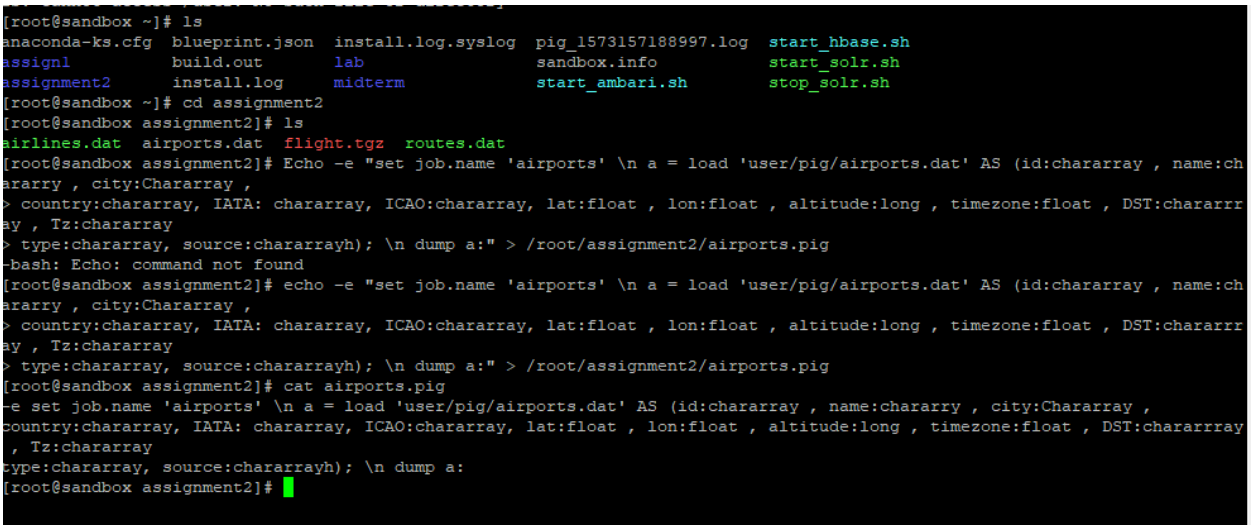
echo –e "set job.name 'airports' \n a = load 'user/pig/airports.dat' AS (id:chararray , name:chararry , city:Chararray ,country:chararray , IATA: chararray, ICAO:chararray, lat:float , lon:float , altitude:long , timezone:float , DST:chararrray , Tz:chararray, type:chararray, source:chararrayh); \n dump a:" >/root/assignment2/airports.pig
cat airports.pig

set job.name 'airports
a = load '/user/pig/airports.dat' AS (id:chararray, name:chararry, city:Chararray, country:chararray, IATA: chararray, ICAO:chararray, lat:float , lon:float, altitude:long, timezone:float, DST:chararrray, Tz:chararray
type:chararray, source:chararrayh);
dump a;

```
login as: root
root@127.0.0.1's password:
Last login: Sat Nov  9 21:01:38 2019 from 10.0.2.2
[root@sandbox ~]# cd /root/assignment2
[root@sandbox assignment2]# ls
flight.tgz
[root@sandbox assignment2]# tar -xvf flight.tgz
./._airlines.dat
airlines.dat
./._routes.dat
routes.dat
airports.dat
[root@sandbox assignment2]# ls
airlines.dat  airports.dat  flight.tgz  routes.dat
[root@sandbox assignment2]# hadoop fs -ls
Found 3 items
drwx------   - root hdfs          0 2019-11-10 03:55 .Trash
drwxr-xr-x   - root hdfs          0 2019-10-08 14:09 .hiveJars
drwx------   - root hdfs          0 2019-10-26 15:44 .staging
[root@sandbox assignment2]# hadoop fs -ls /user/pig
Found 2 items
-rw-r--r--   3 root hdfs   57135918 2019-10-26 15:01 /user/pig/full_text.txt
drwxr-xr-x   - root hdfs          0 2019-10-26 15:44 /user/pig/full_text_1.txt
[root@sandbox assignment2]# hadoop fs -put *.dat /user/pig
[root@sandbox assignment2]# hadoop fs -ls /user/pig
Found 5 items
-rw-r--r--   3 root hdfs     321974 2019-11-11 02:43 /user/pig/airlines.dat
-rw-r--r--   3 root hdfs     943570 2019-11-11 02:43 /user/pig/airports.dat
-rw-r--r--   3 root hdfs   57135918 2019-10-26 15:01 /user/pig/full_text.txt
drwxr-xr-x   - root hdfs          0 2019-10-26 15:44 /user/pig/full_text_1.txt
-rw-r--r--   3 root hdfs    2377148 2019-11-11 02:43 /user/pig/routes.dat
[root@sandbox assignment2]#
```


