CSC14118

Introduction to Big Data

Group 03

Lab02: MapReduce Programming

Student ID	Full name
20127449	Trần Quốc Bảo
20127452	Hồ Đăng Cao
20127476	Đỗ Đức Duy

Project	Version	Date
PLAN001	v1.0	2023-07-22

CSC14118

Introduction to Big Data

REPORT

Project	Version	Date
PLAN001	v1.0	2023-07-22

Summary

Section	Completed percentage	Issues
S01	100%	
S02	100%	
S03	100%	
S04	100%	
S05	100%	
S06	100%	
S07	100%	
S08	100%	
S09	100%	
S10	10%	

CSC14118

Introduction to Big Data

REPORT

Project Version Date

PLAN001 v1.0 2023-07-22

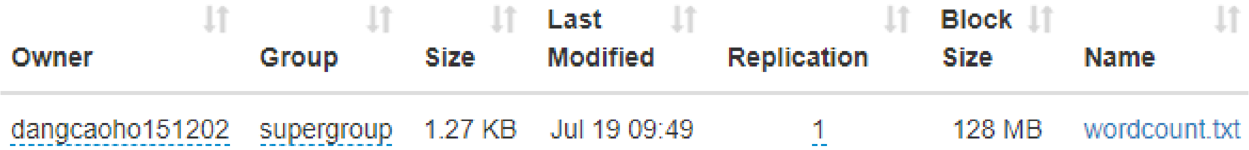
Contents

1 Wordcount Program

Note: The source code is taken from the source code provided at lab 01.

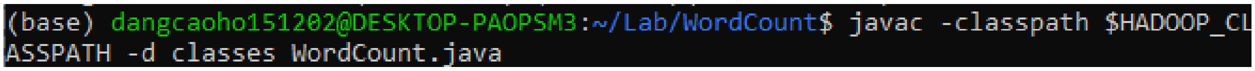
Step 1: Type the following command to export the hadoop classpath into bash.
export HADOOP_CLASSPATH=$(hadoop classpath)
echo $HADOOP_CLASSPATH

Step 2: Create directories on hdfs and put the input data file to hdfs.
hadoop fs -mkdir /WordCount
hadoop fs -mkdir /WordCount/Input
hadoop fs -put <input file’s path> /WordCount/Input



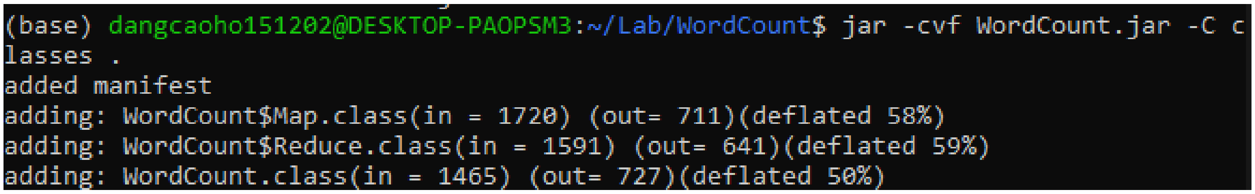
Owner	Group	Size	Last Modified	Replication	Block Size	Name
dangcaoho151202	supergroup	1.27 KB	Jul 19 09:49	1	128 MB	wordcount.txt

Step 3: Compile the WeatherData.java file
javac -classpath $HADOOP_CLASSPATH -d <classes folder’s path> <source’s path>



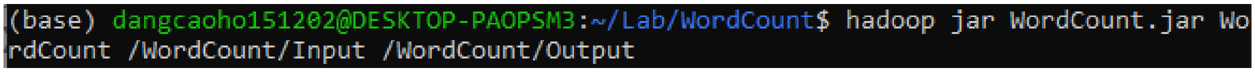
```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/WordCount$ javac -classpath $HADOOP_CL
ASSPATH -d classes WordCount.java
```

Step 4: Put the output files in a jar file.
jar -cvf <.jar file’s path> -C <classes folder’s path> .

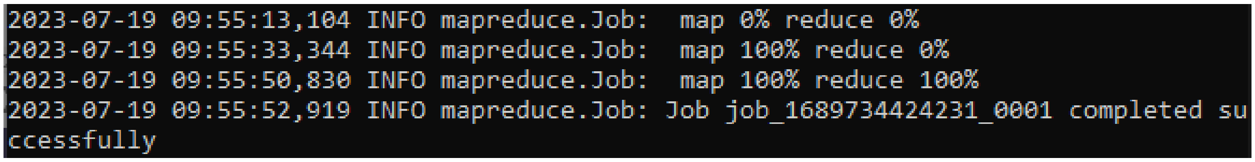


```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/WordCount$ jar -cvf WordCount.jar -C c
lasses .
added manifest
adding: WordCount$Map.class(in = 1720) (out= 711)(deflated 58%)
adding: WordCount$Reduce.class(in = 1591) (out= 641)(deflated 59%)
adding: WordCount.class(in = 1465) (out= 727)(deflated 50%)
```

Step 5: Run the jar file on Hadoop.
hadoop jar <.jar file’s path> WordCount /WordCount/Input /WordCount/Output

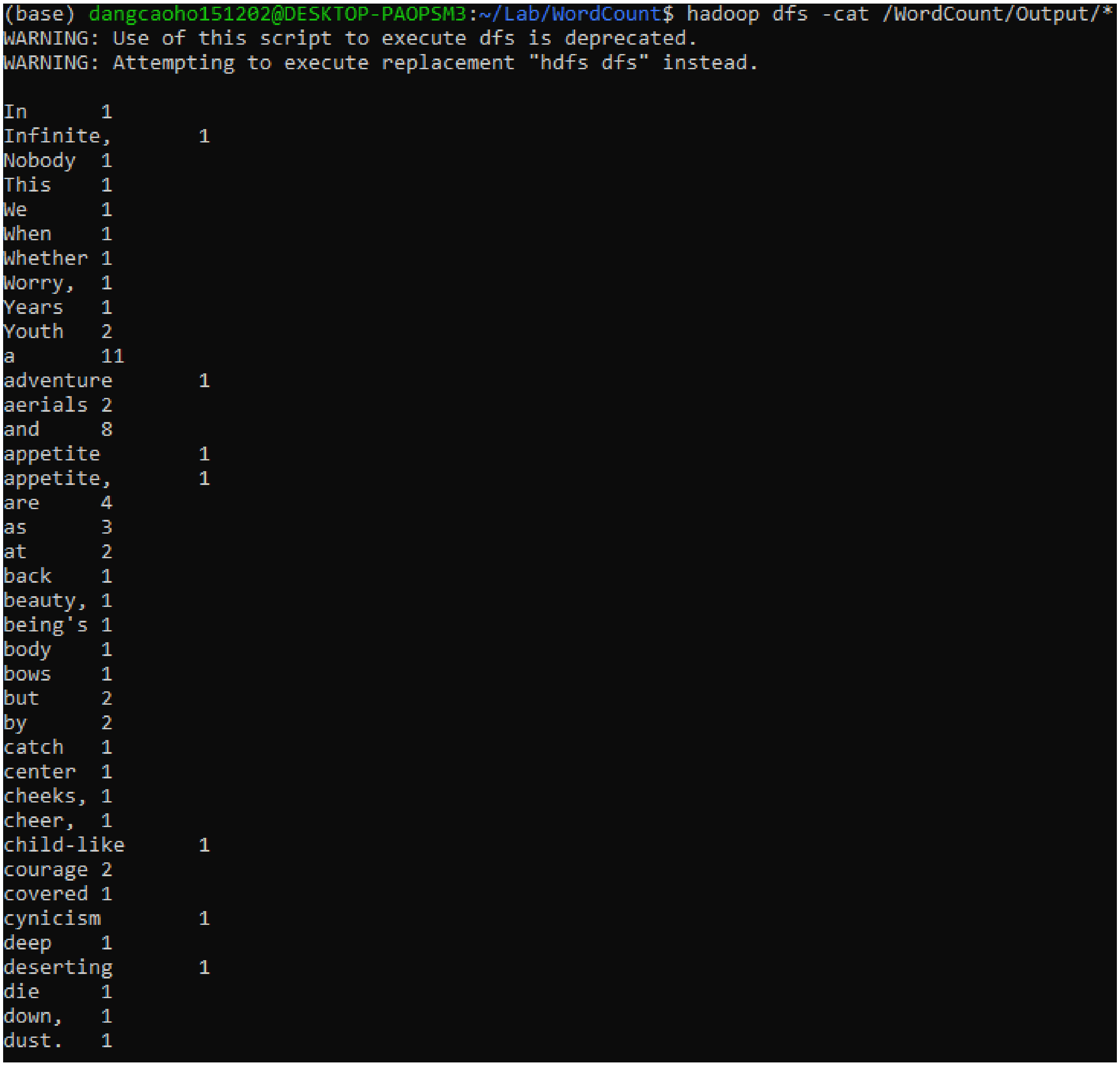


```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/WordCount$ hadoop jar WordCount.jar Wo
rdCount /WordCount/Input /WordCount/Output
```



```
2023-07-19 09:55:13,104 INFO mapreduce.Job:  map 0% reduce 0%
2023-07-19 09:55:33,344 INFO mapreduce.Job:  map 100% reduce 0%
2023-07-19 09:55:50,830 INFO mapreduce.Job:  map 100% reduce 100%
2023-07-19 09:55:52,919 INFO mapreduce.Job: Job job_1689734424231_0001 completed su
ccessfully
```

Step 6: Result
hadoop dfs -cat /WordCount/Output/*



```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/WordCount$ hadoop dfs -cat /WordCount/Output/*
WARNING: Use of this script to execute dfs is deprecated.
WARNING: Attempting to execute replacement "hdfs dfs" instead.

In      1
Infinite,       1
Nobody  1
This    1
We      1
When    1
Whether 1
Worry,  1
Years   1
Youth   2
a       11
adventure       1
aerials 2
and     8
appetite        1
appetite,       1
are     4
as      3
at      2
back    1
beauty, 1
being's 1
body    1
bows    1
but     2
by      2
catch   1
center  1
cheeks, 1
cheer,  1
child-like      1
courage 2
covered 1
cynicism        1
deep    1
deserting       1
die     1
down,   1
dust.   1
```

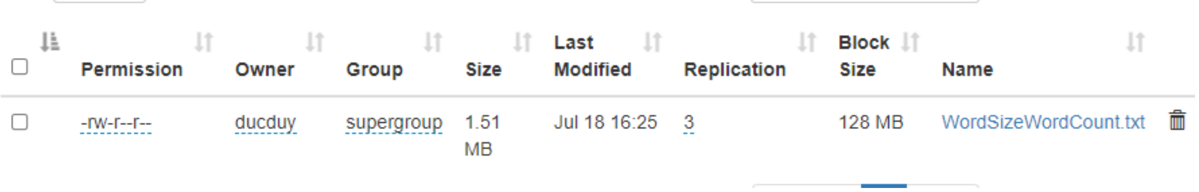
2 WordSizeWordCount Program

Note: the source code is based on the provided requirements file and link.

Step 1: Type the following command to export the hadoop classpath into bash.
export HADOOP_CLASSPATH=$(hadoop classpath)
echo $HADOOP_CLASSPATH

Step 2: Create directories on hdfs and put the input data file to hdfs.

