Hadoop streaming

Uruchomienie joba hadoop streaming:

1. skopiuj skrypty mapper.py i reducer.py na adminsk01
2. Zaloguj się na adminsk01
3. uruchomienie joba

```
hadoop jar <sciezka_do_jara_hadoop_streaming> -files <lista_plikow_z_mapperem_i_reducerem> \
  -mapper <plik_z_mapperem> -reducer <plik_z_reducerem> -input <katalog_wejsciowy> -output <katalog_wyjsciowy>

hadoop jar /opt/cloudera/parcels/CDH/jars/hadoop-streaming-2.6.0-cdh5.14.0.jar -files mapper.py,reducer.py \
  -mapper mapper.py -reducer reducer.py -input /user/xyz/loremipsum -output /user/xyz/outputs/output-2
```

Uwaga: Żeby polecenie się wykonało <katalog_wyjsciowy> nie może istnieć!

Zadania

1. policz literki w tekscie loremipsum
2. posortuj policzone literki po ilości wystąpień. Dlaczego jest to trudniejsze niż przy wykorzystaniu javovego api?
3. dla każdego konta policz ile było unikalnych numerów kont, z których wysłano przelewy na to konto
4. policz ile było unikalnych numerów kont w ogóle
5. Zadanie dodatkowe: napisz mapper i reducer w C# i użyj w jobie hadoop streaming

Przydatne parametry

Włączenie kompresji:

```
-D mapreduce.output.fileoutputformat.compress=true \
  -D mapreduce.output.fileoutputformat.compress.codec=org.apache.hadoop.io.compress.GzipCodec
```

Użycie innego input formatu:

```
-inputformat org.apache.hadoop.mapred.SequenceFileInputFormat
```

Użycie innego output formatu:

```
-outputformat org.apache.hadoop.mapred.SequenceFileOutputFormat
```

Identity mapper:

```
-mapper org.apache.hadoop.mapred.lib.IdentityMapper
```

Dokumentacja:
`https://hadoop.apache.org/docs/r1.2.1/streaming.html`