[root@sandbox ~]# ls
anaconda-ks.cfg blueprint.json install.log.syslog pig_1573157188997.log start_hbase.sh
assign1   build.out   lab   sandbox.info   start_solr.sh
assignment2   install.log   midterm   start_ambari.sh   stop_solr.sh
[root@sandbox ~]# cd assignment2
[root@sandbox assignment2]# ls
airlines.dat airports.dat flight.tgz routes.dat
[root@sandbox assignment2]# Echo -e "set job.name 'airports' \n a = load 'user/pig/airports.dat' AS (id:chararray , name:ch
ararry , city:Chararray ,
> country:chararray, IATA: chararray, ICAO:chararray, lat:float , lon:float , altitude:long , timezone:float , DST:chararrr
ay , Tz:chararray
> type:chararray, source:chararrayh); \n dump a:" > /root/assignment2/airports.pig
-bash: Echo: command not found
[root@sandbox assignment2]# echo -e "set job.name 'airports' \n a = load 'user/pig/airports.dat' AS (id:chararray , name:ch
ararry , city:Chararray ,
> country:chararray, IATA: chararray, ICAO:chararray, lat:float , lon:float , altitude:long , timezone:float , DST:chararrr
ay , Tz:chararray
> type:chararray, source:chararrayh); \n dump a:" > /root/assignment2/airports.pig
[root@sandbox assignment2]# cat airports.pig
-e set job.name 'airports' \n a = load 'user/pig/airports.dat' AS (id:chararray , name:chararry , city:Chararray ,
country:chararray, IATA: chararray, ICAO:chararray, lat:float , lon:float , altitude:long , timezone:float , DST:chararrray
 , Tz:chararray
type:chararray, source:chararrayh); \n dump a:
[root@sandbox assignment2]#

1) (2 pts) List the Airline_ID and name of all airlines where the name includes "Air Canada".
You search should be non-case sensitive and include "Air Canada" with or without the spaces.

a = load '/user/pig/airlines.dat' USING PigStorage(',') AS (airline_id:long , name:chararray , alias:chararray , IATA:chararray , ICAO:chararray, callsign:chararray , country:chararray , active:chararray);
b = foreach a generate airline_id,LOWER(name) as name;
c = filter b by (name matches '.*air.*canada.*');
dump c;

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.1.2.4.0.0-169       0.15.0.2.4.0.0-169      root    2019-11-13 00:46:10     2019-11-13 00:46:41     FILTER

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgR
educeTime       MedianReducetime        Alias   Feature Outputs
job_1573605810403_0001  1       0       5       5       5       5       0       0       0       0       a,b,c   MAP_ONLY   h
dfs://sandbox.hortonworks.com:8020/tmp/temp2064621586/tmp-899524364,

Input(s):
Successfully read 6162 records (322358 bytes) from: "/user/pig/airlines.dat"

Output(s):
Successfully stored 5 records (135 bytes) in: "hdfs://sandbox.hortonworks.com:8020/tmp/temp2064621586/tmp-899524364"

Counters:
Total records written : 5
Total bytes written : 135
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1573605810403_0001


