

**CSE488: Big Data Analytics**

**[SPRING 2023]**

**Assignment**

**All Online Assignment**

**Submitted by:**

Student ID: Mujahidul Islam

Student Name: 2019-2-60-072

LAB 1

QUES 1

Bomb.createOrReplaceTempView("Bomb")

query2 = """

SELECT AirCraft,ContryFlyingMission,TargetCountry

FROM Bomb

WHERE TimeOnTarget< '1000'

"""

result2=spark.sql(query2)

result2.show()

QUES 2

query3 = """

SELECT TargetCountry, COUNT(TargetCountry) AS Attack\_time

FROM Bomb

GROUP BY TargetCountry

ORDER BY COUNT(TargetCountry) DESC

"""

result3=spark.sql(query3)

result3.show()

QUES 3

query3 = """

SELECT TargetCountry, COUNT(TargetCountry) AS Attack\_time

FROM Bomb

GROUP BY TargetCountry

ORDER BY COUNT(TargetCountry) DESC

"""

result3=spark.sql(query3)

result3.show()

result3.createOrReplaceTempView("result3")

query4="""

SELECT Attack\_time,TargetCountry

FROM result3

WHERE Attack\_time = (SELECT MAX(Attack\_time) FROM result3)

"""

result4=spark.sql(query4)

result4.show()

QUES4

query5 = """

SELECT ContryFlyingMission, COUNT(ContryFlyingMission) AS Most\_Attack\_time

FROM Bomb

GROUP BY ContryFlyingMission

ORDER BY COUNT(ContryFlyingMission) DESC

"""

result5=spark.sql(query5)

result5.createOrReplaceTempView("result5")

query6 = """

SELECT ContryFlyingMission, Most\_Attack\_time

FROM result5

WHERE Most\_Attack\_time = (SELECT MAX(Most\_Attack\_time) FROM result5 )

"""

result6=spark.sql(query6)

result6.show()

QUES 5

query7 = """

SELECT TakeoffLocation, COUNT(TakeoffLocation)

FROM Bomb

GROUP BY TakeoffLocation

"""

result7=spark.sql(query7)

result7.show()

QUES 6

query12 = """

SELECT COUNT(AirCraftType)

FROM Aircraft

WHERE AirCraftType == 'Fighter Jet'

"""

result12= spark.sql(query12)

result12.show()

QUES 7

Aircraft.createOrReplaceTempView("Aircraft")

query8="""

SELECT AirCraftType, COUNT(AirCraftType)

FROM Aircraft

GROUP BY AirCraftType

"""

result8=spark.sql(query8)

result8.show()

QUES 8

query9= """

SELECT AirCraftName, TargetCountry

FROM Bomb B, Aircraft A

WHERE B.AirCraft = A.AirCraft

"""

result9 = spark.sql(query9)

result9.show()

result9.createOrReplaceTempView("result9")

query10 = """

SELECT AirCraftName, COUNT(AirCraftName) AS Most\_used

FROM result9

GROUP BY AirCraftName

"""

result10 = spark.sql(query10)

result10.show()

result10.createOrReplaceTempView("result10")

query11 = """

SELECT AirCraftName,Most\_used

FROM result10

WHERE Most\_used==(SELECT MAX(Most\_used) FROM result10 )

"""

result11 = spark.sql(query11)

result11.show()

