference data is stored

```
$ tree /tmp/gaia-spark-test/
/tmp/gaia-spark-test/
├── file1.txt
├── file2.txt
├── file3.txt
├── file4.txt
├── file5.txt
├── file6.txt
├── file7.txt
├── file8.txt
└── ip-lookups.txt
```

The filen.txt files contain one record per line, each **valid** record is a comma separated list of at least two elements:

```
$ head -n3  /tmp/gaia-spark-test/file1.txt
961ccf9e-6928-4560-b811-0d48b13df1a5,18.144.83.42
67d7ce0e-f676-4f8d-a4c5-11d5b039d677,150.100.113.182
135cd4cc-056c-415f-9719-34d9aac09324,195.208.240.73
```

The first element is a unique id and the second is an IP address

The ip-lookups file contains one record per line, each record is a comma separated list of two elements:

```
$ head -n3  /tmp/gaia-spark-test/ip-lookups.txt
64.1.199.159,ec416de6-3c7e-4b9b-83f2-82ec09f5843b.ericsson.com
18.144.83.42,38408266-5d7e-4aa6-95e5-32e8fc0948b2.ericsson.com
118.84.225.111,113d0029-be01-441b-8b04-95ff4b619499.ericsson.com
```

The first element is an IP address and the second element is a domain name.

Expected output

Your job is to load all the input files and to use the information from the lookup file to replace the

IP address from the filen.txt files with their corresponding domain as per the data provided in the lookup file.

For example, with the samples showed in the above section, the output record for the first element of the input file should be saved as:

```
961ccf9e-6928-4560-b811-0d48b13df1a5,38408266-5d7e-4aa6-95e5-32e8fc0948b2.ericsson.com
```

Once all the data has been transformed as per this rule, the result should be saved to the `/tmp/gaia-spark-test/output` directory

Furthermore, the application must correctly handle errors:
- some records may not be properly parsed (not following the described format)
- some IP addresses may not be looked-up in the lookup file (the ip would be missing)

These potential errors must also be accounted for and upon completion, the application needs to print out:
- the total amount of records that were read from the input file (including invalid ones)
- the amount of records that could not be parsed
- the amount of records for which the IP address could not be looked-up

The application should scale to:
- Terabytes of filen.txt files
- Up to 200 MiB of lookup data in ip-lookups.txt

In this exercise, you may assume that both the driver and executor processes can read or write to `/tmp/gaia-spark-test`

Provided file

With this instruction, you are also given a tar.gz archive file to get you started, the archive contains some sample input data as well as an empty scala sbt project you can modify with your solution. If you´d rather use Java or Python, feel free to start from an empty project

```
$ tree test
test
├── scala
│   ├── build.sbt
│   ├── project
│   │   └── build.properties
│   └── src
│       └── main
│           └── scala
│               └── com
│                   └── ericsson
│                       └── gaia
│                           └── spark
│                               └── tests
│                                   └── datamanipulation
│                                       └── SparkTestApplication.scala
└── tmp
    └── gaia-spark-test
        ├── file1.txt
        ├── file2.txt
        ├── file3.txt
```

```
        ├── file4.txt
        ├── file5.txt
        ├── file6.txt
        ├── file7.txt
        ├── file8.txt
        └── ip-lookups.txt
```