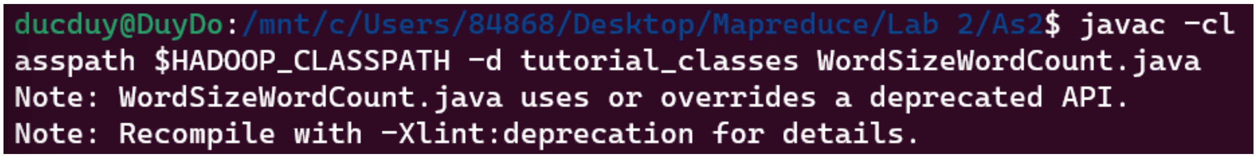
```
hadoop fs -mkdir /WordSizeWordCount
hadoop fs -mkdir /WordSizeWordCount/Input
hadoop fs -put <input file's path> /WordSizeWordCount/Input
```



□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
□	-rw-r--r--	ducduy	supergroup	1.51 MB	Jul 18 16:25	3	128 MB	WordSizeWordCount.txt	🗑

Step 3: Compile the WordSizeWordCount.java file

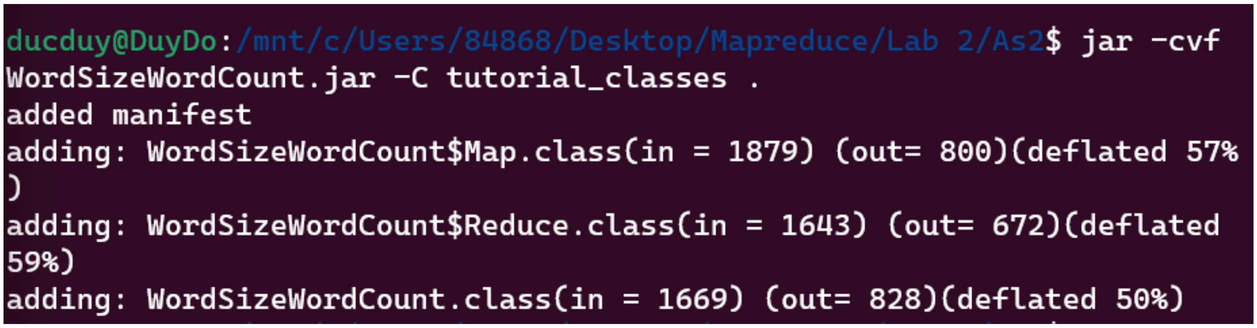
```
javac -classpath $HADOOP_CLASSPATH -d <classes folder's path> <source's path>
```



```
ducduy@DuyDo:/mnt/c/Users/84868/Desktop/Mapreduce/Lab 2/As2$ javac -cl
asspath $HADOOP_CLASSPATH -d tutorial_classes WordSizeWordCount.java
Note: WordSizeWordCount.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
```

Step 4: Put the output files in a jar file.

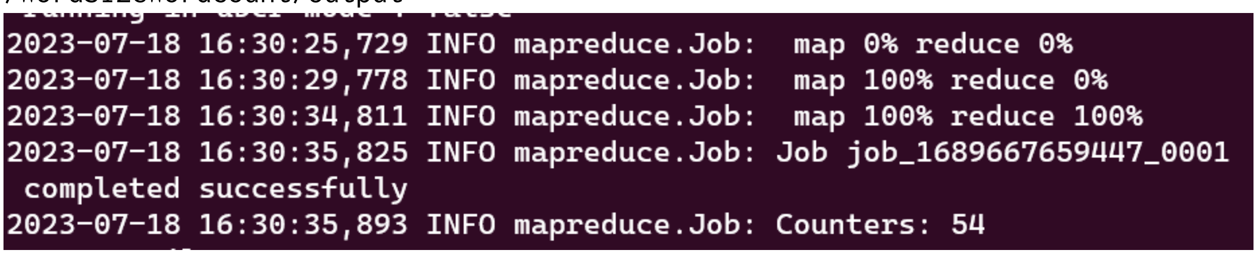
```
jar -cvf <.jar file's path> -C <classes folder's path> .
```



```
ducduy@DuyDo:/mnt/c/Users/84868/Desktop/Mapreduce/Lab 2/As2$ jar -cvf
WordSizeWordCount.jar -C tutorial_classes .
added manifest
adding: WordSizeWordCount$Map.class(in = 1879) (out= 800)(deflated 57%
)
adding: WordSizeWordCount$Reduce.class(in = 1643) (out= 672)(deflated
59%)
adding: WordSizeWordCount.class(in = 1669) (out= 828)(deflated 50%)
```

Step 5: Run the jar file on Hadoop.

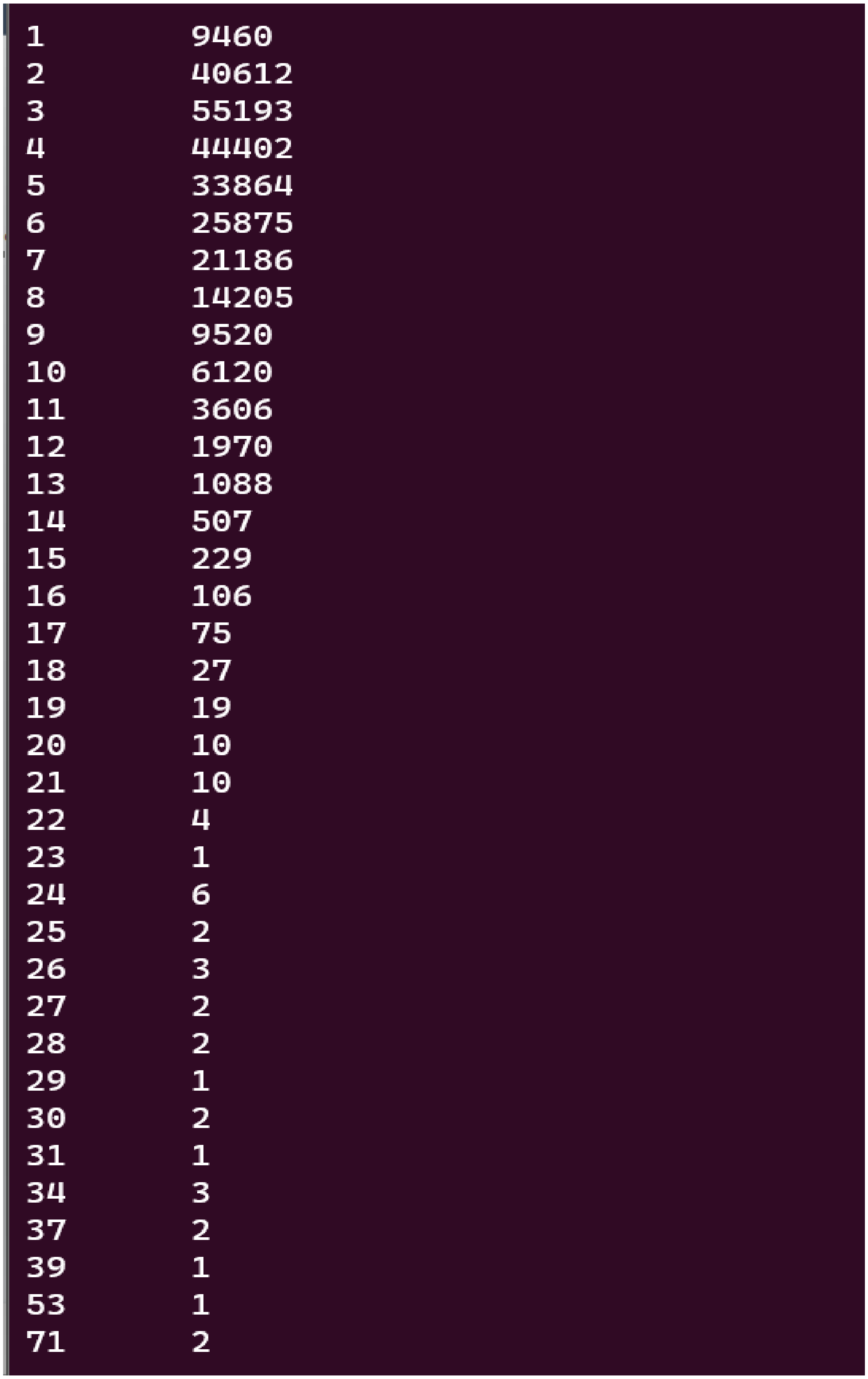
```
hadoop jar <.jar file's path> WordSizeWordCount /WordSizeWordCount/Input
/WordSizeWordCount/Output
```



```
2023-07-18 16:30:25,729 INFO mapreduce.Job:  map 0% reduce 0%
2023-07-18 16:30:29,778 INFO mapreduce.Job:  map 100% reduce 0%
2023-07-18 16:30:34,811 INFO mapreduce.Job:  map 100% reduce 100%
2023-07-18 16:30:35,825 INFO mapreduce.Job: Job job_1689667659447_0001
 completed successfully
2023-07-18 16:30:35,893 INFO mapreduce.Job: Counters: 54
```

Step 6: Result

```
hadoop dfs -cat /WordSizeWordCount/Output/*
```



```
1       9460
2       40612
3       55193
4       44402
5       33864
6       25875
7       21186
8       14205
9       9520
10      6120
11      3606
12      1970
13      1088
14      507
15      229
16      106
17      75
18      27
19      19
20      10
21      10
22      4
23      1
24      6
25      2
26      3
27      2
28      2
29      1
30      2
31      1
34      3
37      2
39      1
53      1
71      2
```

3 WeatherData Program

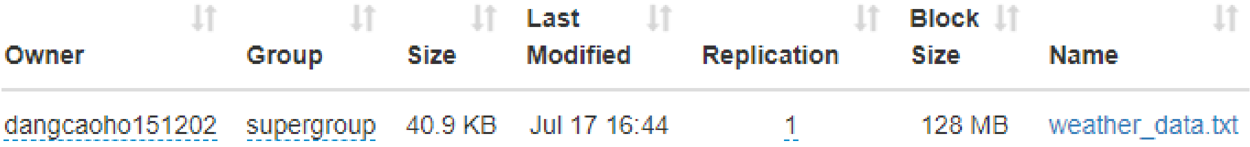
Note: the source code is referenced from the provided requirements file.

Step 1: Type the following command to export the hadoop classpath into bash.

```
export HADOOP_CLASSPATH=$(hadoop classpath)
echo $HADOOP_CLASSPATH
```

Step 2: Create directories on hdfs and put the input data file to hdfs.

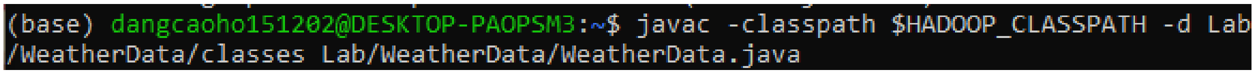
```
hadoop fs -mkdir /WeatherData
hadoop fs -mkdir /WeatherData/Input
hadoop fs -put <input file’s path> /WeatherData/Input
```



Owner	Group	Size	Last Modified	Replication	Block Size	Name
dangcaoho151202	supergroup	40.9 KB	Jul 17 16:44	1	128 MB	weather_data.txt

Step 3: Compile the WeatherData.java file

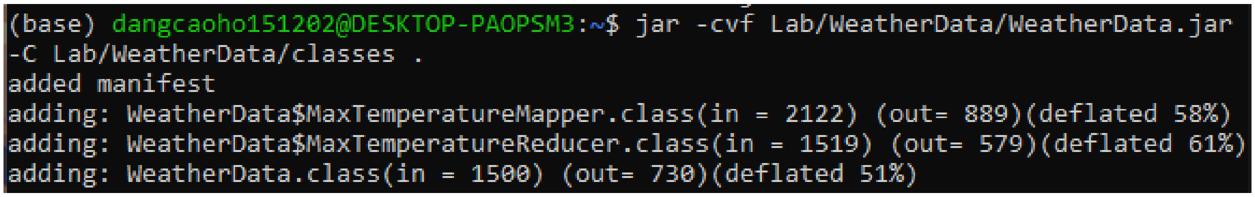
```
javac -classpath $HADOOP_CLASSPATH -d <classes folder’s path> <source’s path>
```



```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~$ javac -classpath $HADOOP_CLASSPATH -d Lab
/WeatherData/classes Lab/WeatherData/WeatherData.java
```

Step 4: Put the output files in a jar file.