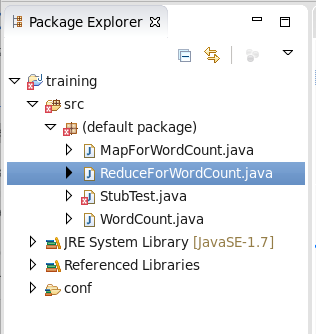
LAB2

LAB2

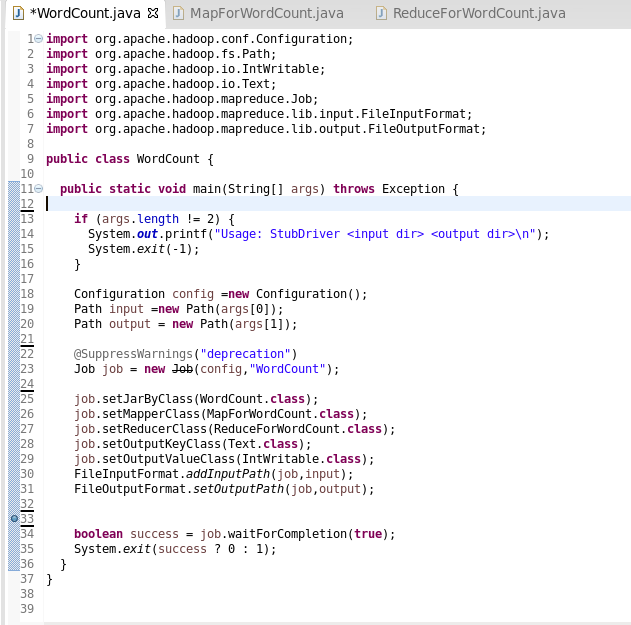
Showing the execution of the WordCount application in Hadoop MapReduce Environment step by step.

Step A: Writing Code

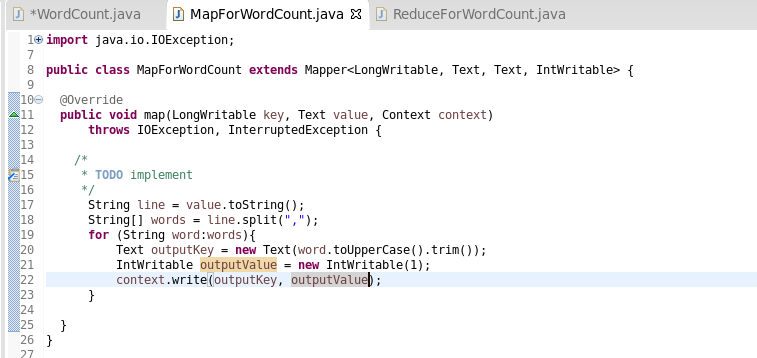
1. Launch Eclipse on Cloudera Environment.
2. Rename the file under /training/src/default package according to the file name given in image. [Remain the subtest as it is]