2019-11-13 00:46:41,279 [main] INFO  org.apache.hadoop.yarn.client.api.impl.TimelineClientImpl - Timeline service address: h
ttp://sandbox.hortonworks.com:8188/ws/v1/timeline/
2019-11-13 00:46:41,280 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at sandbox.horton
works.com/10.0.2.15:8050
2019-11-13 00:46:41,289 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalA
pplicationStatus=SUCCEEDED. Redirecting to job history server
2019-11-13 00:46:41,482 [main] INFO  org.apache.hadoop.yarn.client.api.impl.TimelineClientImpl - Timeline service address: h
ttp://sandbox.hortonworks.com:8188/ws/v1/timeline/
2019-11-13 00:46:41,482 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at sandbox.horton
works.com/10.0.2.15:8050
2019-11-13 00:46:41,493 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalA
pplicationStatus=SUCCEEDED. Redirecting to job history server
2019-11-13 00:46:41,685 [main] INFO  org.apache.hadoop.yarn.client.api.impl.TimelineClientImpl - Timeline service address: h
ttp://sandbox.hortonworks.com:8188/ws/v1/timeline/
2019-11-13 00:46:41,685 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at sandbox.horton
works.com/10.0.2.15:8050
2019-11-13 00:46:41,696 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalA
pplicationStatus=SUCCEEDED. Redirecting to job history server
2019-11-13 00:46:41,771 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Succes
s!
2019-11-13 00:46:41,776 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not
generate code.
2019-11-13 00:46:41,794 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process :
1
2019-11-13 00:46:41,795 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to pr
ocess : 1
(330,air canada)
(341,air north charter - canada)
(983,air canada jazz)
(2442,fortunair canada)
(19675,rainbow air canada)
```

2) (2 pts) Find the number of airports in each country.
Submit the first five countries with the highest number of airports, together with the country names.

```
a = load '/user/pig/airports.dat' using PigStorage(',') AS
(airport_id: chararray,
name:chararray,city:chararray,country:chararray,IATA:chararray,ICAO:chararray,lat:floa
t,long:float,altitude:long,timezone:float,DST:chararray,Tzdata:chararray,type:chararray,s
ource:chararray);
b = group a by country;
c = foreach b generate group as country,COUNT(a) as cnt;
d = order c by cnt desc;
e = limit d 5;
dump e;
```

```
2019-11-13 00:53:02,055 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher -
s!
2019-11-13 00:53:02,057 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... wil
generate code.
2019-11-13 00:53:02,069 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to proc
1
2019-11-13 00:53:02,069 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths
ocess : 1
(United States,1099)
(Mexico,440)
(United Kingdom,413)
(Canada,323)
(Russia,238)
```

3) (4 pts) Find the distinct routes between airports, based on source and destination airports. Submit the first five rows.

```
a = load '/user/pig/routes.dat' using PigStorage(',') AS (airline:
chararray,
ID:long,source:chararray,sid:long,dest:chararray,did:chararray,codeshare:chararray,direct:int,equip:long
);
b = foreach a generate source,dest;
c = distinct b;
d = limit c 5;
dump d;
```

```
Counters:
Total records written : 5
Total bytes written : 80
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1573605810403_0012  ->      job_1573605810403_0013,
job_1573605810403_0013