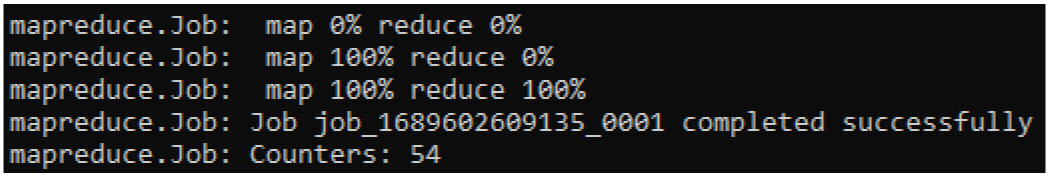
```
jar -cvf <.jar file’s path> -C <classes folder’s path> .
```



```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~$ jar -cvf Lab/WeatherData/WeatherData.jar
-C Lab/WeatherData/classes .
added manifest
adding: WeatherData$MaxTemperatureMapper.class(in = 2122) (out= 889)(deflated 58%)
adding: WeatherData$MaxTemperatureReducer.class(in = 1519) (out= 579)(deflated 61%)
adding: WeatherData.class(in = 1500) (out= 730)(deflated 51%)
```

Step 5: Run the jar file on Hadoop.

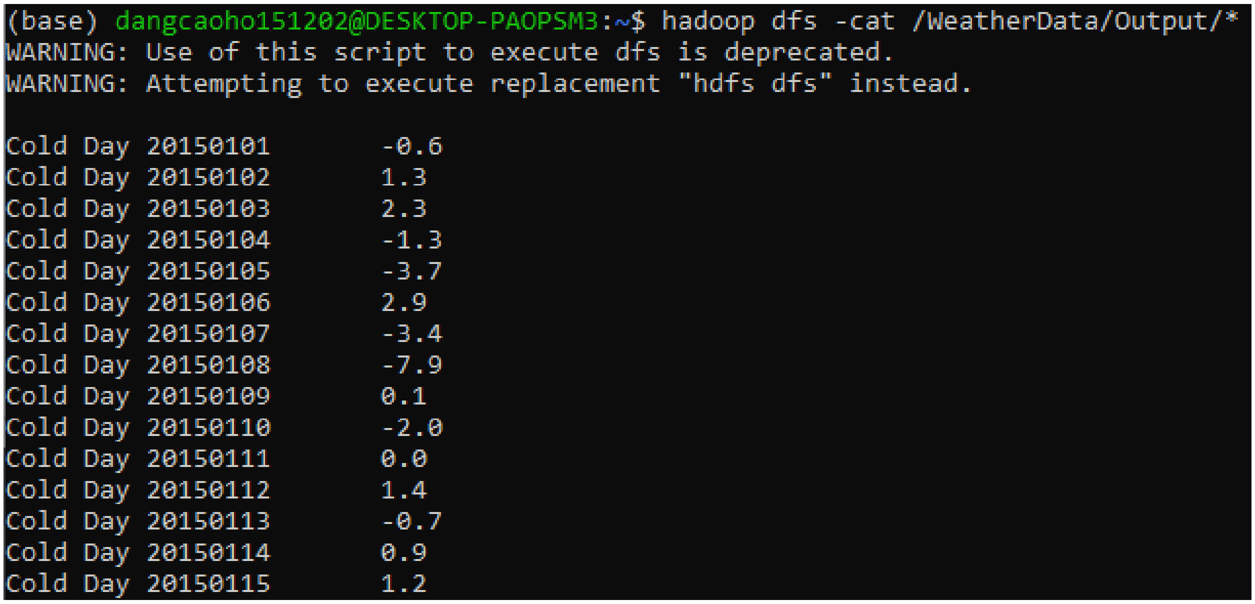
```
hadoop jar <.jar file’s path> WeatherData /WeatherData/Input
/WeatherData/Output
```



```
mapreduce.Job:  map 0% reduce 0%
mapreduce.Job:  map 100% reduce 0%
mapreduce.Job:  map 100% reduce 100%
mapreduce.Job: Job job_1689602609135_0001 completed successfully
mapreduce.Job: Counters: 54
```

Step 6: Result

```
hadoop dfs -cat /WeatherData/Output/*
```



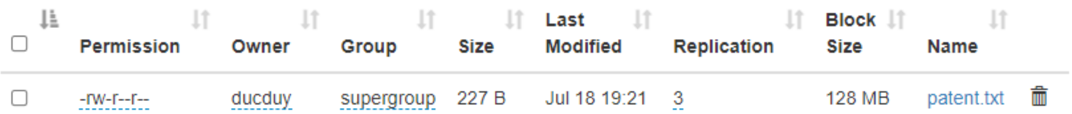
```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~$ hadoop dfs -cat /WeatherData/Output/*
WARNING: Use of this script to execute dfs is deprecated.
WARNING: Attempting to execute replacement "hdfs dfs" instead.

Cold Day 20150101       -0.6
Cold Day 20150102       1.3
Cold Day 20150103       2.3
Cold Day 20150104       -1.3
Cold Day 20150105       -3.7
Cold Day 20150106       2.9
Cold Day 20150107       -3.4
Cold Day 20150108       -7.9
Cold Day 20150109       0.1
Cold Day 20150110       -2.0
Cold Day 20150111       0.0
Cold Day 20150112       1.4
Cold Day 20150113       -0.7
Cold Day 20150114       0.9
Cold Day 20150115       1.2
```

4 Patent Program

Note: the source code is based on the provided requirements file and link.

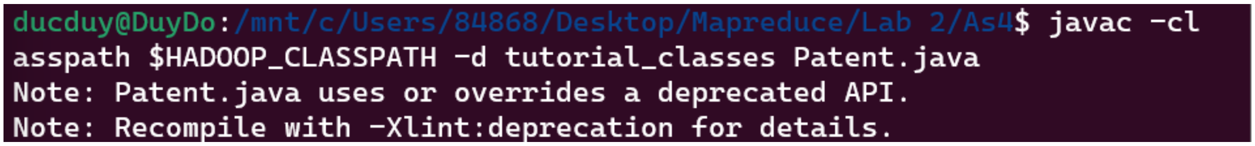
Step 1: Type the following command to export the hadoop classpath into bash.
export HADOOP_CLASSPATH=$(hadoop classpath)
echo $HADOOP_CLASSPATH

Step 2: Create directories on hdfs and put the input data file to hdfs.
hadoop fs -mkdir /Patent
hadoop fs -mkdir /Patent/Input
hadoop fs -put <input file's path> /Patent/Input



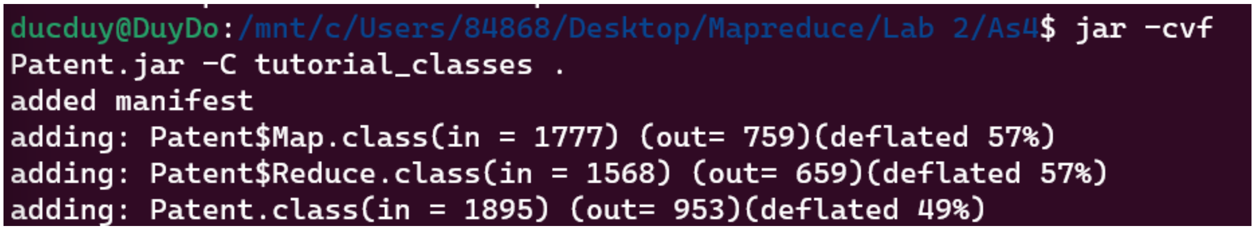
☐	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
☐	-rw-r--r--	ducduy	supergroup	227 B	Jul 18 19:21	3	128 MB	patent.txt	🗑

Step 3: Compile the Patent.java file
javac -classpath $HADOOP_CLASSPATH -d <classes folder's path> <source's path>



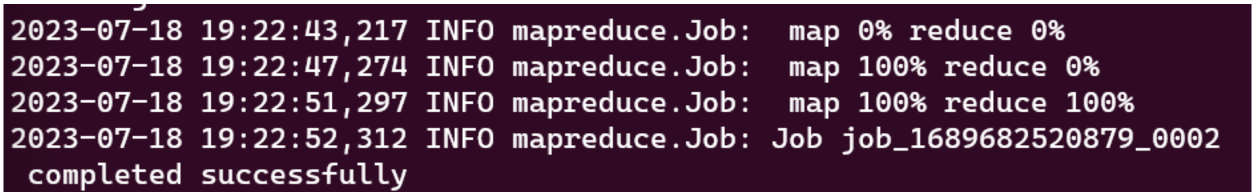
```
ducduy@DuyDo:/mnt/c/Users/84868/Desktop/Mapreduce/Lab 2/As4$ javac -classpath $HADOOP_CLASSPATH -d tutorial_classes Patent.java
Note: Patent.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
```

Step 4: Put the output files in a jar file.
jar -cvf <.jar file's path> -C <classes folder's path> .



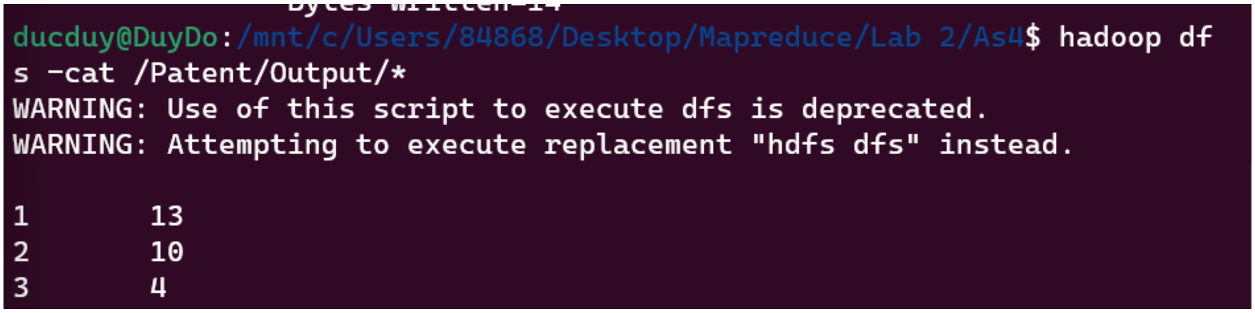
```
ducduy@DuyDo:/mnt/c/Users/84868/Desktop/Mapreduce/Lab 2/As4$ jar -cvf
Patent.jar -C tutorial_classes .
added manifest
adding: Patent$Map.class(in = 1777) (out= 759)(deflated 57%)
adding: Patent$Reduce.class(in = 1568) (out= 659)(deflated 57%)
adding: Patent.class(in = 1895) (out= 953)(deflated 49%)
```

Step 5: Run the jar file on Hadoop.
hadoop jar <.jar file's path> Patent /Patent/Input /Patent/Output



```
2023-07-18 19:22:43,217 INFO mapreduce.Job:  map 0% reduce 0%
2023-07-18 19:22:47,274 INFO mapreduce.Job:  map 100% reduce 0%
2023-07-18 19:22:51,297 INFO mapreduce.Job:  map 100% reduce 100%
2023-07-18 19:22:52,312 INFO mapreduce.Job: Job job_1689682520879_0002
 completed successfully
```

Step 6: Result
hadoop dfs -cat /Patent/Output/*



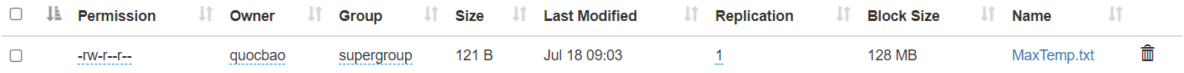
```
ducduy@DuyDo:/mnt/c/Users/84868/Desktop/Mapreduce/Lab 2/As4$ hadoop df
s -cat /Patent/Output/*
WARNING: Use of this script to execute dfs is deprecated.
WARNING: Attempting to execute replacement "hdfs dfs" instead.

1       13
2       10
3       4
```

5 MaxTemp Program

Note: the source code is based on the provided requirements file.

Step 1: Type the following command to export the hadoop classpath into bash.
export HADOOP_CLASSPATH=$(hadoop classpath)

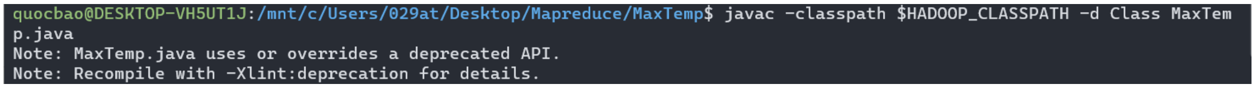
Step 2: Create directories on hdfs and put the input data file to hdfs.
hadoop fs -mkdir /MaxTemp
hadoop fs -mkdir /MaxTemp/Input
hadoop fs -put MaxTemp.txt /MaxTemp/Input



□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
□	-rw-r--r--	quocbao	supergroup	121 B	Jul 18 09:03	1	128 MB	MaxTemp.txt

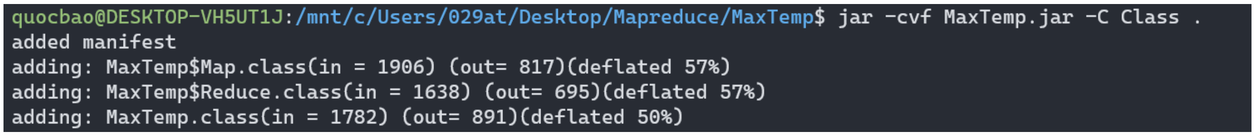
Step 3: Compile the MaxTemp.java file

javac -classpath $HADOOP_CLASSPATH -d <classes folder's path> <source's path>



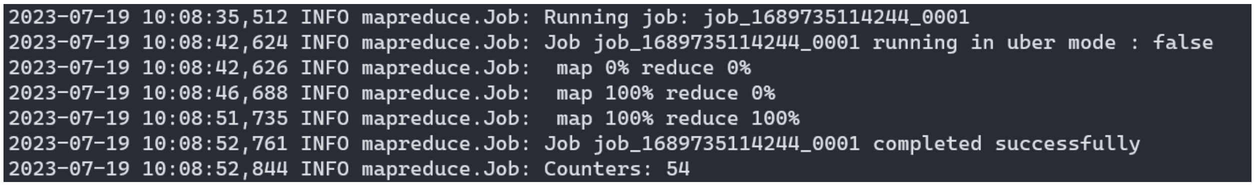
```
quocbao@DESKTOP-VH5UT1J:/mnt/c/Users/029at/Desktop/Mapreduce/MaxTemp$ javac -classpath $HADOOP_CLASSPATH -d Class MaxTem
p.java
Note: MaxTemp.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
```

Step 4: Put the output files in a jar file.
jar -cvf <.jar file's path> -C <classes folder's path> .



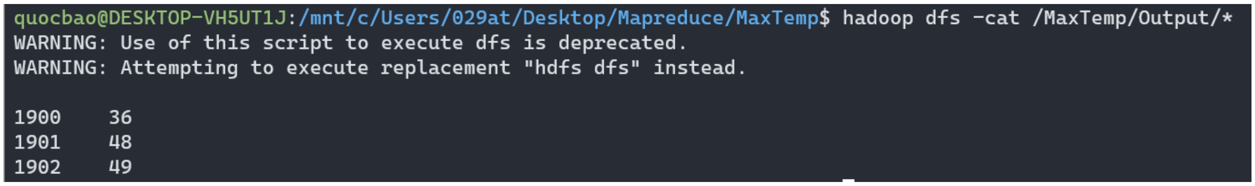
```
quocbao@DESKTOP-VH5UT1J:/mnt/c/Users/029at/Desktop/Mapreduce/MaxTemp$ jar -cvf MaxTemp.jar -C Class .
added manifest
adding: MaxTemp$Map.class(in = 1906) (out= 817)(deflated 57%)
adding: MaxTemp$Reduce.class(in = 1638) (out= 695)(deflated 57%)
adding: MaxTemp.class(in = 1782) (out= 891)(deflated 50%)
```

Step 5: Run the jar file on Hadoop.
hadoop jar MaxTemp.jar MaxTemp MaxTemp/Input /MaxTemp/Output



```
2023-07-19 10:08:35,512 INFO mapreduce.Job: Running job: job_1689735114244_0001
2023-07-19 10:08:42,624 INFO mapreduce.Job: Job job_1689735114244_0001 running in uber mode : false
2023-07-19 10:08:42,626 INFO mapreduce.Job:  map 0% reduce 0%
2023-07-19 10:08:46,688 INFO mapreduce.Job:  map 100% reduce 0%
2023-07-19 10:08:51,735 INFO mapreduce.Job:  map 100% reduce 100%
2023-07-19 10:08:52,761 INFO mapreduce.Job: Job job_1689735114244_0001 completed successfully
2023-07-19 10:08:52,844 INFO mapreduce.Job: Counters: 54
```

Step 6: Result
hadoop dfs -cat /MaxTemp/Output/*



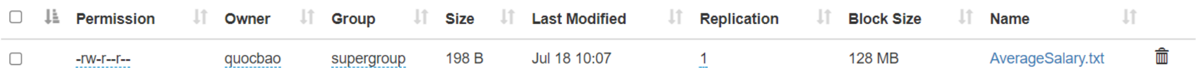
```
quocbao@DESKTOP-VH5UT1J:/mnt/c/Users/029at/Desktop/Mapreduce/MaxTemp$ hadoop dfs -cat /MaxTemp/Output/*
WARNING: Use of this script to execute dfs is deprecated.
WARNING: Attempting to execute replacement "hdfs dfs" instead.

1900    36
1901    48
1902    49
```

6 AverageSalary Program

Note: the source code is based on the provided requirements file.

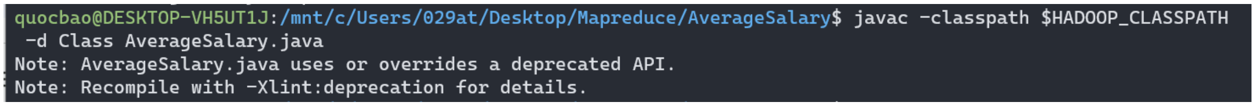
Step 1: Type the following command to export the hadoop classpath into bash.
export HADOOP_CLASSPATH=$(hadoop classpath)

Step 2: Create directories on hdfs and put the input data file to hdfs.
hadoop fs -mkdir /AverageSalary
hadoop fs -mkdir /AverageSalary/Input
hadoop fs -put AverageSalary.txt /AverageSalary/Input



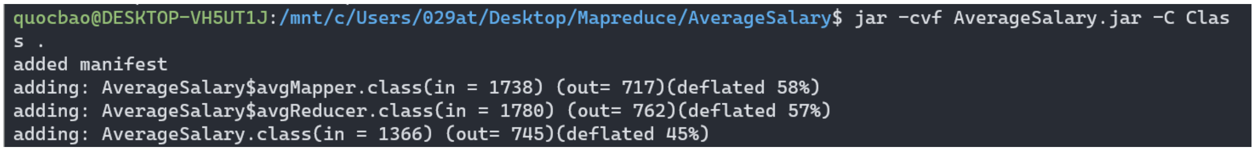
□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
□	-rw-r--r--	quocbao	supergroup	198 B	Jul 18 10:07	1	128 MB	AverageSalary.txt	🗑

Step 3: Compile the AverageSalary.java file
javac -classpath $HADOOP_CLASSPATH -d <classes folder's path> <source's path>



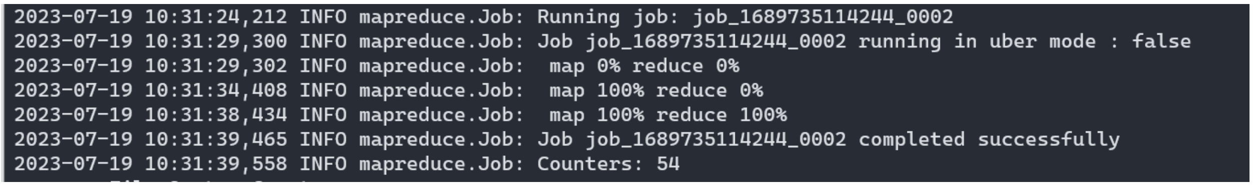
```
quocbao@DESKTOP-VH5UT1J:/mnt/c/Users/029at/Desktop/Mapreduce/AverageSalary$ javac -classpath $HADOOP_CLASSPATH
 -d Class AverageSalary.java
Note: AverageSalary.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
```

Step 4: Put the output files in a jar file.
jar -cvf <.jar file's path> -C <classes folder's path> .



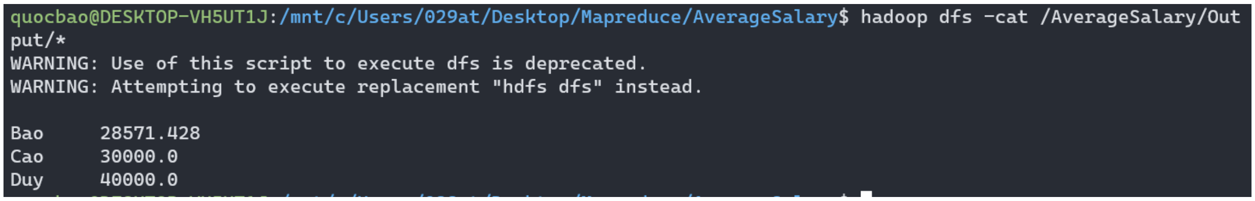
```
quocbao@DESKTOP-VH5UT1J:/mnt/c/Users/029at/Desktop/Mapreduce/AverageSalary$ jar -cvf AverageSalary.jar -C Clas
s .
added manifest
adding: AverageSalary$avgMapper.class(in = 1738) (out= 717)(deflated 58%)
adding: AverageSalary$avgReducer.class(in = 1780) (out= 762)(deflated 57%)
adding: AverageSalary.class(in = 1366) (out= 745)(deflated 45%)
```

Step 5: Run the jar file on Hadoop.
hadoop jar AverageSalary.jar AverageSalary AverageSalary/Input /AverageSalary/Output



```
2023-07-19 10:31:24,212 INFO mapreduce.Job: Running job: job_1689735114244_0002
2023-07-19 10:31:29,300 INFO mapreduce.Job: Job job_1689735114244_0002 running in uber mode : false
2023-07-19 10:31:29,302 INFO mapreduce.Job:  map 0% reduce 0%
2023-07-19 10:31:34,408 INFO mapreduce.Job:  map 100% reduce 0%
2023-07-19 10:31:38,434 INFO mapreduce.Job:  map 100% reduce 100%
2023-07-19 10:31:39,465 INFO mapreduce.Job: Job job_1689735114244_0002 completed successfully
2023-07-19 10:31:39,558 INFO mapreduce.Job: Counters: 54
```

Step 6: Result
hadoop dfs -cat /AverageSalary/Output/*



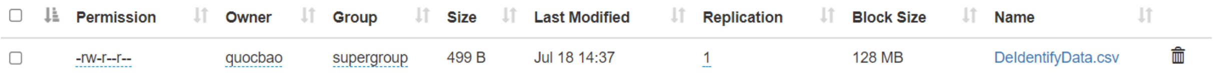
```
quocbao@DESKTOP-VH5UT1J:/mnt/c/Users/029at/Desktop/Mapreduce/AverageSalary$ hadoop dfs -cat /AverageSalary/Out
put/*
WARNING: Use of this script to execute dfs is deprecated.
WARNING: Attempting to execute replacement "hdfs dfs" instead.

Bao     28571.428
Cao     30000.0
Duy     40000.0
```

7 De Identify HealthCare Program

Note: the source code is based on the provided requirements file.

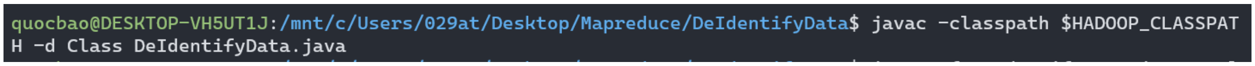
Step 1: Type the following command to export the hadoop classpath into bash.
export HADOOP_CLASSPATH=$(hadoop classpath)

Step 2: Create directories on hdfs and put the input data file to hdfs.
hadoop fs -mkdir /DeIdentifyData
hadoop fs -mkdir /DeIdentifyData/Input
hadoop fs -put DeIdentifyData.txt /DeIdentifyData



□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
□	-rw-r--r--	quocbao	supergroup	499 B	Jul 18 14:37	1	128 MB	DeIdentifyData.csv

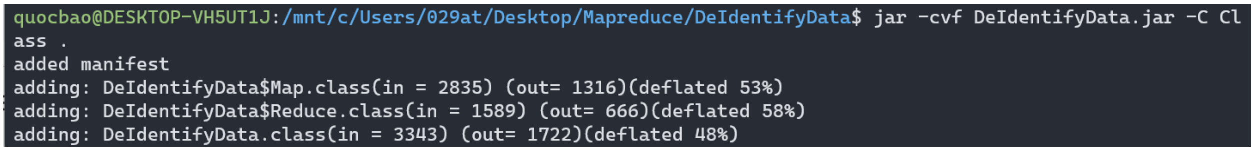
Step 3: Compile the DeIdentifyData.java file
javac -classpath $HADOOP_CLASSPATH -d <classes folder's path> <source's path>



```
quocbao@DESKTOP-VH5UT1J:/mnt/c/Users/029at/Desktop/Mapreduce/DeIdentifyData$ javac -classpath $HADOOP_CLASSPAT
H -d Class DeIdentifyData.java
```

Step 4: Put the output files in a jar file.

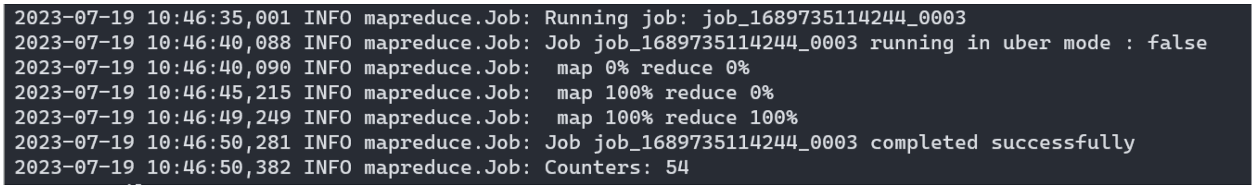
jar -cvf <.jar file’s path> -C <classes folder’s path> .



```
quocbao@DESKTOP-VH5UT1J:/mnt/c/Users/029at/Desktop/Mapreduce/DeIdentifyData$ jar -cvf DeIdentifyData.jar -C Cl
ass .
added manifest
adding: DeIdentifyData$Map.class(in = 2835) (out= 1316)(deflated 53%)
adding: DeIdentifyData$Reduce.class(in = 1589) (out= 666)(deflated 58%)
adding: DeIdentifyData.class(in = 3343) (out= 1722)(deflated 48%)
```

Step 5: Run the jar file on Hadoop.

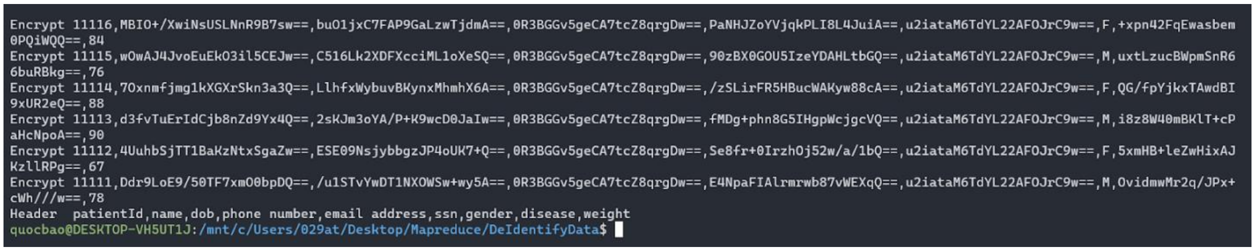
hadoop jar DeIdentifyData.jar DeIdentifyData DeIdentifyData/Input /DeIdentifyData/Output



```
2023-07-19 10:46:35,001 INFO mapreduce.Job: Running job: job_1689735114244_0003
2023-07-19 10:46:40,088 INFO mapreduce.Job: Job job_1689735114244_0003 running in uber mode : false
2023-07-19 10:46:40,090 INFO mapreduce.Job:  map 0% reduce 0%
2023-07-19 10:46:45,215 INFO mapreduce.Job:  map 100% reduce 0%
2023-07-19 10:46:49,249 INFO mapreduce.Job:  map 100% reduce 100%
2023-07-19 10:46:50,281 INFO mapreduce.Job: Job job_1689735114244_0003 completed successfully
2023-07-19 10:46:50,382 INFO mapreduce.Job: Counters: 54
```

Step 6: Result

hadoop dfs -cat /DeIdentifyData/Output/*



```
Encrypt 11116,MBIO+/XwiNsUSLNnR9B7sw==,buO1jxC7FAP9GaLzwTjdmA==,0R3BGGv5geCA7tcZ8qrgDw==,PaNHJZoYVjqkPLI8L4JuiA==,u2iataM6TdYL22AFOJrC9w==,F,+xpn42FqEwasbem
0PQiWQQ==,84
Encrypt 11115,wOwAJ4JvoEuEkO3il5CEJw==,C516Lk2XDFXcciML1oXeSQ==,0R3BGGv5geCA7tcZ8qrgDw==,90zBX0GOU5IzeYDAHLtbGQ==,u2iataM6TdYL22AFOJrC9w==,M,uxtLzucBWpmSnR6
6buRBkg==,76
Encrypt 11114,7Oxnmfjmg1kXGXrSkn3a3Q==,LlhfxWybuvBKynxMhmhX6A==,0R3BGGv5geCA7tcZ8qrgDw==,/zSLirFR5HBucWAKyw88cA==,u2iataM6TdYL22AFOJrC9w==,F,QG/fpYjkxTAwdBI
9xUR2eQ==,88
Encrypt 11113,d3fvTuErIdCjb8nZd9Yx4Q==,2sKJm3oYA/P+K9wcD0JaIw==,0R3BGGv5geCA7tcZ8qrgDw==,fMDg+phn8G5IHgpWcjgcVQ==,u2iataM6TdYL22AFOJrC9w==,M,i8z8W40mBKlT+cP
aHcNpoA==,90
Encrypt 11112,4UuhbSjTT1BaKzNtxSgaZw==,ESE09NsjybbgzJP4oUK7+Q==,0R3BGGv5geCA7tcZ8qrgDw==,Se8fr+0IrzhOj52w/a/1bQ==,u2iataM6TdYL22AFOJrC9w==,F,5xmHB+leZwHixAJ
KzllRPg==,67
Encrypt 11111,Ddr9LoE9/50TF7xmO0bpDQ==,/u1STvYwDT1NXOWSw+wy5A==,0R3BGGv5geCA7tcZ8qrgDw==,E4NpaFIAlrmrwb87vWEXqQ==,u2iataM6TdYL22AFOJrC9w==,M,OvidmwMr2q/JPx+
cWh///w==,78
Header  patientId,name,dob,phone number,email address,ssn,gender,disease,weight
quocbao@DESKTOP-VH5UT1J:/mnt/c/Users/029at/Desktop/Mapreduce/DeIdentifyData$
```

8 Music Track Program

Note: Statement 1 is based on the provided requirements file.
The statements from 2 to 5 are based on the Blogs.

Step 1: Type the following command to export the hadoop classpath into bash.

export HADOOP_CLASSPATH=$(hadoop classpath)

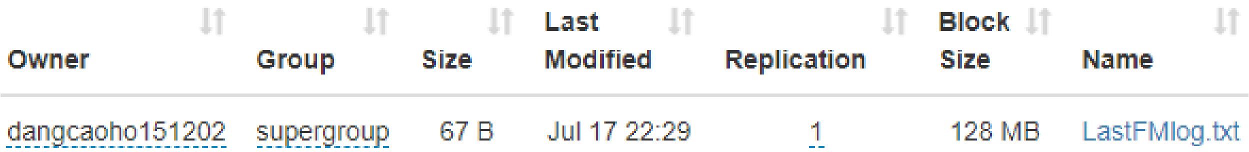
echo $HADOOP_CLASSPATH

Step 2: Create directories on hdfs and put the input data file to hdfs.

hadoop fs -mkdir /MusicTrack

hadoop fs -mkdir /MusicTrack/Input

hadoop fs -put <input file’s path> /MusicTrack/Input

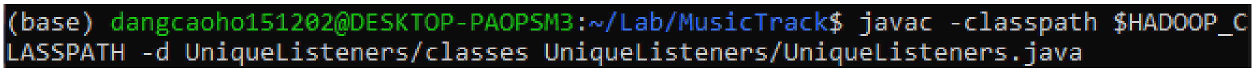


Owner	Group	Size	Last Modified	Replication	Block Size	Name
dangcaoho151202	supergroup	67 B	Jul 17 22:29	1	128 MB	LastFMlog.txt

Statement 1: Number of unique listeners

Step 3: Compile the UniqueListeners.java file

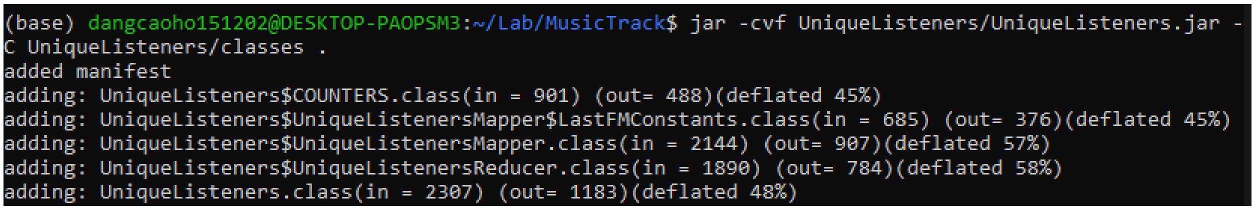
javac -classpath $HADOOP_CLASSPATH -d <classes folder’s path> <source’s path>



```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/MusicTrack$ javac -classpath $HADOOP_C
LASSPATH -d UniqueListeners/classes UniqueListeners/UniqueListeners.java
```

Step 4: Put the output files in a jar file.

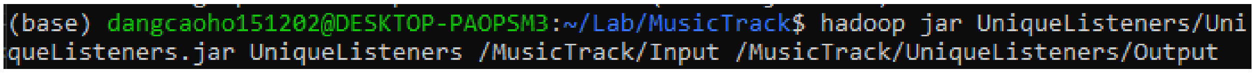
jar -cvf <.jar file’s path> -C <classes folder’s path> .



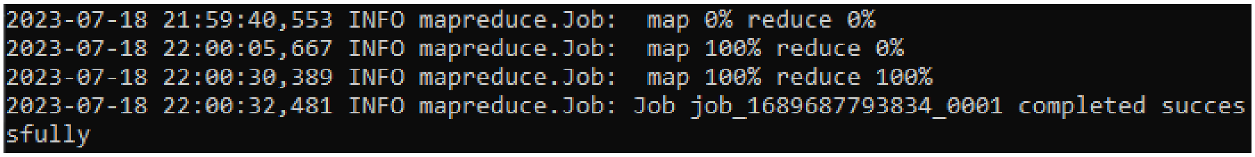
```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/MusicTrack$ jar -cvf UniqueListeners/UniqueListeners.jar -
C UniqueListeners/classes .
added manifest
adding: UniqueListeners$COUNTERS.class(in = 901) (out= 488)(deflated 45%)
adding: UniqueListeners$UniqueListenersMapper$LastFMConstants.class(in = 685) (out= 376)(deflated 45%)
adding: UniqueListeners$UniqueListenersMapper.class(in = 2144) (out= 907)(deflated 57%)
adding: UniqueListeners$UniqueListenersReducer.class(in = 1890) (out= 784)(deflated 58%)
adding: UniqueListeners.class(in = 2307) (out= 1183)(deflated 48%)
```

Step 5: Run the jar file on Hadoop.

hadoop jar <.jar file’s path> UniqueListeners /MusicTrack/Input /MusicTrack/UniqueListeners/Output



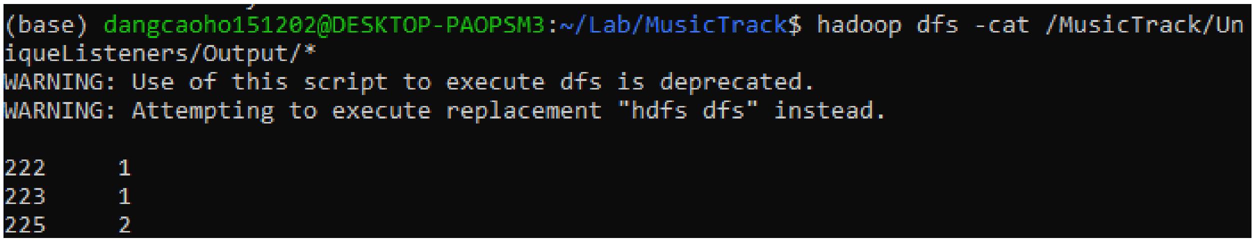
```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/MusicTrack$ hadoop jar UniqueListeners/Uni
queListeners.jar UniqueListeners /MusicTrack/Input /MusicTrack/UniqueListeners/Output
```



```
2023-07-18 21:59:40,553 INFO mapreduce.Job:  map 0% reduce 0%
2023-07-18 22:00:05,667 INFO mapreduce.Job:  map 100% reduce 0%
2023-07-18 22:00:30,389 INFO mapreduce.Job:  map 100% reduce 100%
2023-07-18 22:00:32,481 INFO mapreduce.Job: Job job_1689687793834_0001 completed succes
sfully
```

Step 6: Result

hadoop dfs -cat /MusicTrack/UniqueListeners/Output/*

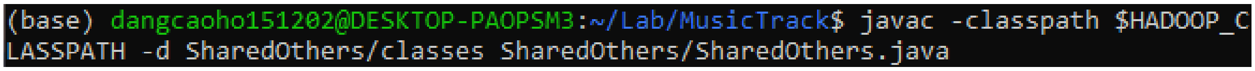


```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/MusicTrack$ hadoop dfs -cat /MusicTrack/Un
iqueListeners/Output/*
WARNING: Use of this script to execute dfs is deprecated.
WARNING: Attempting to execute replacement "hdfs dfs" instead.

222     1
223     1
225     2
```

Statement 2: Number of times the track was shared with others

Step 3: Compile the SharedOthers.java file

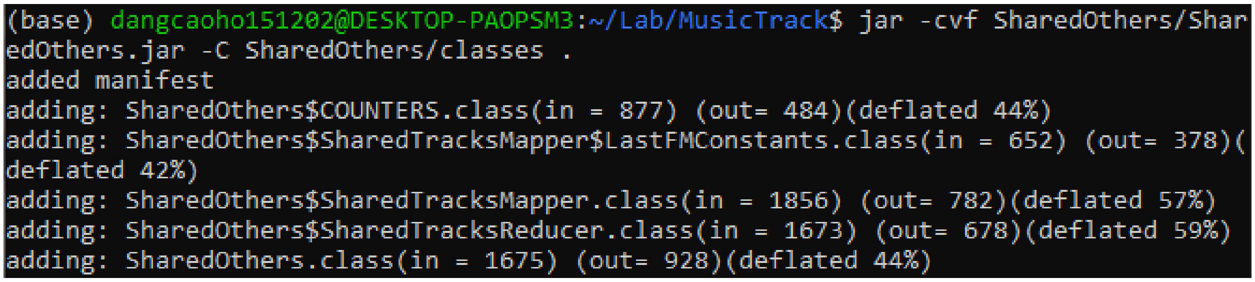
javac -classpath $HADOOP_CLASSPATH -d <classes folder’s path> <source’s path>



```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/MusicTrack$ javac -classpath $HADOOP_C
LASSPATH -d SharedOthers/classes SharedOthers/SharedOthers.java
```

Step 4: Put the output files in a jar file.

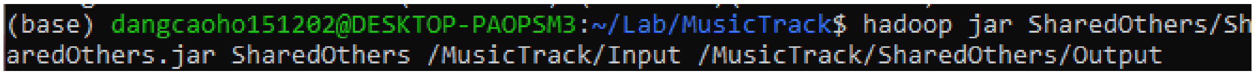
jar -cvf <.jar file’s path> -C <classes folder’s path> .



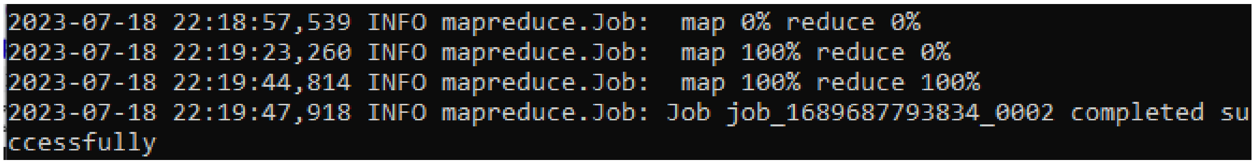
```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/MusicTrack$ jar -cvf SharedOthers/SharedOthers.jar -C SharedOthers/classes .
added manifest
adding: SharedOthers$COUNTERS.class(in = 877) (out= 484)(deflated 44%)
adding: SharedOthers$SharedTracksMapper$LastFMConstants.class(in = 652) (out= 378)(deflated 42%)
adding: SharedOthers$SharedTracksMapper.class(in = 1856) (out= 782)(deflated 57%)
adding: SharedOthers$SharedTracksReducer.class(in = 1673) (out= 678)(deflated 59%)
adding: SharedOthers.class(in = 1675) (out= 928)(deflated 44%)
```

Step 5: Run the jar file on Hadoop.

hadoop jar <.jar file’s path> SharedOthers /MusicTrack/Input /MusicTrack/SharedOthers/Output



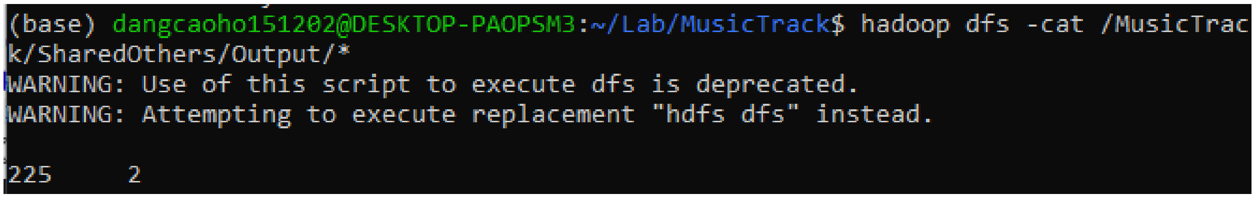
```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/MusicTrack$ hadoop jar SharedOthers/SharedOthers.jar SharedOthers /MusicTrack/Input /MusicTrack/SharedOthers/Output
```



```
2023-07-18 22:18:57,539 INFO mapreduce.Job:  map 0% reduce 0%
2023-07-18 22:19:23,260 INFO mapreduce.Job:  map 100% reduce 0%
2023-07-18 22:19:44,814 INFO mapreduce.Job:  map 100% reduce 100%
2023-07-18 22:19:47,918 INFO mapreduce.Job: Job job_1689687793834_0002 completed successfully
```

Step 6: Result

hadoop dfs -cat /MusicTrack/SharedOthers/Output/*

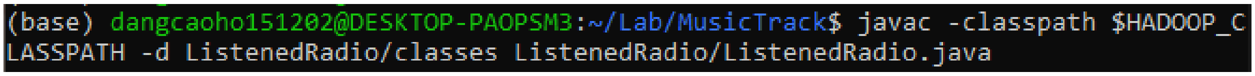


```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/MusicTrack$ hadoop dfs -cat /MusicTrack/SharedOthers/Output/*
WARNING: Use of this script to execute dfs is deprecated.
WARNING: Attempting to execute replacement "hdfs dfs" instead.

225     2
```

Statement 3: Number of times the track was listened to on the radio

Step 3: Compile the ListenedRadio.java file

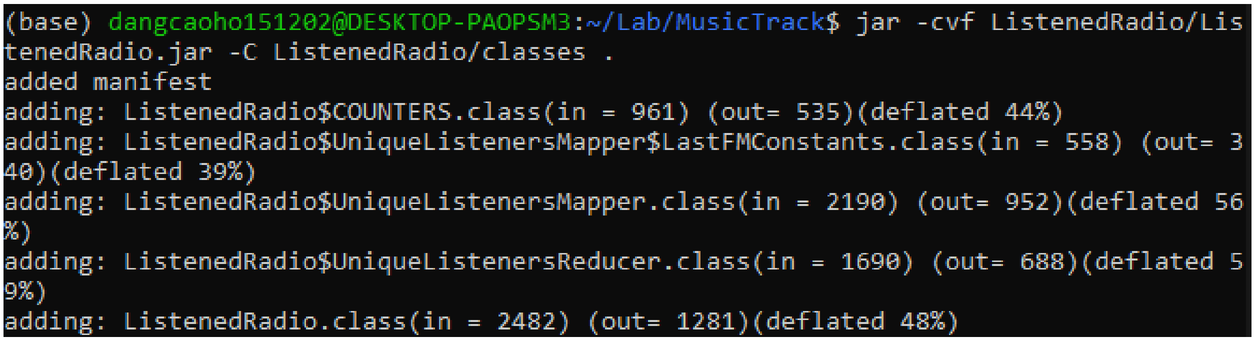
javac -classpath $HADOOP_CLASSPATH -d <classes folder’s path> <source’s path>



```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/MusicTrack$ javac -classpath $HADOOP_CLASSPATH -d ListenedRadio/classes ListenedRadio/ListenedRadio.java
```

Step 4: Put the output files in a jar file.

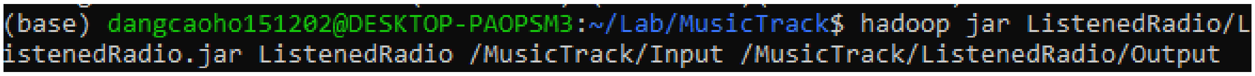
jar -cvf <.jar file’s path> -C <classes folder’s path> .



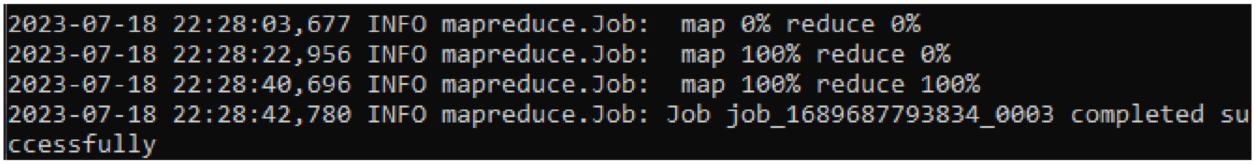
```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/MusicTrack$ jar -cvf ListenedRadio/ListenedRadio.jar -C ListenedRadio/classes .
added manifest
adding: ListenedRadio$COUNTERS.class(in = 961) (out= 535)(deflated 44%)
adding: ListenedRadio$UniqueListenersMapper$LastFMConstants.class(in = 558) (out= 340)(deflated 39%)
adding: ListenedRadio$UniqueListenersMapper.class(in = 2190) (out= 952)(deflated 56%)
adding: ListenedRadio$UniqueListenersReducer.class(in = 1690) (out= 688)(deflated 59%)
adding: ListenedRadio.class(in = 2482) (out= 1281)(deflated 48%)
```

Step 5: Run the jar file on Hadoop.

hadoop jar <.jar file’s path> ListenedRadio /MusicTrack/Input /MusicTrack/ListenedRadio/Output



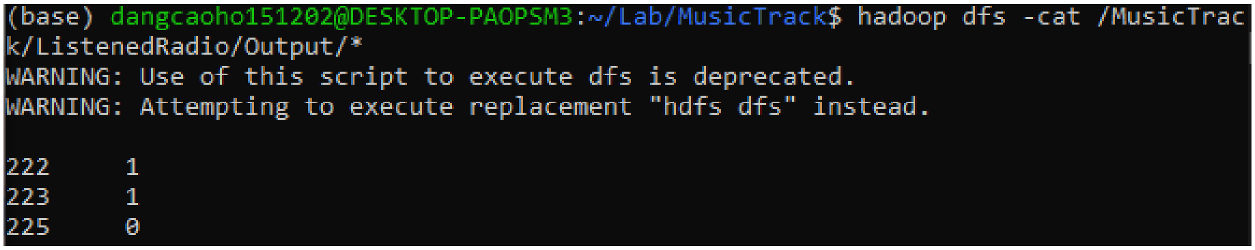
```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/MusicTrack$ hadoop jar ListenedRadio/L
istenedRadio.jar ListenedRadio /MusicTrack/Input /MusicTrack/ListenedRadio/Output
```



```
2023-07-18 22:28:03,677 INFO mapreduce.Job:  map 0% reduce 0%
2023-07-18 22:28:22,956 INFO mapreduce.Job:  map 100% reduce 0%
2023-07-18 22:28:40,696 INFO mapreduce.Job:  map 100% reduce 100%
2023-07-18 22:28:42,780 INFO mapreduce.Job: Job job_1689687793834_0003 completed su
ccessfully
```

Step 6: Result

hadoop dfs -cat /MusicTrack/ListenedRadio/Output/*

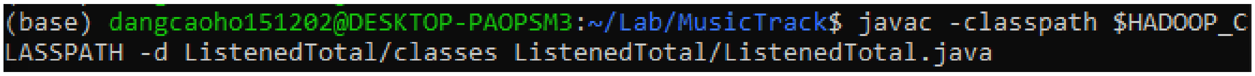


```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/MusicTrack$ hadoop dfs -cat /MusicTrac
k/ListenedRadio/Output/*
WARNING: Use of this script to execute dfs is deprecated.
WARNING: Attempting to execute replacement "hdfs dfs" instead.

222     1
223     1
225     0
```

Statement 4: Number of times the track was listened to in total

Step 3: Compile the ListenedTotal.java file

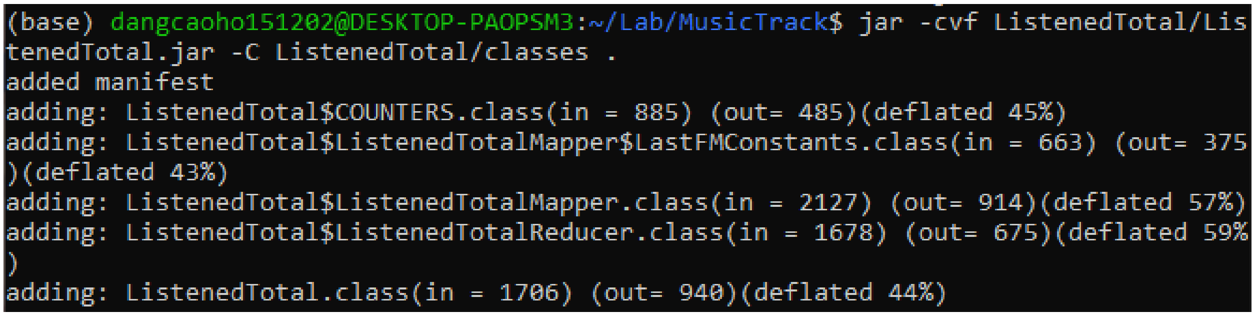
javac -classpath $HADOOP_CLASSPATH -d <classes folder’s path> <source’s path>



```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/MusicTrack$ javac -classpath $HADOOP_C
LASSPATH -d ListenedTotal/classes ListenedTotal/ListenedTotal.java
```

Step 4: Put the output files in a jar file.

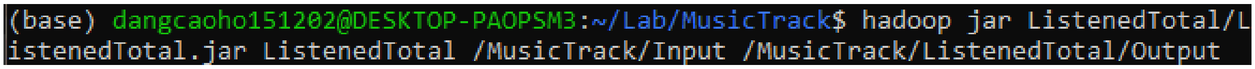
jar -cvf <.jar file’s path> -C <classes folder’s path> .



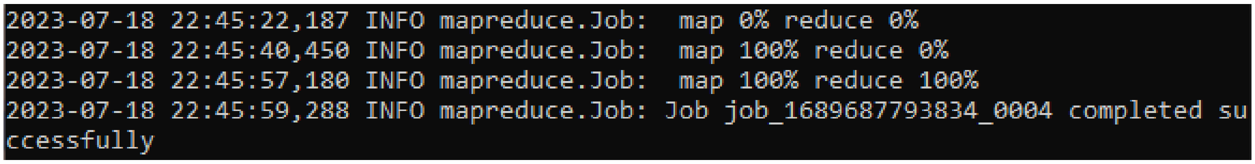
```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/MusicTrack$ jar -cvf ListenedTotal/Lis
tenedTotal.jar -C ListenedTotal/classes .
added manifest
adding: ListenedTotal$COUNTERS.class(in = 885) (out= 485)(deflated 45%)
adding: ListenedTotal$ListenedTotalMapper$LastFMConstants.class(in = 663) (out= 375
)(deflated 43%)
adding: ListenedTotal$ListenedTotalMapper.class(in = 2127) (out= 914)(deflated 57%)
adding: ListenedTotal$ListenedTotalReducer.class(in = 1678) (out= 675)(deflated 59%
)
adding: ListenedTotal.class(in = 1706) (out= 940)(deflated 44%)
```

Step 5: Run the jar file on Hadoop.

hadoop jar <.jar file’s path> ListenedTotal /MusicTrack/Input /MusicTrack/ListenedTotal/Output



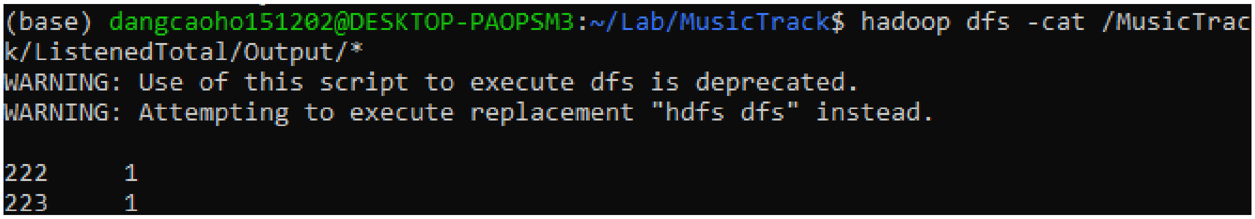
```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/MusicTrack$ hadoop jar ListenedTotal/L
istenedTotal.jar ListenedTotal /MusicTrack/Input /MusicTrack/ListenedTotal/Output
```



```
2023-07-18 22:45:22,187 INFO mapreduce.Job:  map 0% reduce 0%
2023-07-18 22:45:40,450 INFO mapreduce.Job:  map 100% reduce 0%
2023-07-18 22:45:57,180 INFO mapreduce.Job:  map 100% reduce 100%
2023-07-18 22:45:59,288 INFO mapreduce.Job: Job job_1689687793834_0004 completed su
ccessfully
```

Step 6: Result

hadoop dfs -cat /MusicTrack/ListenedTotal/Output/*

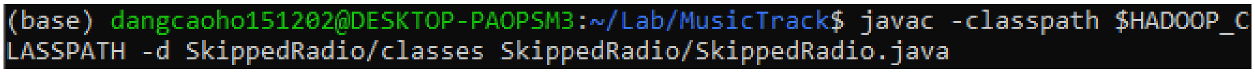


```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/MusicTrack$ hadoop dfs -cat /MusicTrac
k/ListenedTotal/Output/*
WARNING: Use of this script to execute dfs is deprecated.
WARNING: Attempting to execute replacement "hdfs dfs" instead.

222     1
223     1
```

Statement 5: Number of times the track was skipped on the radio

Step 3: Compile the SkippedRadio.java file

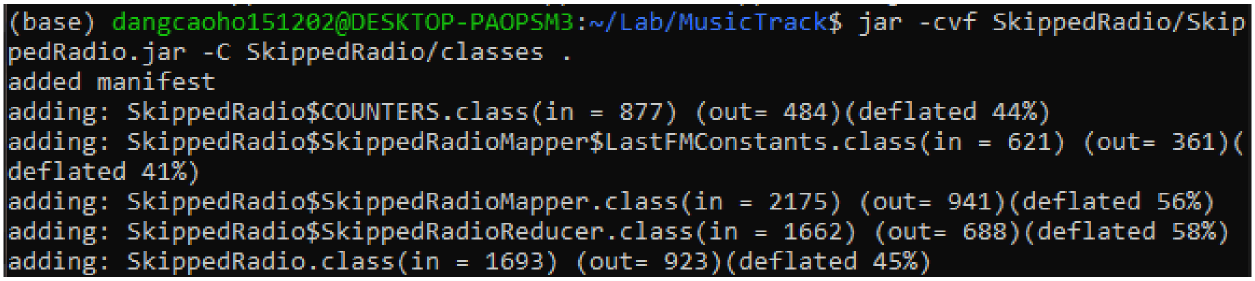
javac -classpath $HADOOP_CLASSPATH -d <classes folder’s path> <source’s path>



```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/MusicTrack$ javac -classpath $HADOOP_C
LASSPATH -d SkippedRadio/classes SkippedRadio/SkippedRadio.java
```

Step 4: Put the output files in a jar file.

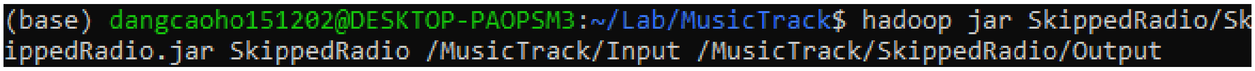
jar -cvf <.jar file’s path> -C <classes folder’s path> .



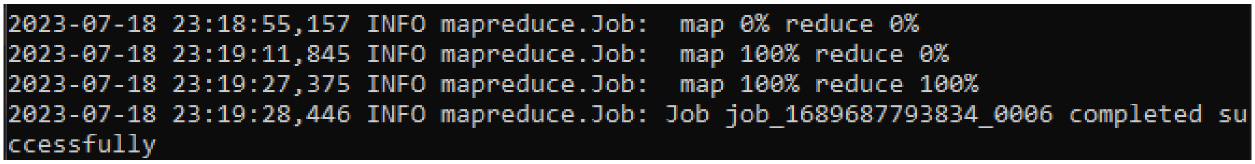
```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/MusicTrack$ jar -cvf SkippedRadio/Skip
pedRadio.jar -C SkippedRadio/classes .
added manifest
adding: SkippedRadio$COUNTERS.class(in = 877) (out= 484)(deflated 44%)
adding: SkippedRadio$SkippedRadioMapper$LastFMConstants.class(in = 621) (out= 361)(
deflated 41%)
adding: SkippedRadio$SkippedRadioMapper.class(in = 2175) (out= 941)(deflated 56%)
adding: SkippedRadio$SkippedRadioReducer.class(in = 1662) (out= 688)(deflated 58%)
adding: SkippedRadio.class(in = 1693) (out= 923)(deflated 45%)
```

Step 5: Run the jar file on Hadoop.

hadoop jar <.jar file’s path> SkippedRadio /MusicTrack/Input /MusicTrack/SkippedRadio/Output



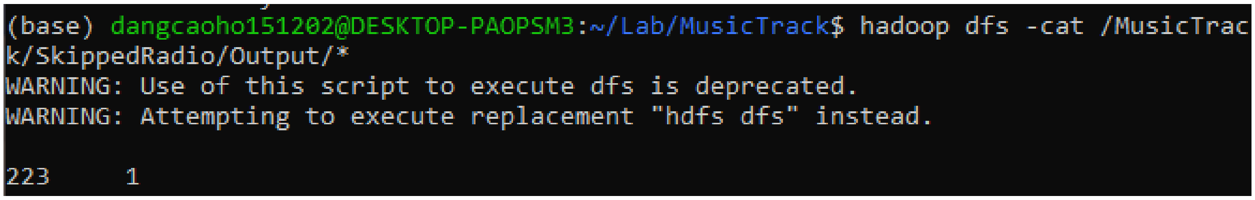
```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/MusicTrack$ hadoop jar SkippedRadio/Sk
ippedRadio.jar SkippedRadio /MusicTrack/Input /MusicTrack/SkippedRadio/Output
```



```
2023-07-18 23:18:55,157 INFO mapreduce.Job:  map 0% reduce 0%
2023-07-18 23:19:11,845 INFO mapreduce.Job:  map 100% reduce 0%
2023-07-18 23:19:27,375 INFO mapreduce.Job:  map 100% reduce 100%
2023-07-18 23:19:28,446 INFO mapreduce.Job: Job job_1689687793834_0006 completed su
ccessfully
```

Step 6: Result

hadoop dfs -cat /MusicTrack/SkippedRadio/Output/*



```
(base) dangcaoho151202@DESKTOP-PAOPSM3:~/Lab/MusicTrack$ hadoop dfs -cat /MusicTrac
k/SkippedRadio/Output/*
WARNING: Use of this script to execute dfs is deprecated.
WARNING: Attempting to execute replacement "hdfs dfs" instead.

223     1
```

9 Telecom Call Data Record Program

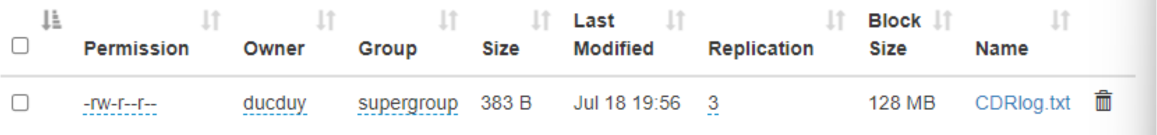
Note: the source code is referenced from the provided requirements file.

Step 1: Type the following command to export the hadoop classpath into bash.

```
export HADOOP_CLASSPATH=$(hadoop classpath)
echo $HADOOP_CLASSPATH
```

Step 2: Create directories on hdfs and put the input data file to hdfs.

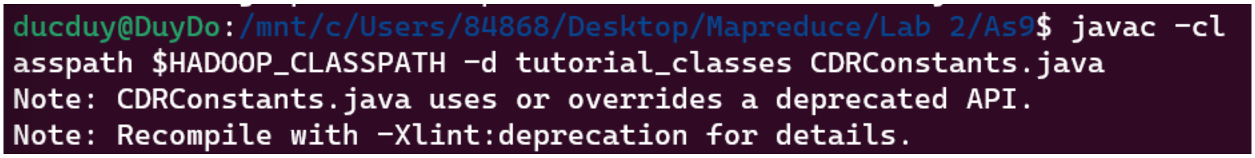
```
hadoop fs -mkdir /CDRlog
hadoop fs -mkdir /CDRlog/Input
hadoop fs -put <input file’s path> /CDRlog/Input
```



□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
□	-rw-r--r--	ducduy	supergroup	383 B	Jul 18 19:56	3	128 MB	CDRlog.txt	🗑

Step 3: Compile the CDRConstants.java file

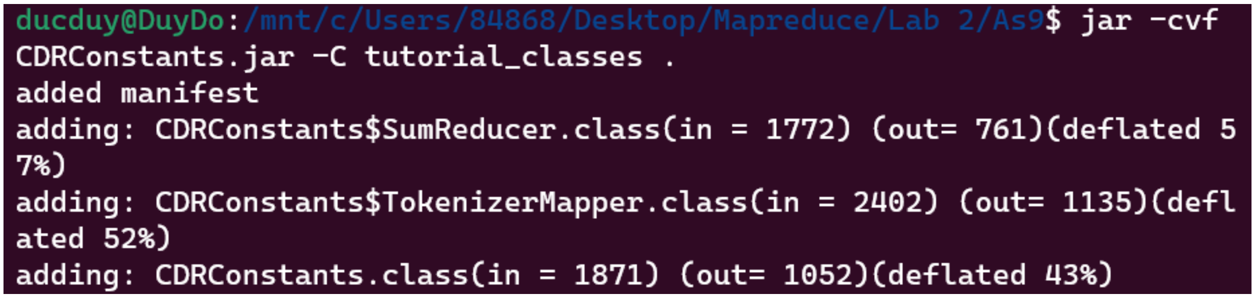
```
javac -classpath $HADOOP_CLASSPATH -d <classes folder’s path> <source’s path>
```



```
ducduy@DuyDo:/mnt/c/Users/84868/Desktop/Mapreduce/Lab 2/As9$ javac -cl
asspath $HADOOP_CLASSPATH -d tutorial_classes CDRConstants.java
Note: CDRConstants.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
```

Step 4: Put the output files in a jar file.

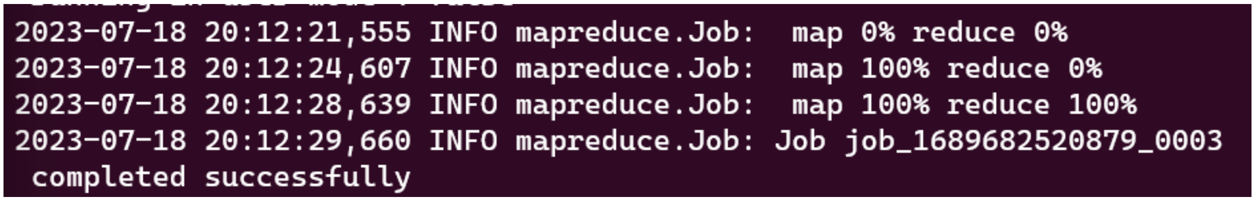
```
jar -cvf <.jar file’s path> -C <classes folder’s path> .
```



```
ducduy@DuyDo:/mnt/c/Users/84868/Desktop/Mapreduce/Lab 2/As9$ jar -cvf
CDRConstants.jar -C tutorial_classes .
added manifest
adding: CDRConstants$SumReducer.class(in = 1772) (out= 761)(deflated 5
7%)
adding: CDRConstants$TokenizerMapper.class(in = 2402) (out= 1135)(defl
ated 52%)
adding: CDRConstants.class(in = 1871) (out= 1052)(deflated 43%)
```

Step 5: Run the jar file on Hadoop.

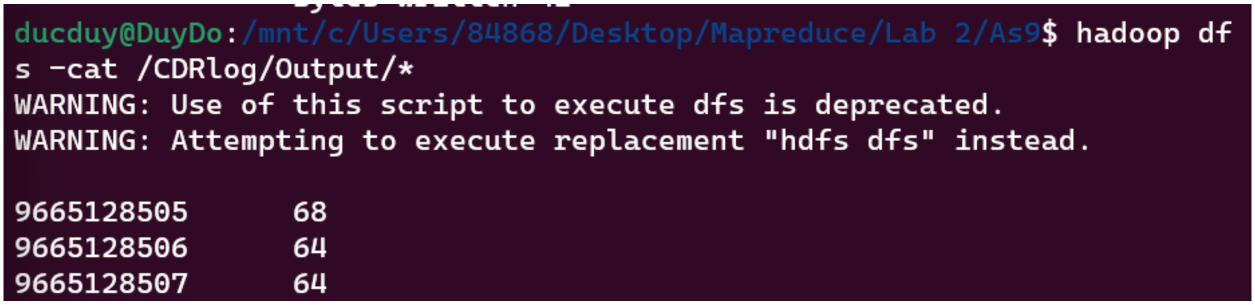
```
hadoop jar <.jar file’s path> CDRlog /CDRlog/Input /CDRlog/Output
```



```
2023-07-18 20:12:21,555 INFO mapreduce.Job:  map 0% reduce 0%
2023-07-18 20:12:24,607 INFO mapreduce.Job:  map 100% reduce 0%
2023-07-18 20:12:28,639 INFO mapreduce.Job:  map 100% reduce 100%
2023-07-18 20:12:29,660 INFO mapreduce.Job: Job job_1689682520879_0003
 completed successfully
```

Step 6: Result

hadoop dfs -cat /CDRlog/Output/*

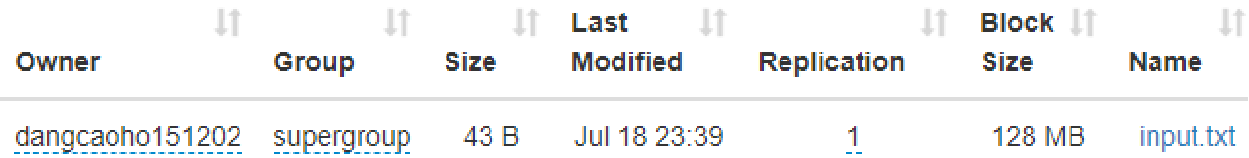


```
ducduy@DuyDo:/mnt/c/Users/84868/Desktop/Mapreduce/Lab 2/As9$ hadoop df
s -cat /CDRlog/Output/*
WARNING: Use of this script to execute dfs is deprecated.
WARNING: Attempting to execute replacement "hdfs dfs" instead.

9665128505      68
9665128506      64
9665128507      64
```

10 Count Connected Component Program

Step 1: Type the following command to export the hadoop classpath into bash.
export HADOOP_CLASSPATH=$(hadoop classpath)
echo $HADOOP_CLASSPATH

Step 2: Create directories on hdfs and put the input data file to hdfs.
hadoop fs -mkdir /CountConnectedComponentProgram
hadoop fs -mkdir /CountConnectedComponentProgram/Input
hadoop fs -put <input file’s path> /CountConnectedComponentProgram/Input



Owner	Group	Size	Last Modified	Replication	Block Size	Name
dangcaoho151202	supergroup	43 B	Jul 18 23:39	1	128 MB	input.txt

References

The provided requirements file.
Manohar, 2 August 2017, MapReduce Real time.
Rkrahul04, May 8, 2017, Word_size_Count_Mapreduce.
Rkrahul04, 2017, Sub-Patents_count_mapreduce.