1. Write the code as it is-



Code for main class



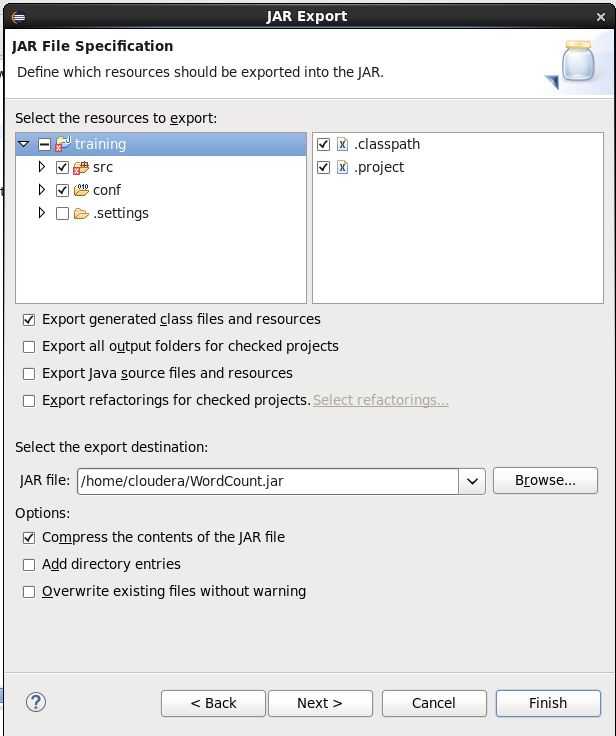
Code for Mapper Class



Code for Reducer Class

Step B: Creating Jar file

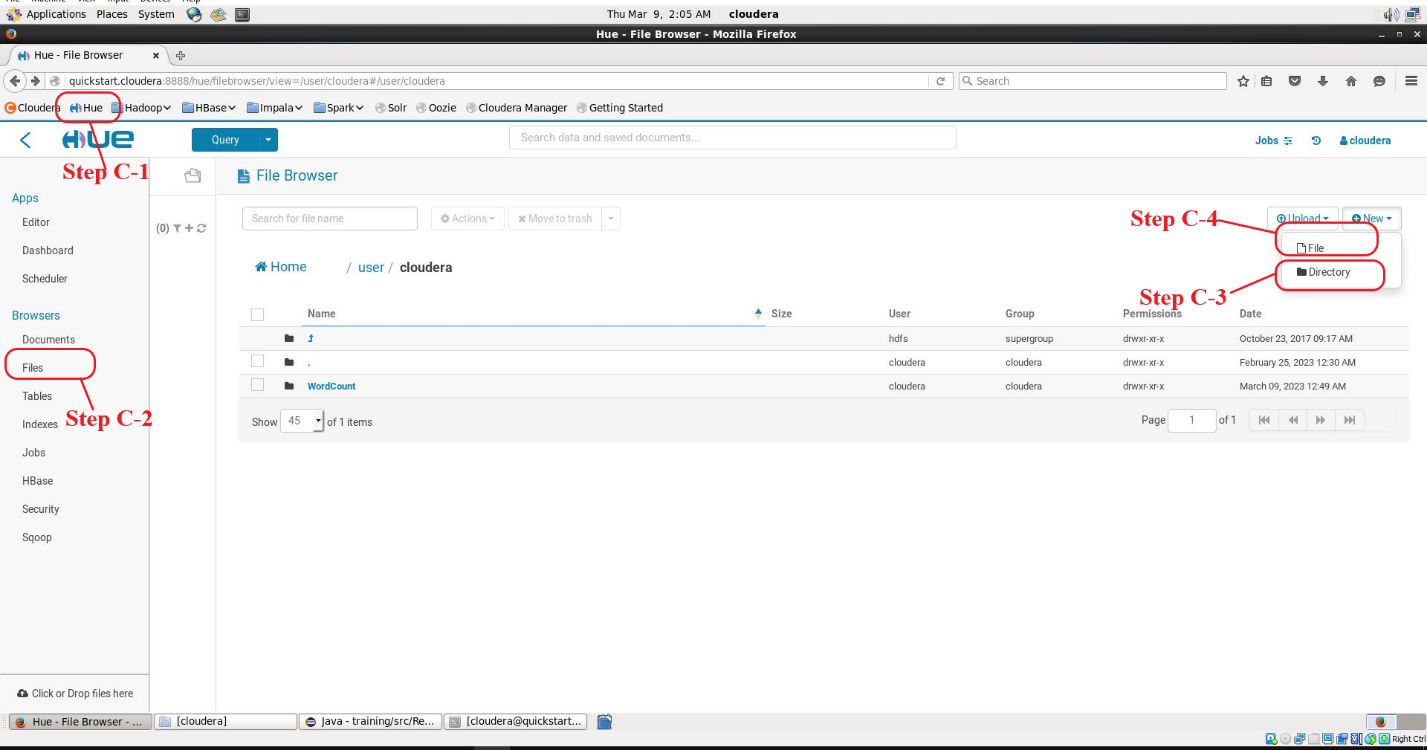
1. To create a jar file navigate through file🡪export. A window will pop-up. Select java🡪jar. Click next.
2. Make sure to check the resources to export where you need the jar file. Then create e destination and give file a name. (For instance WordCount.jar).