2019-11-13 02:06:50,937 [main] INFO  org.apache.hadoop.yarn.client.api.impl.TimelineClientImpl - Timeline service address: h
ttp://sandbox.hortonworks.com:8188/ws/v1/timeline/
2019-11-13 02:06:50,938 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at sandbox.horton
works.com/10.0.2.15:8050
2019-11-13 02:06:50,946 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalA
pplicationStatus=SUCCEEDED. Redirecting to job history server
2019-11-13 02:06:51,136 [main] INFO  org.apache.hadoop.yarn.client.api.impl.TimelineClientImpl - Timeline service address: h
ttp://sandbox.hortonworks.com:8188/ws/v1/timeline/
2019-11-13 02:06:51,136 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at sandbox.horton
works.com/10.0.2.15:8050
2019-11-13 02:06:51,144 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalA
pplicationStatus=SUCCEEDED. Redirecting to job history server
2019-11-13 02:06:51,288 [main] INFO  org.apache.hadoop.yarn.client.api.impl.TimelineClientImpl - Timeline service address: h
ttp://sandbox.hortonworks.com:8188/ws/v1/timeline/
2019-11-13 02:06:51,288 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at sandbox.horton
works.com/10.0.2.15:8050
2019-11-13 02:06:51,296 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalA
pplicationStatus=SUCCEEDED. Redirecting to job history server
2019-11-13 02:06:51,452 [main] INFO  org.apache.hadoop.yarn.client.api.impl.TimelineClientImpl - Timeline service address: h
ttp://sandbox.hortonworks.com:8188/ws/v1/timeline/
2019-11-13 02:06:51,452 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at sandbox.horton
works.com/10.0.2.15:8050
2019-11-13 02:06:51,460 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalA
pplicationStatus=SUCCEEDED. Redirecting to job history server
2019-11-13 02:06:51,638 [main] INFO  org.apache.hadoop.yarn.client.api.impl.TimelineClientImpl - Timeline service address: h
ttp://sandbox.hortonworks.com:8188/ws/v1/timeline/
2019-11-13 02:06:51,639 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at sandbox.horton
works.com/10.0.2.15:8050
2019-11-13 02:06:51,649 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalA
pplicationStatus=SUCCEEDED. Redirecting to job history server
2019-11-13 02:06:51,860 [main] INFO  org.apache.hadoop.yarn.client.api.impl.TimelineClientImpl - Timeline service address: h
ttp://sandbox.hortonworks.com:8188/ws/v1/timeline/
2019-11-13 02:06:51,861 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at sandbox.horton
works.com/10.0.2.15:8050
2019-11-13 02:06:51,870 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalA
pplicationStatus=SUCCEEDED. Redirecting to job history server
2019-11-13 02:06:51,923 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Succes
s!
2019-11-13 02:06:51,924 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not
generate code.
2019-11-13 02:06:51,935 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process :
1
2019-11-13 02:06:51,935 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to pr
ocess : 1
(AAE,ALG)
(AAE,CDG)
(AAE,IST)
(AAE,LYS)
(AAE,MRS)
```

4) (7 pts) Generate a table with source airport ID, source airport name, destination airport id, destination
airport name and distance in kilometres using the output from the previous question.

Save your output in a tab separated file in an HDFS directory named 'routes_with_distances'.
Submit the screenshot of the directory listing and the first five lines of your output file.

Remember that you will have to get the latitude and longitude of each airport,
using two joins-one for source and one for destination airport.

Each degree of latitude and longitude (close to the equator) is roughly 111 km.
Calculate the distance in kilometres using the simple Euclidian formula:

distance = SQRT((lat2 - lat1) * (lat2 - lat1) + (lon2 - lon1) * (lon2 - lon1)) * 111

```
a = load '/user/pig/routes.dat' using PigStorage (',') AS (airline:chararray , ID:long , source:chararray , sid:chararray , dest:chararray , did:chararray , codeshare:chararray , direct:int , equip:long);
b = load '/user/pig/airports.dat' using PigStorage(',') AS (airport_id:chararray , name:chararray , city:chararray , country:chararray , IATA:chararray , ICAO:chararray , lat:float , lon:float , altitude:long , timezone:float , DST:chararray , Tzdata:chararray , type:chararray , src:chararray);
c = join a by sid, b by airport_id;
d = foreach c generate sid , source , lat as lats , lon as lons , dest , did;
e = join d by did , b by airport_id;
f = foreach e generate sid , source , lats , lons , did , dest , lat as latd , lon as lond;
g = foreach f generate sid , source , did , dest , SQRT((latd - lats) * (latd- lats) + (lond - lons) * (lond - lons)) * 111 as distance;
h = limit g 5;
STORE h into '/user/pig/routes_with_distances' using PigStorage('\t') ;
```

```
fs -ls/user/pig;
fs -ls /user/pig/routes_with_distances;
fs -cat /user/pig/routes_with_distances/part-r-00000;
```

```
grunt> fs -cat /user/pig/routes_with_distances/part-r-00000;
2       MAG     1       GKA     106.61514667864691
3       HGU     1       GKA     124.90211430866324
4       LAE     1       GKA     157.67350145135615
5       POM     1       GKA     424.7481116184964
5       POM     1       GKA     424.7481116184964
```