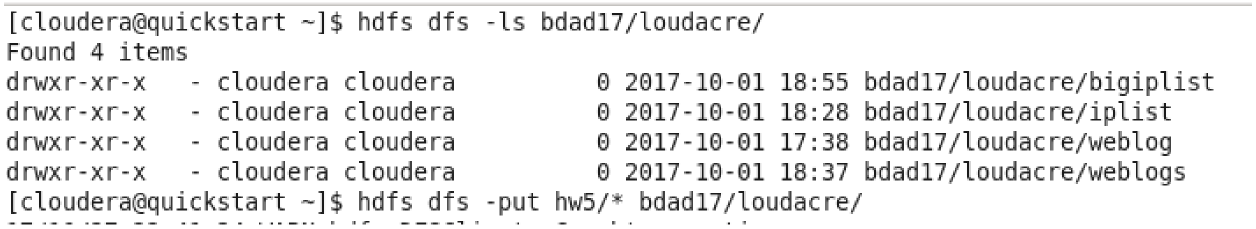
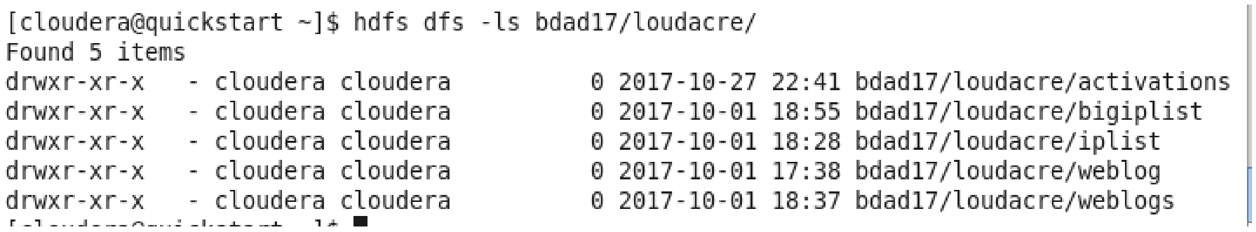
2.a) & 2.b)

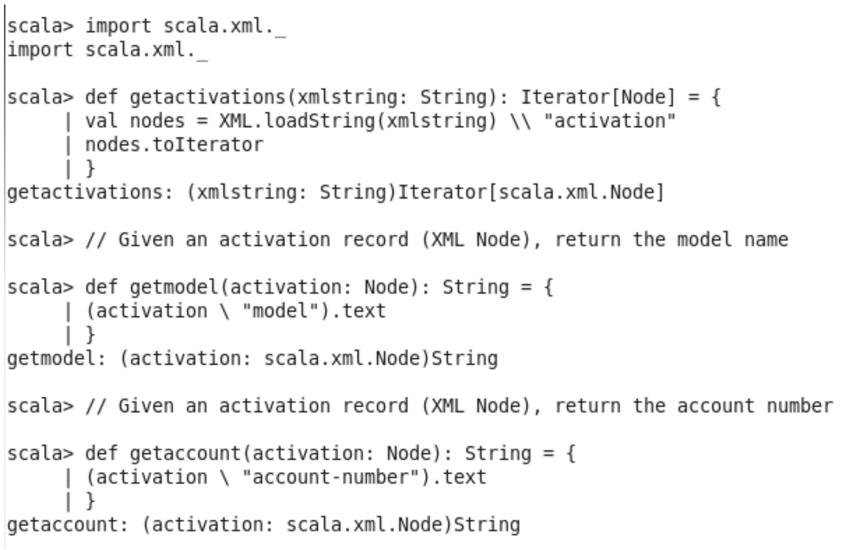


```
[cloudera@quickstart ~]$ hdfs dfs -ls bdad17/loudacre/
Found 4 items
drwxr-xr-x   - cloudera cloudera          0 2017-10-01 18:55 bdad17/loudacre/bigiplist
drwxr-xr-x   - cloudera cloudera          0 2017-10-01 18:28 bdad17/loudacre/iplist
drwxr-xr-x   - cloudera cloudera          0 2017-10-01 17:38 bdad17/loudacre/weblog
drwxr-xr-x   - cloudera cloudera          0 2017-10-01 18:37 bdad17/loudacre/weblogs
[cloudera@quickstart ~]$ hdfs dfs -put hw5/* bdad17/loudacre/
```

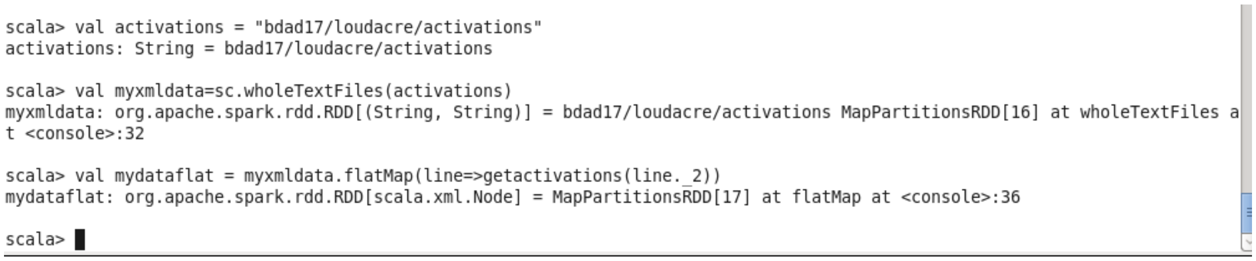


```
[cloudera@quickstart ~]$ hdfs dfs -ls bdad17/loudacre/
Found 5 items
drwxr-xr-x   - cloudera cloudera          0 2017-10-27 22:41 bdad17/loudacre/activations
drwxr-xr-x   - cloudera cloudera          0 2017-10-01 18:55 bdad17/loudacre/bigiplist
drwxr-xr-x   - cloudera cloudera          0 2017-10-01 18:28 bdad17/loudacre/iplist
drwxr-xr-x   - cloudera cloudera          0 2017-10-01 17:38 bdad17/loudacre/weblog
drwxr-xr-x   - cloudera cloudera          0 2017-10-01 18:37 bdad17/loudacre/weblogs
```

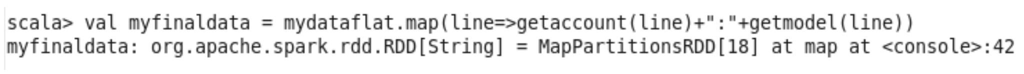
2.c)-f)



```
scala> import scala.xml._
import scala.xml._

scala> def getactivations(xmlstring: String): Iterator[Node] = {
     | val nodes = XML.loadString(xmlstring) \\ "activation"
     | nodes.toIterator
     | }
getactivations: (xmlstring: String)Iterator[scala.xml.Node]

scala> // Given an activation record (XML Node), return the model name

scala> def getmodel(activation: Node): String = {
     | (activation \ "model").text
     | }
getmodel: (activation: scala.xml.Node)String

scala> // Given an activation record (XML Node), return the account number

scala> def getaccount(activation: Node): String = {
     | (activation \ "account-number").text
     | }
getaccount: (activation: scala.xml.Node)String
```

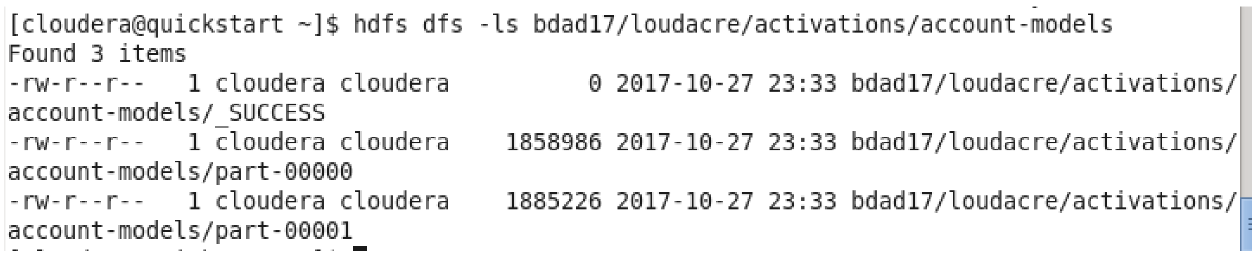


```
scala> val activations = "bdad17/loudacre/activations"
activations: String = bdad17/loudacre/activations

scala> val myxmldata=sc.wholeTextFiles(activations)
myxmldata: org.apache.spark.rdd.RDD[(String, String)] = bdad17/loudacre/activations MapPartitionsRDD[16] at wholeTextFiles a
t <console>:32

scala> val mydataflat = myxmldata.flatMap(line=>getactivations(line._2))
mydataflat: org.apache.spark.rdd.RDD[scala.xml.Node] = MapPartitionsRDD[17] at flatMap at <console>:36

scala> 
```



```
scala> val myfinaldata = mydataflat.map(line=>getaccount(line)+":"+getmodel(line))
myfinaldata: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[18] at map at <console>:42
```

```
scala> myfinaldata.saveAsTextFile("bdad17/loudacre/activations/account-models")
```



```
[cloudera@quickstart ~]$ hdfs dfs -ls bdad17/loudacre/activations/account-models
Found 3 items
-rw-r--r--   1 cloudera cloudera          0 2017-10-27 23:33 bdad17/loudacre/activations/
account-models/_SUCCESS
-rw-r--r--   1 cloudera cloudera    1858986 2017-10-27 23:33 bdad17/loudacre/activations/
account-models/part-00000
-rw-r--r--   1 cloudera cloudera    1885226 2017-10-27 23:33 bdad17/loudacre/activations/
account-models/part-00001
```

```
import scala.xml._

def getactivations(xmlstring: String): Iterator[Node] = {
val nodes = XML.loadString(xmlstring) \\ "activation"
nodes.toIterator
}
// Given an activation record (XML Node), return the model name
def getmodel(activation: Node): String = {
(activation \ "model").text
}
// Given an activation record (XML Node), return the account number
def getaccount(activation: Node): String = {
(activation \ "account-number").text
}

val activations = "bdad17/loudacre/activations"

val myxmldata=sc.wholeTextFiles(activations)

val mydataflat = myxmldata.flatMap(line=>getactivations(line._2))

val myfinaldata = mydataflat.map(line=>getaccount(line)+":"+getmodel(line))

myfinaldata.saveAsTextFile("bdad17/loudacre/activations/account-models")
```