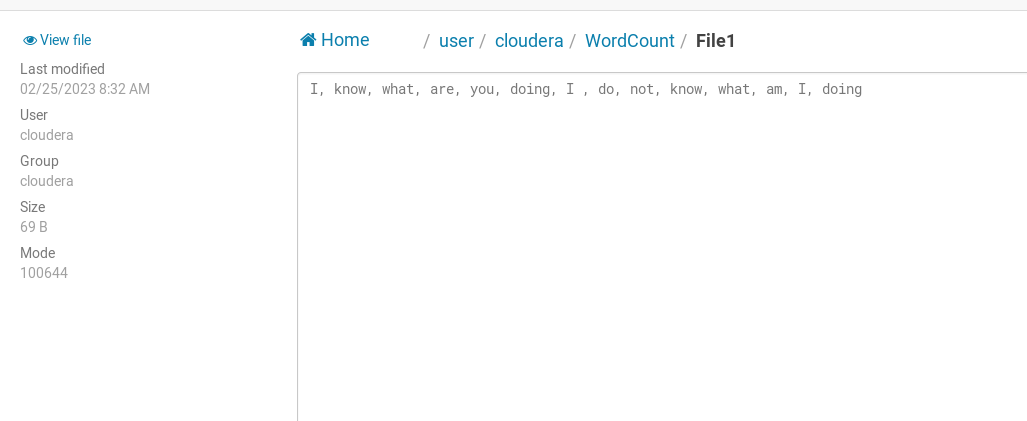


1. Click next🡪next🡪next🡪finish.

Step C: Creating Input file

1. Launch web browser (mozilla) on eclipse. Click on hue tab. [Note: You may ask the username and password. Both are ‘cloudera’]
2. Open navigation and open files tab.
3. Create a new directory. Name this according to you.
4. Under the directory create file
5. Open the file and edit it according to you. Make sure the file is comma separated. [Make sure to create multiple file]
6. Save the files.





Input file

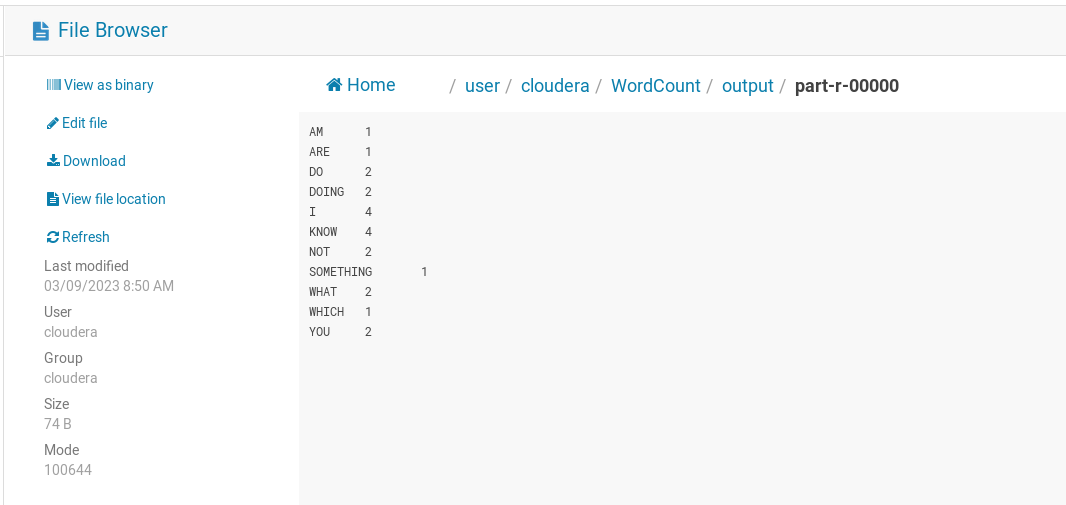
Step D: Executing the program through terminal

1. Open terminal from eclipse.
2. Execute this command line “Hadoop jar jarfilename.jar mainfilename inputfiledirectory outputfiledirectory/outputfilename”

For my programme I used this command “hadoop jar WordCount.jar WordCount /user/cloudera/WordCount /user/cloudera/WordCount/output”

1. Press enter. A output file should be generated in you declared output directory.

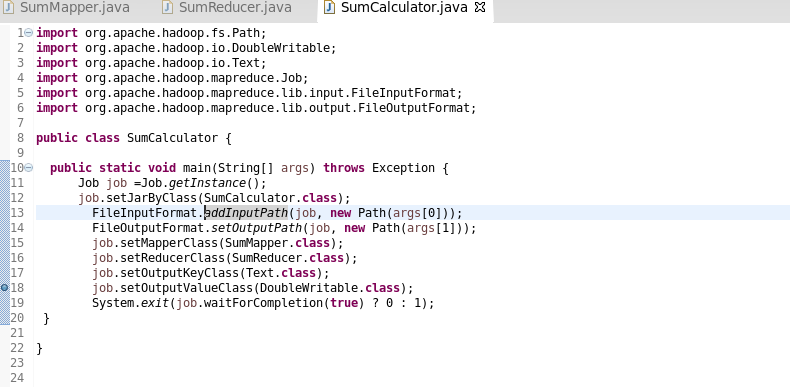




Output File

LAB3

Main Class code:



Mapper Class:

Graphical user interface, text, application, email

Description automatically generated

Reducer Class:

Graphical user interface, text, application, email

Description automatically generated

Input Files:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Execution Command Prompt:

Text

Description automatically generated

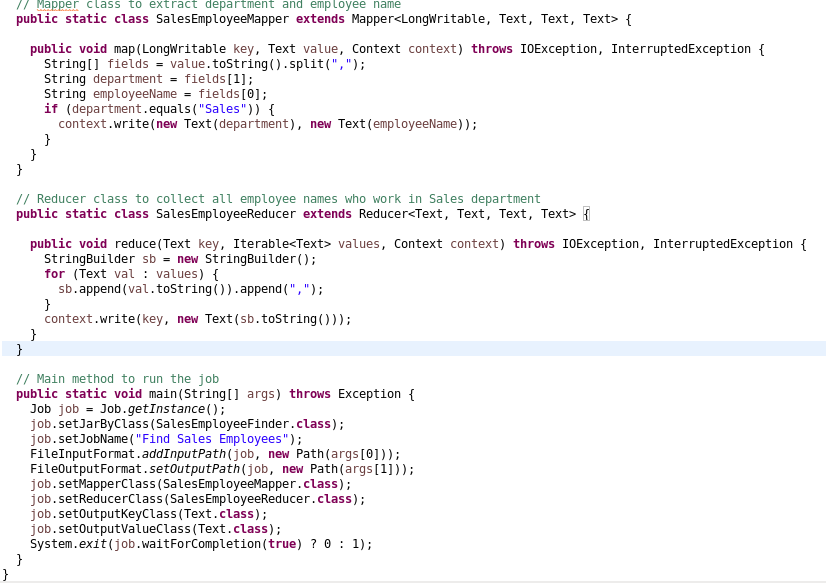
Output:

Graphical user interface, application

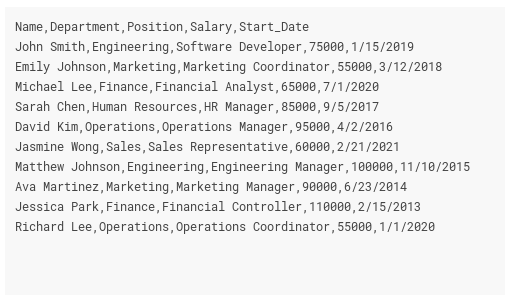
Description automatically generated

LAB4

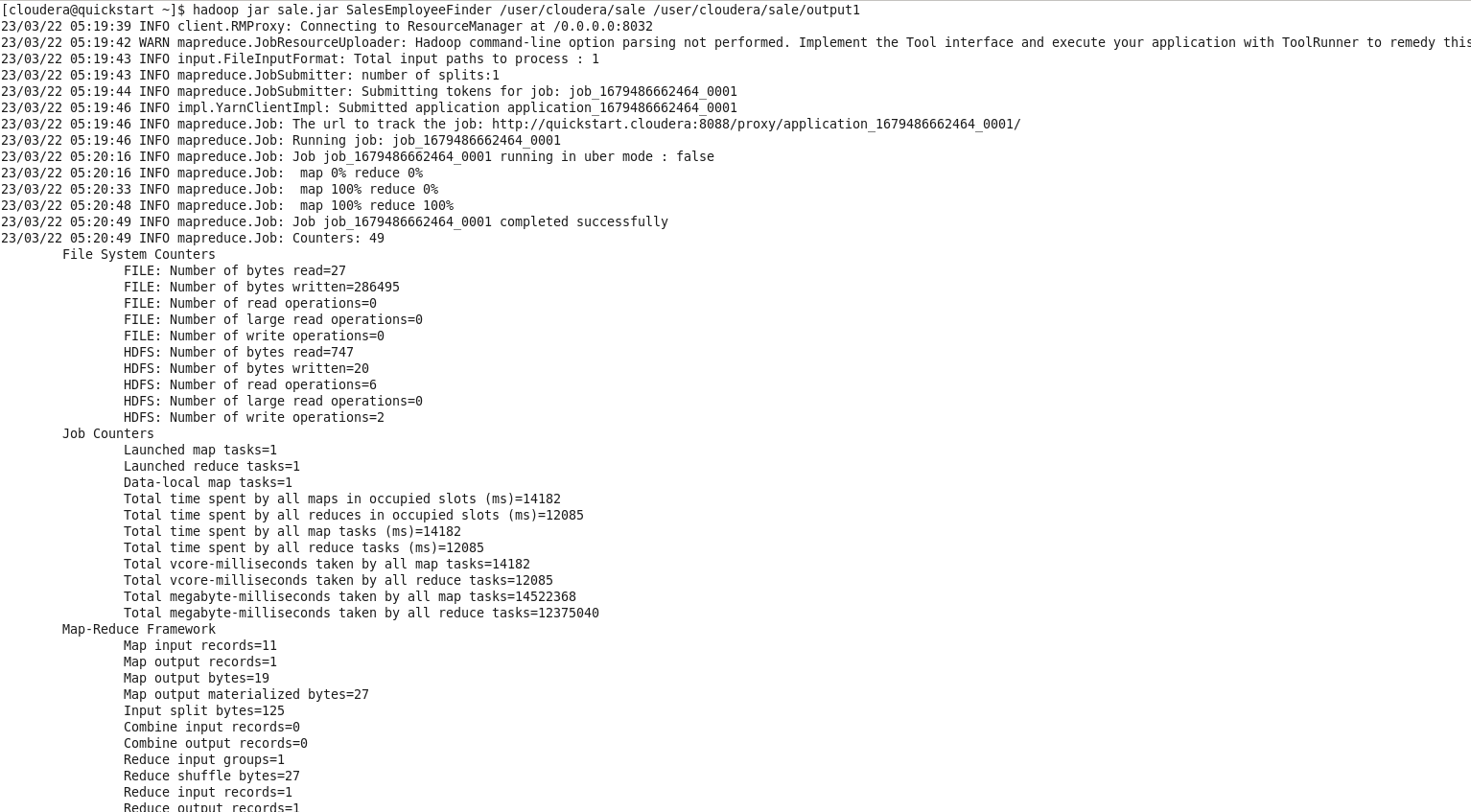
Code:



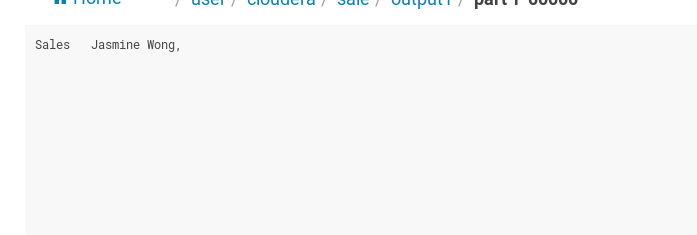
Input Files



Execution Command Prompt:

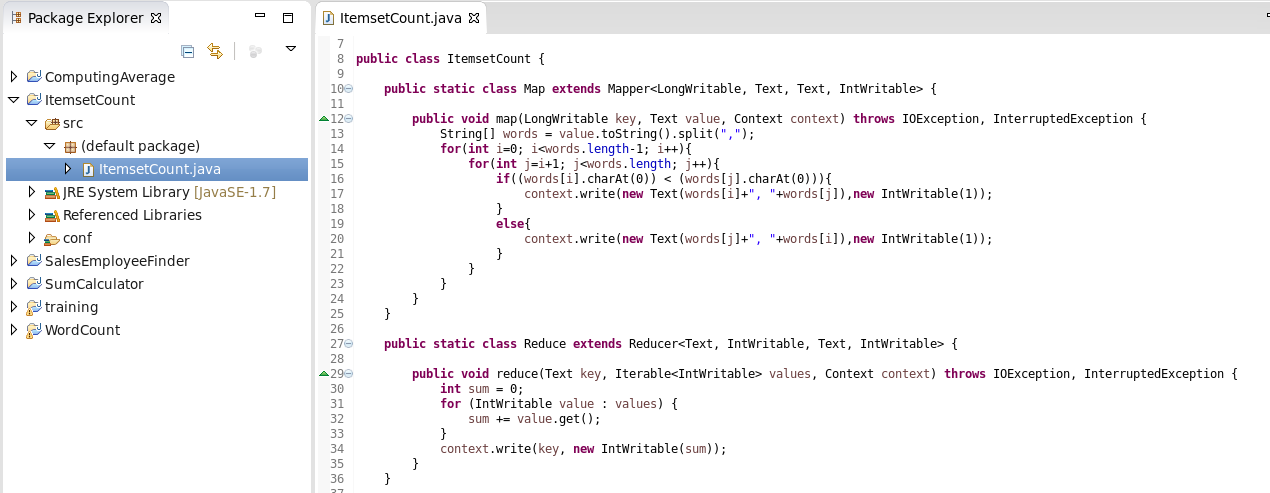


Output:

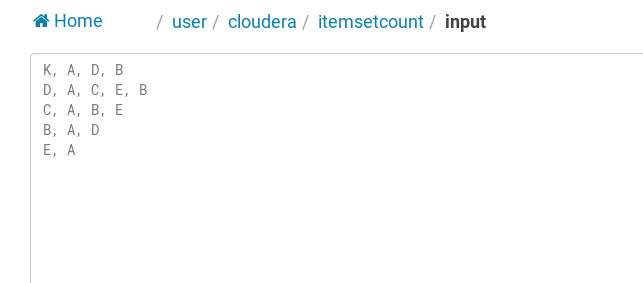


LAB5

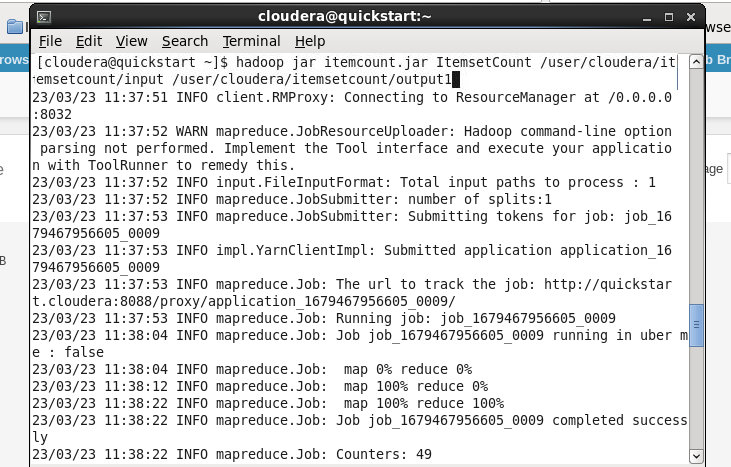
Code:



Input Files:



Executing Command Prompt:



Output:

