

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT on

BIG DATA ANALYTICS (20CS6PEBDA)

Submitted by

Pranav Kumar (1BM19CS114)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING



B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019

May-2022 to July-2022

B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering



CERTIFICATE

This is to certify that the Lab work entitled “**BIG DATAANALYTICS** ” carried out by **Pranav Kumar (1BM19CS114)**, who is bonafide student of **B. M. S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a **BIG DATA ANALYTICS - (20CS6PEBDA)** work prescribed for the said degree.

Antara Roy Choudhury
Assistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

Index Sheet

Sl. No.	Experiment Title	Page No.
1	MongoDB- CRUD Demonstration	4
2	Employee database using Cassandra	8
3	Library database using Cassandra	13

Course Outcome

CO1	Apply the concept of NoSQL, Hadoop or Spark for a given task.
CO2	Analyze the Big Data and obtain insight using data analytics mechanisms.
CO3	Design and implement Big data applications by applying NoSQL, Hadoop or Spark.

LAB PROGRAM 1: MongoDB- CRUD Demonstration

1) Using MongoDB

i) Create a database for Students and Create a Student Collection (_id,Name, USN, Semester, Dept_Name, CGPA, Hobbies(Set)).

```
use myDB;
```

```
db.createCollection("Student");
```

ii) Insert required documents to the collection.

```
> db.Student.insert({_id:1,Name: "Pranav", sem:"VI",dept: "CSE",CGPA: 8.2,hobbies: ['cycling']});
```

```
WriteResult({ "nInserted" : 1 })
```

```
> db.Student.insert({_id:2,Name: "Anurag", sem:"VII",dept: "ECE",CGPA: 6.8,hobbies: ["Biking"]});
```

```
WriteResult({ "nInserted" : 1 })
```

```
> db.Student.insert({_id:3,Name: "Saurab", sem:"VI",dept:"Architecture",CGPA: 8.8,hobbies: ['Gaming']});
```

```
WriteResult({ "nInserted" : 1 })
```

```
> db.Student.insert({_id:4,Name: "Prateek", sem:"V",dept: "ISE",CGPA: 9.1,hobbies: ["Badminton"]});
```

```
WriteResult({ "nInserted" : 1 })
```

```

> db.Student.insert({_id:1,Name: "Pranav", sem:"VI",dept: "CSE",CGPA: 8.2,hobbies: ['cycling']});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:2,Name: "Anurag", sem:"VII",dept: "ECE",CGPA: 6.8,hobbies: ["Biking"]});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:3,Name: "Saurab", sem:"VI",dept:"Architecture",CGPA: 8.8,hobbies: ['Gaming']});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:4,Name: "Prateek", sem:"V",dept: "ISE",CGPA: 9.1,hobbies: ["Badminton"]});
WriteResult({ "nInserted" : 1 })
> db.Student.find()
{ "_id" : 1, "Name" : "Pranav", "sem" : "VI", "dept" : "CSE", "CGPA" : 8.2, "hobbies" : [ "cycling" ] }
{ "_id" : 2, "Name" : "Anurag", "sem" : "VII", "dept" : "ECE", "CGPA" : 6.8, "hobbies" : [ "Biking" ] }
{ "_id" : 3, "Name" : "Saurab", "sem" : "VI", "dept" : "Architecture", "CGPA" : 8.8, "hobbies" : [ "Gaming" ] }
{ "_id" : 4, "Name" : "Prateek", "sem" : "V", "dept" : "ISE", "CGPA" : 9.1, "hobbies" : [ "Badminton" ] }
>

```

iii) First Filter on “Dept_Name:CSE” and then group it on “Semester” and compute the Average CPGA for that semester and filter those documents where the “Avg_CPGA” is greater than 7.5.

```

>db.Student.aggregate({$match:{dept:"CSE"}},{ $group:{_id:"$sem",AverageCGPA:{ $avg:"$CGPA"} }},{ $match:{AverageCGPA:{ $gt:7.5}}});

```

```

> db.Student.aggregate({$match:{dept:"CSE"}},{ $group:{_id:"$sem",AverageCGPA:{ $avg:"$CGPA"} }},{ $match:{AverageCGPA:{ $gt:7.5}}});
{ "_id" : "VI", "AverageCGPA" : 8.2 }
>

```

iv) Insert the document for “Bhuvan” in to the Students collection only if it does not already exist in the collection. However, if it is already present in the collection, then update the document with new values. (Update his Hobbies to “Skating”) Use “Update else insert” (if there is an existing document, it will attempt to update it, if there is no existing document then it will insert it).

```

>db.Student.update({_id:5},{ $set:{ "hobbies": "Cricket" }},{ $upsert:true});

```

```

> db.Student.update({_id:5},{ $set:{ "hobbies": "Cricket" }},{ $upsert:true});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Student.find({_id:5});
{ "_id" : 5, "Name" : "Bhuvan", "sem" : "VI", "dept" : "CSE", "CGPA" : 9.5, "hobbies" : "Cricket" }
>

```

v)To display only the StudName and Grade from all the documents of the Students collection. The identifier_id should be suppressed and NOT displayed.

```
> db.Student.find({}, {_id:0,"Name":1,"sem":1});
```

```
> db.Student.find({}, {_id:0,"Name":1,"sem":1});
{ "Name" : "Pranav", "sem" : "VI" }
{ "Name" : "Anurag", "sem" : "VII" }
{ "Name" : "Saurab", "sem" : "VI" }
{ "Name" : "Prateek", "sem" : "V" }
{ "Name" : "Bhuvan", "sem" : "VI" }
```

vi) To find those documents where the Grade is set to 'VII'.

```
> db.Student.find({"sem":"VII"});
```

```
> db.Student.find({"sem":"VII"});
{ "_id" : 2, "Name" : "Anurag", "sem" : "VII", "dept" : "ECE", "CGPA" : 6.8, "hobbies" : [ "Biking" ] }
>
```

vii)To find those documents from the Students collection where the Hobbies is set to either 'Chess' or is set to 'Skating'.

```
> db.Student.find({"hobbies":{$in:["Badminton","Gaming"]} });
```

```
> db.Student.find({"hobbies":{$in:["Badminton","Gaming"]} });
{ "_id" : 3, "Name" : "Saurab", "sem" : "VI", "dept" : "Architecture", "CGPA" : 8.8, "hobbies" : [ "Gaming" ] }
{ "_id" : 4, "Name" : "Prateek", "sem" : "V", "dept" : "ISE", "CGPA" : 9.1, "hobbies" : [ "Badminton" ] }
{ "_id" : 5, "Name" : "Bhuvan", "sem" : "VI", "dept" : "CSE", "CGPA" : 9.5, "hobbies" : [ "Cricket", "Badminton" ] }
```

viii)To find documents from the Students collection where the StudName begins with "B" .

```
> db.Student.find({"Name":/^A/});
```

```
> db.Student.find({"Name":/^A/});
{ "_id" : 2, "Name" : "Anurag", "sem" : "VII", "dept" : "ECE", "CGPA" : 6.8, "hobbies" : [ "Biking" ] }
```

ix) To find the number of documents in the Students collection.

```
> db.Student.count();
```

```
> db.Student.count();
5
```

x) To sort the documents from the Students collection in the descending order of StudName.

```
> db.Student.find().sort({"Name":-1});
```

```
> db.Student.find().sort({"Name":-1});
{ "_id" : 3, "Name" : "Saurab", "sem" : "VI", "dept" : "Architecture", "CGPA" : 8.8, "hobbies" : [ "Gaming" ] }
{ "_id" : 4, "Name" : "Prateek", "sem" : "V", "dept" : "ISE", "CGPA" : 9.1, "hobbies" : [ "Badminton" ] }
{ "_id" : 1, "Name" : "Pranav", "sem" : "VI", "dept" : "CSE", "CGPA" : 8.2, "hobbies" : [ "cycling" ] }
{ "_id" : 5, "Name" : "Bhuvan", "sem" : "VI", "dept" : "CSE", "CGPA" : 9.5, "hobbies" : [ "Cricket", "Badminton" ] }
{ "_id" : 2, "Name" : "Anurag", "sem" : "VII", "dept" : "ECE", "CGPA" : 6.8, "hobbies" : [ "Biking" ] }
```

xi) Command used to export MongoDB JSON documents from “Student” Collection into the “Students” database into a CSV file “Output.txt”

```
> mongoexport --host localhost --db Student --collection Student --csv --out /Downloads/student.txt -fields "Name","sem";
```

```
> mongoexport --host localhost --db Student --collection Student --csv --out /Downloads/student.txt -fields "Name","sem";
uncaught exception: SyntaxError: unexpected token: identifier :
@(shell):1:14
```

LAB PROGRAM 2: Employee database using Cassandra

Program 1. Perform the following DB operations using Cassandra.

1. Create a key space by name Employee

```
cqlsh> create keyspace Employee with REPLICATION ={  
'class': 'SimpleStrategy', 'replication_factor': 1 ... };
```

```
bmsce@bmsce-Precision-T1700:~/cassandra/apache-cassandra-3.11.0/bin$ cqlsh  
Connected to Test Cluster at 127.0.0.1:9042.  
[cqlsh 5.0.1 | Cassandra 3.11.4 | CQL spec 3.4.4 | Native protocol v4]  
Use HELP for help.
```

```
cqlsh> use Employee; cqlsh:employee> describe keyspaces;
```

```
students system_auth system_distributed system_traces system_schema system  
employee
```

```
cqlsh> describe keyspace employee;  
  
CREATE KEYSPACE employee WITH replication = {'class': 'SimpleStrategy', 'replication_factor': '1'} AND durable_writes = true;
```

2. Create a column family by name Employee-Info with attributes Emp_Id Primary Key, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name

```
cqlsh:employee> CREATE TABLE Employee_Info( ... emp_id int PRIMARY KEY, ...  
emp_name text, ... designation text, ... date_of_joining timestamp, ... salary  
double, ... dept_name text ... );
```

```
cqlsh:employee> describe tables
```

```
employee_info
```



```
cqlsh:employee> describe table employee_info

CREATE TABLE employee.employee_info (
  emp_id int PRIMARY KEY,
  date_of_joining timestamp,
  dept_name text,
  designation text,
  emp_name text,
  salary double
) WITH additional_write_policy = '99p'
   AND bloom_filter_fp_chance = 0.01
   AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
   AND cdc = false
   AND comment = ''
   AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
   AND compression = {'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
   AND crc_check_chance = 1.0
   AND default_time_to_live = 0
   AND extensions = {}
   AND gc_grace_seconds = 864000
   AND max_index_interval = 2048
   AND memtable_flush_period_in_ms = 0
   AND min_index_interval = 128
   AND read_repair = 'BLOCKING'
   AND speculative_retry = '99p';
```

3. Insert the values into the table in batch cqlsh:employee>BEGIN BATCH

```
cqlsh:employees> BEGIN BATCH
```

```
    ... INSERT INTO
```

```
    ...
```

```
employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name
)
```

```
    ... values(124,'Pranav','Manager','2000-09-24',750000,'Export')
```

```
    ...
```

```
employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name
)
```

```
    ... values(125,'Anurag','AsstManager','2000-01-04',550000,'Export')
```

```
    ...
```

```
employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name
)
```

```
    ... values(126,'Prateek','HR','2000-05-04',650000,'HR')
```

```
    ... APPLY BATCH;
```

```
cqlsh:employee> select * from employee_info;
```

emp_id	date_of_joining	dept_name	designation	emp_name	salary
125	2000-01-03 18:30:00.000000+0000	Export	AsstManager	Anurag	5.5e+05
126	2000-05-03 18:30:00.000000+0000	HR	HR	Prateek	6.5e+05
124	2000-09-23 18:30:00.000000+0000	Export	Manager	Pranav	7.5e+05

4. Update Employee name and Department of Emp-Id 125

```
cqlsh:employees> update employee_info set dept_name='import' where  
emp_id=125;
```

```
cqlsh:employees> SELECT* FROM employee_info;
```

emp_id	date_of_joining	dept_name	designation	emp_name	salary
125	2000-01-03 18:30:00.000000+0000	import	AsstManager	Saurab	5.5e+05
126	2000-05-03 18:30:00.000000+0000	HR	HR	Prateek	6.5e+05
124	2000-09-23 18:30:00.000000+0000	Export	Manager	Pranav	7.5e+05

6. Alter the schema of the table Employee_Info to add a column Projects which stores a set of Projects done by the corresponding Employee.

```
cqlsh:employee> alter table employee_info ... add project text; cqlsh:employee>
select * from employee_info;
```

```
cqlsh:employees> ALTER TABLE employee_info add project set<text>;
```

```
cqlsh:employees> SELECT* FROM employee_info;
```

emp_id	date_of_joining	dept_name	designation	emp_name	project	salary
125	2000-01-03 18:30:00.000000+0000	import	AsstManager	Saurab	null	5.5e+05
126	2000-05-03 18:30:00.000000+0000	HR	HR	Prateek	null	6.5e+05
124	2000-09-23 18:30:00.000000+0000	Export	Manager	Pranav	null	7.5e+05

7. Update the altered table to add project names.

```
cqlsh:employees> update employee_info set project={'pro4555','pro2566'} where
emp_id=126;
```

```
cqlsh:employees> update employee_info set project={'pro45','pro25'} where
emp_id=124;
```

```
cqlsh:employees> update employee_info set project={'pro1','pro2'} where
emp_id=125;
```

```
cqlsh:employees> SELECT* FROM employee_info;
```

emp_id	date_of_joining	dept_name	designation	emp_name	project	salary
--------	-----------------	-----------	-------------	----------	---------	--------

-----+-----+-----+-----+-----
+-----

125 | 2000-01-03 18:30:00.000000+0000 | import | AsstManager | Saurab |
{'pro1', 'pro2'} | 5.5e+05

126 | 2000-05-03 18:30:00.000000+0000 | HR | HR | Prateek |
{'pro2566', 'pro4555'} | 6.5e+05

124 | 2000-09-23 18:30:00.000000+0000 | Export | Manager | Pranav |
{'pro25', 'pro45'} | 7.5e+05

LAB PROGRAM 3: Library database using Cassandra

1 Create a key space by name Library

```
cqlsh> create keyspace libraries with  
replication={'class':'SimpleStrategy','replication_factor':1};
```

```
cqlsh> use libraries;
```

2. Create a column family by name Library-Info with attributes Stud_Id Primary Key, Counter_value of type Counter, Stud_Name, Book-Name, Book-Id, Date_of_issue

```
cqlsh:libraries> CREATE TABLE library_info(Stud_id int, Stud_name text,  
Book_name text, Book_id int, Date_of_issue timestamp, counter_value counter,  
PRIMARY KEY(Stud_id,Stud_name,Book_name,Book_id,Date_of_issue));
```

3. Insert the values into the table in batch cqlsh:library

```
cqlsh:libraries> UPDATE library_info SET counter_value = counter_value + 1  
WHERE Stud_id = 123 AND Stud_name = 'Anurag' AND Book_name = 'BDA' AND  
Book_id = 455 AND Date_of_issue = '2000-09-24';
```

```
cqlsh:libraries> UPDATE library_info SET counter_value = counter_value + 1  
WHERE Stud_id = 123 AND Stud_name = 'Pranav' AND Book_name = 'ADS' AND  
Book_id = 45 AND Date_of_issue = '2003-05-04';
```

```
cqlsh:libraries> UPDATE library_info SET counter_value = counter_value + 1  
WHERE Stud_id = 123 AND Stud_name = 'Saurab' AND Book_name = 'CHY' AND  
Book_id = 245 AND Date_of_issue = '2003-05-07';
```

```
cqlsh:libraries> UPDATE library_info SET counter_value = counter_value + 1  
WHERE Stud_id = 123 AND Stud_name = 'Prateek' AND Book_name = 'CNS' AND  
Book_id = 25 AND Date_of_issue = '2003-05-09';cqlsh:libraries> select* from  
library_info;
```

stud_id	stud_name	book_name	book_id	date_of_issue	counter_value
123	Pranav	ADS	45	2003-05-03 18:30:00.000000+0000	1
123	Anurag	BDA	455	2000-09-23 18:30:00.000000+0000	1
123	Saurab	CHY	245	2003-05-06 18:30:00.000000+0000	1
123	Prateek	CNS	25	2003-05-08 18:30:00.000000+0000	1

(4 rows)

4. Display the details of the table created and increase the value of the counter

```
cqlsh:libraries> UPDATE library_info SET counter_value = counter_value + 1
WHERE Stud_id = 123 AND Stud_name = 'Prateek' AND Book_name = 'CNS' AND
Book_id = 25 AND Date_of_issue = '2003-05-09';
```

```
cqlsh:libraries> select* from library_info;
```

stud_id	stud_name	book_name	book_id	date_of_issue	counter_value
123	Pranav	ADS	45	2003-05-03 18:30:00.000000+0000	1
123	Anurag	BDA	455	2000-09-23 18:30:00.000000+0000	1
123	Saurab	CHY	245	2003-05-06 18:30:00.000000+0000	1

```
123 | Prateek | CNS | 25 | 2003-05-08 18:30:00.000000+0000 |
2
```

(4 rows)

5. Write a query to show that a student with id 1 has taken a book “BDA” 2 times.

```
cqlsh:libraries> UPDATE library_info SET counter_value = counter_value + 1
WHERE Stud_id = 123 AND Stud_name = 'Anurag' AND Book_name = 'BDA' AND
Book_id = 455 AND Date_of_issue = '2000-09-24';
```

```
cqlsh:libraries> select* from library_info;
```

```
stud_id | stud_name | book_name | book_id | date_of_issue          |
counter_value
-----+-----+-----+-----+-----+-----+-----
123 | Pranav | ADS | 45 | 2003-05-03 18:30:00.000000+0000 |      1
123 | Anurag | BDA | 455 | 2000-09-23 18:30:00.000000+0000 |
2
123 | Saurab | CHY | 245 | 2003-05-06 18:30:00.000000+0000 |
1
123 | Prateek | CNS | 25 | 2003-05-08 18:30:00.000000+0000 |
2
```

6. Export the created column to a csv file

```
cqlsh:lab2_library> copy library_info(stud_id,stud_name,book_id,date_of_issue,counter_value)to 'lib.csv';
Using 7 child processes

Starting copy of lab2_library.library_info with columns [stud_id, stud_name, book_id, date_of_issue, counter_v
alue].
Processed: 2 rows; Rate:      9 rows/s; Avg. rate:      9 rows/s
2 rows exported to 1 files in 0.250 seconds.
```

7. Import a given csv dataset from local file system into Cassandra column family

```
cqlsh:library>truncate library_info; cqlsh:library>copy
library_info(stud_id,stud_name,book_id,date_of_issue,counter_value) from
'lib.csv';
```


Lab Program 4:Hadoop Installation

```
[shashi@Shashis-MacBook-Air-2 ~ %] hadoop -version
ERROR: -version is not COMMAND nor fully qualified CLASSNAME.
Usage: hadoop [OPTIONS] SUBCOMMAND [SUBCOMMAND OPTIONS]
or hadoop [OPTIONS] CLASSNAME [CLASSNAME OPTIONS]
where CLASSNAME is a user-provided Java class

OPTIONS is none or any of:
--config dir      Hadoop config directory
--debug          turn on shell script debug mode
--help           usage information
buildpaths       attempt to add class files from build tree
hostnames list[,of,host,names] hosts to use in slave mode
hosts filename   list of hosts to use in slave mode
loglevel level   set the log4j level for this command
workers         turn on worker mode

SUBCOMMAND is one of:

Admin Commands:
daemonlog        get/set the log level for each daemon

Client Commands:
archive          creates a Hadoop archive
checknative      check native Hadoop and compression libraries availability
classpath        prints the class path needed to get the Hadoop jar and the
                  required libraries
conftest         validate configuration XML files
credential        interact with credential providers
distcp           distributed metadata changer
distcp           copy file or directories recursively
dtutil           operations related to delegation tokens
envvars          display computed Hadoop environment variables
fs              run a generic filesystem user client
gridmix          submit a mix of synthetic job, modeling a profiled from
                  production load
jar <jar>        run a jar file. NOTE: please use "yarn jar" to launch YARN
                  applications, not this command.
jnipath          prints the java.library.path
kdiag            Diagnose Kerberos Problems
kerbname         show auth_to_local principal conversion
key             manage keys via the KeyProvider
rumenfolder      scale a rumen input trace
rumentrace       convert logs into a rumen trace
s3guard          manage metadata on S3
trace            view and modify Hadoop tracing settings
version          print the version

Daemon Commands:
kms              run KMS, the Key Management Server
registrydns      run the registry DNS server

SUBCOMMAND may print help when invoked w/o parameters or with -h.
```

Lab Program 5: Execution of HDFS Commands for interaction with Hadoop Environment. (Minimum 10 commands to be executed)

```
c:\hadoop_new\sbin>hdfs dfs -mkdir /temp
```

```
c:\hadoop_new\sbin>hdfs dfs -copyFromLocal E:\Desktop\sample.txt \temp
```

```
c:\hadoop_new\sbin>hdfs dfs -ls \temp
```

Found 1 items

```
-rw-r--r-- 1 Admin supergroup 11 2021-06-11 21:12 /temp/sample.txt
```

```
c:\hadoop_new\sbin>hdfs dfs -cat \temp\sample.txt hello
```

world

```
c:\hadoop_new\sbin>hdfs dfs -get \temp\sample.txt E:\Desktop\temp
```

```
c:\hadoop_new\sbin>hdfs dfs -put E:\Desktop\temp \temp
```

```
c:\hadoop_new\sbin>hdfs dfs -ls \temp
```

Found 2 items

```
-rw-r--r-- 1 Admin supergroup 11 2021-06-11 21:12 /temp/sample.txt drwxr-xr-x -
```

```
Admin supergroup 0 2021-06-11 21:15 /temp/temp
```

```
c:\hadoop_new\sbin>hdfs dfs -mv \lab1 \temp
```

```
c:\hadoop_new\sbin>hdfs dfs -ls \temp Found 3 items drwxr-xr-x - Admin
```

```
supergroup 0 2021-04-19 15:07 /temp/lab1 -rw-r--r-- 1 Admin
```

7

```
supergroup 11 2021-06-11 21:12 /temp/sample.txt drwxr-xr-x -
```

```
Admin supergroup 0 2021-06-11 21:15 /temp/temp
```

```
c:\hadoop_new\sbin>hdfs dfs -rm /temp/sample.txt
```

Deleted /temp/sample.txt

```
c:\hadoop_new\sbin>hdfs dfs -ls \temp Found 2 items drwxr-xr-x - Admin
```

supergroup 0 2021-04-19 15:07 /temp/lab1 drwxr-xr-x - Admin

supergroup 0 2021-06-11 21:15 /temp/temp

c:\hadoop_new\sbin>hdfs dfs -copyFromLocal E:\Desktop\sample.txt \temp

c:\hadoop_new\sbin>hdfs dfs -ls \temp Found 3 items drwxr-xr-x - Admin

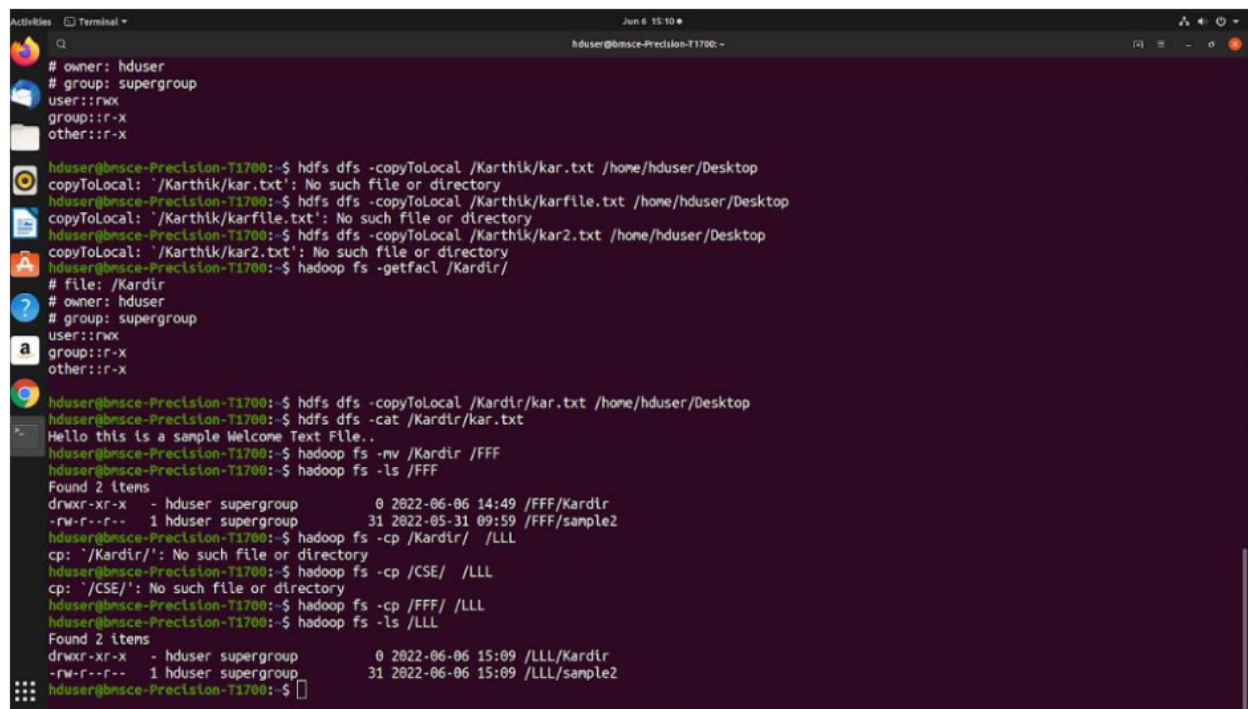
supergroup 0 2021-04-19 15:07 /temp/lab1 -rw-r--r-- 1 Admin supergroup

11 2021-06-11 21:17 /temp/sample.txt drwxr-xr-x - Admin supergroup 0

2021-06-11 21:15 /temp/temp

c:\hadoop_new\sbin>hdfs dfs -copyToLocal \temp\sample.txt

E:\Desktop\sample.txt



```
Activities Terminal Jun 6 15:10 hdsuser@bmsce-Precision-T1700 ~
# owner: hdsuser
# group: supergroup
user::rwx
group::r-x
other::r-x

hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /Karthik/kar.txt /home/hdsuser/Desktop
copyToLocal: '/Karthik/kar.txt': No such file or directory
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /Karthik/karfile.txt /home/hdsuser/Desktop
copyToLocal: '/Karthik/karfile.txt': No such file or directory
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /Karthik/kar2.txt /home/hdsuser/Desktop
copyToLocal: '/Karthik/kar2.txt': No such file or directory
hdsuser@bmsce-Precision-T1700:~$ hadoop fs -getfacl /Kardir/
# file: /Kardir
# owner: hdsuser
# group: supergroup
user::rwx
group::r-x
other::r-x

hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /Kardir/kar.txt /home/hdsuser/Desktop
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -cat /Kardir/kar.txt
Hello this is a sample Welcome Text File..
hdsuser@bmsce-Precision-T1700:~$ hadoop fs -mv /Kardir /FFF
hdsuser@bmsce-Precision-T1700:~$ hadoop fs -ls /FFF
Found 2 items
drwxr-xr-x - hdsuser supergroup 0 2022-06-06 14:49 /FFF/Kardir
-rw-r--r-- 1 hdsuser supergroup 31 2022-05-31 09:59 /FFF/sample2
hdsuser@bmsce-Precision-T1700:~$ hadoop fs -cp /Kardir/ /LLL
cp: '/Kardir/': No such file or directory
hdsuser@bmsce-Precision-T1700:~$ hadoop fs -cp /CSE/ /LLL
cp: '/CSE/': No such file or directory
hdsuser@bmsce-Precision-T1700:~$ hadoop fs -cp /FFF/ /LLL
hdsuser@bmsce-Precision-T1700:~$ hadoop fs -ls /LLL
Found 2 items
drwxr-xr-x - hdsuser supergroup 0 2022-06-06 15:09 /LLL/Kardir
-rw-r--r-- 1 hdsuser supergroup 31 2022-06-06 15:09 /LLL/sample2
hdsuser@bmsce-Precision-T1700:~$
```

```
Activities Terminal Jun 6 15:05 hduser@bmsce-Precision-T1700: -
command 'fs' from deb openafs-client (1.8.4-pre1-1ubuntu2.4)

Try: sudo apt install <deb name>

hduser@bmsce-Precision-T1700:~$ hdfs dfs -getmerge /Kardir/kar.txt /Kardir/kar2.txt /home/hduser/Desktop/Merge.txt
getmerge: '/Kardir/kar2.txt': No such file or directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs -getmerge /Kardir/kar.txt /Kardir/kar.txt /home/hduser/Desktop/Merge.txt
hduser@bmsce-Precision-T1700:~$ hadoop fs -getfacl /Karthik/
# file: /Karthik
# owner: hduser
# group: supergroup
user::rwx
group::r-x
other::r-x

hduser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /Karthik/kar.txt /home/hduser/Desktop
copyToLocal: '/Karthik/kar.txt': No such file or directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /Karthik/karfile.txt /home/hduser/Desktop
copyToLocal: '/Karthik/karfile.txt': No such file or directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /Karthik/kar2.txt /home/hduser/Desktop
copyToLocal: '/Karthik/kar2.txt': No such file or directory
hduser@bmsce-Precision-T1700:~$ hadoop fs -getfacl /Kardir/
# file: /Kardir
# owner: hduser
# group: supergroup
user::rwx
group::r-x
other::r-x

hduser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /Kardir/kar.txt /home/hduser/Desktop
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /Kardir/kar.txt
Hello this is a sample Welcome Text File..
hduser@bmsce-Precision-T1700:~$ hadoop fs -mv /Kardir /FFF
hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /FFF
Found 2 items
drwxr-xr-x - hduser supergroup 0 2022-06-06 14:49 /FFF/Kardir
-rw-r--r-- 1 hduser supergroup 31 2022-05-31 09:59 /FFF/sample2
hduser@bmsce-Precision-T1700:~$ hadoop fs -cp /Kardir/ /LLL
cp: '/Kardir/': No such file or directory
```

LAB 6 : For the given file, Create a Map Reduce program to

a) Find the average temperature for each year from the NCDC data set.

· **Program**

AverageDriver

```
package temp;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class AverageDriver {

    public static void main(String[] args) throws Exception {

        if (args.length != 2) {

            System.err.println("&quot;Please Enter the input and output
            parameters&quot;");

            System.exit(-1);

        }

        Job job = new Job();

        job.setJarByClass(AverageDriver.class);

        job.setJobName("&quot;Max temperature&quot;");
```

```

FileInputFormat.addInputPath(job, new Path(args[0]));

FileOutputFormat.setOutputPath(job, new Path(args[1]));

job.setMapperClass(AverageMapper.class);

job.setReducerClass(AverageReducer.class);

job.setOutputKeyClass(Text.class);


job.setOutputValueClass(IntWritable.class);

System.exit(job.waitForCompletion(true) ? 0 : 1);

}

}

```

AverageMapper

```

package temp;

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

public class AverageMapper extends Mapper<LongWritable, Text,
Text, IntWritable> {

public static final int MISSING = 9999;

public void map(LongWritable key, Text value,

```

```
Mapper<LongWritable, Text, Text, IntWritable>.Context context)
throws IOException, InterruptedException {
    int temperature;

    String line = value.toString();

    String year = line.substring(15, 19);

    if (line.charAt(87) == '&#39;+&#39;') {
        temperature = Integer.parseInt(line.substring(88, 92));
    } else {
        temperature = Integer.parseInt(line.substring(87, 92));
    }

    String quality = line.substring(92, 93);

    if (temperature != 9999 && quality.matches("&quot;[01459]&quot;"))
        context.write(new Text(year), new
            IntWritable(temperature));
    }
}
```

AverageReducer

```
package temp;

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class AverageReducer extends Reducer<Text, IntWritable,
Text, IntWritable> {

    public void reduce(Text key, Iterable<IntWritable> values,
        Reducer<Text, IntWritable, Text, IntWritable>.Context context)
        throws IOException, InterruptedException {

        int max_temp = 0;

        int count = 0;

        for (IntWritable value : values) {

            max_temp += value.get();

            count++;

        }

        context.write(key, new IntWritable(max_temp / count));

    }

}
```


- **Output**

```
hduser@bmsce-Precision-T1700:~$ sudo su hduser
```

```
[sudo] password for hduser:
```

```
hduser@bmsce-Precision-T1700:~$ start-all.sh
```

```
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
```

```
Starting namenodes on [localhost]
```

```
hduser@localhost's password:
```

```
localhost: starting namenode, logging to  
/usr/local/hadoop/logs/hadoop-hduser-namenode-bmsce-Precision-T1700.out
```

```
hduser@localhost's password:
```

```
localhost: starting datanode, logging to  
/usr/local/hadoop/logs/hadoop-hduser-datanode-bmsce-Precision-T1700.out
```

```
Starting secondary namenodes [0.0.0.0]
```

```
hduser@0.0.0.0's password:
```

```
0.0.0.0: starting secondarynamenode, logging to  
/usr/local/hadoop/logs/hadoop-hduser-secondarynamenode-bmsce-Precision-T1700.out
```

```
starting yarn daemons
```

```
starting resourcemanager, logging to  
/usr/local/hadoop/logs/yarn-hduser-resourcemanager-bmsce-Precision-T1700.out
```

```
hduser@localhost's password:
```

localhost: starting nodemanager, logging to
/usr/local/hadoop/logs/yarn-hduser-nodemanager-bmsce-Precision-T1700.out

hduser@bmsce-Precision-T1700:~\$ jps

7376 DataNode

8212 Jps

8090 NodeManager

3725 org.eclipse.equinox.launcher_1.5.600.v20191014-2022.jar

7758 ResourceManager

7199 NameNode

7599 SecondaryNameNode

hduser@bmsce-Precision-T1700:~\$ hadoop fs -mkdir /input_kundana

hduser@bmsce-Precision-T1700:~\$ hadoop fs -put Downloads/1901
/input_kundana/1901.txt

hduser@bmsce-Precision-T1700:~\$ hadoop jar Desktop/temp.jar
Temperature.AverageDriver /input_kundana/1901.txt /output_1901

Exception in thread "main" java.lang.ClassNotFoundException:
Temperature.AverageDriver

at java.net.URLClassLoader.findClass(URLClassLoader.java:382)

at java.lang.ClassLoader.loadClass(ClassLoader.java:418)

at java.lang.ClassLoader.loadClass(ClassLoader.java:351)

at java.lang.Class.forName0(Native Method)

at java.lang.Class.forName(Class.java:348)

at org.apache.hadoop.util.RunJar.run(RunJar.java:214)

at org.apache.hadoop.util.RunJar.main(RunJar.java:136)

hduser@bmsce-Precision-T1700:~\$ hadoop jar Desktop/temp.jar AverageDriver
/input_kundana/1901.txt /output_1901

22/06/21 10:26:05 INFO Configuration.deprecation: session.id is deprecated.
Instead, use dfs.metrics.session-id

22/06/21 10:26:05 INFO jvm.JvmMetrics: Initializing JVM Metrics with
processName=JobTracker, sessionId=

22/06/21 10:26:05 WARN mapreduce.JobSubmitter: Hadoop command-line
option parsing not performed. Implement the Tool interface and execute your
application with ToolRunner to remedy this.

22/06/21 10:26:05 INFO input.FileInputFormat: Total input paths to process : 1

22/06/21 10:26:05 INFO mapreduce.JobSubmitter: number of splits:1

22/06/21 10:26:05 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_local1195965365_0001

22/06/21 10:26:05 INFO mapreduce.Job: The url to track the job:
<http://localhost:8080/>

22/06/21 10:26:05 INFO mapreduce.Job: Running job: job_local1195965365_0001

22/06/21 10:26:05 INFO mapred.LocalJobRunner: OutputCommitter set in config
null

22/06/21 10:26:05 INFO mapred.LocalJobRunner: OutputCommitter is
org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter

22/06/21 10:26:05 INFO mapred.LocalJobRunner: Waiting for map tasks

22/06/21 10:26:05 INFO mapred.LocalJobRunner: Starting task:
attempt_local1195965365_0001_m_000000_0

22/06/21 10:26:05 INFO mapred.Task: Using ResourceCalculatorProcessTree : []

22/06/21 10:26:05 INFO mapred.MapTask: Processing split:
hdfs://localhost:54310/input_kundana/1901.txt:0+888190

22/06/21 10:26:06 INFO mapred.MapTask: (EQUATOR) 0 kvi
26214396(104857584)

22/06/21 10:26:06 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100

22/06/21 10:26:06 INFO mapred.MapTask: soft limit at 83886080

22/06/21 10:26:06 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600

22/06/21 10:26:06 INFO mapred.MapTask: kvstart = 26214396; length = 6553600

22/06/21 10:26:06 INFO mapred.MapTask: Map output collector class =
org.apache.hadoop.mapred.MapTask\$MapOutputBuffer

22/06/21 10:26:06 INFO mapred.LocalJobRunner:

22/06/21 10:26:06 INFO mapred.MapTask: Starting flush of map output

22/06/21 10:26:06 INFO mapred.MapTask: Spilling map output

22/06/21 10:26:06 INFO mapred.MapTask: bufstart = 0; bufend = 59076; bufvoid =
104857600

22/06/21 10:26:06 INFO mapred.MapTask: kvstart = 26214396(104857584);
kvend = 26188144(104752576); length = 26253/6553600

22/06/21 10:26:06 INFO mapred.MapTask: Finished spill 0

22/06/21 10:26:06 INFO mapred.Task:
Task:attempt_local1195965365_0001_m_000000_0 is done. And is in the process
of committing

22/06/21 10:26:06 INFO mapred.LocalJobRunner: map

22/06/21 10:26:06 INFO mapred.Task: Task
'attempt_local1195965365_0001_m_000000_0' done.

22/06/21 10:26:06 INFO mapred.LocalJobRunner: Finishing task:
attempt_local1195965365_0001_m_000000_0

22/06/21 10:26:06 INFO mapred.LocalJobRunner: map task executor complete.

22/06/21 10:26:06 INFO mapred.LocalJobRunner: Waiting for reduce tasks

22/06/21 10:26:06 INFO mapred.LocalJobRunner: Starting task:
attempt_local1195965365_0001_r_000000_0

22/06/21 10:26:06 INFO mapred.Task: Using ResourceCalculatorProcessTree : []

22/06/21 10:26:06 INFO mapred.ReduceTask: Using ShuffleConsumerPlugin:
org.apache.hadoop.mapreduce.task.reduce.Shuffle@65367f35

22/06/21 10:26:06 INFO reduce.MergeManagerImpl: MergerManager:
memoryLimit=349752512, maxSingleShuffleLimit=87438128,
mergeThreshold=230836672, ioSortFactor=10,
memToMemMergeOutputsThreshold=10

22/06/21 10:26:06 INFO reduce.EventFetcher:
attempt_local1195965365_0001_r_000000_0 Thread started: EventFetcher for
fetching Map Completion Events

22/06/21 10:26:06 INFO reduce.LocalFetcher: localfetcher#1 about to shuffle
output of map attempt_local1195965365_0001_m_000000_0 decomp: 72206
len: 72210 to MEMORY

22/06/21 10:26:06 INFO reduce.InMemoryMapOutput: Read 72206 bytes from
map-output for attempt_local1195965365_0001_m_000000_0

22/06/21 10:26:06 INFO reduce.MergeManagerImpl: closeInMemoryFile ->
map-output of size: 72206, inMemoryMapOutputs.size() -> 1, commitMemory ->
0, usedMemory ->72206

22/06/21 10:26:06 INFO reduce.EventFetcher: EventFetcher is interrupted..
Returning

22/06/21 10:26:06 INFO mapred.LocalJobRunner: 1 / 1 copied.

22/06/21 10:26:06 INFO reduce.MergeManagerImpl: finalMerge called with 1 in-memory map-outputs and 0 on-disk map-outputs

22/06/21 10:26:06 INFO mapred.Merger: Merging 1 sorted segments

22/06/21 10:26:06 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of total size: 72199 bytes

22/06/21 10:26:06 INFO reduce.MergeManagerImpl: Merged 1 segments, 72206 bytes to disk to satisfy reduce memory limit

22/06/21 10:26:06 INFO reduce.MergeManagerImpl: Merging 1 files, 72210 bytes from disk

22/06/21 10:26:06 INFO reduce.MergeManagerImpl: Merging 0 segments, 0 bytes from memory into reduce

22/06/21 10:26:06 INFO mapred.Merger: Merging 1 sorted segments

22/06/21 10:26:06 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of total size: 72199 bytes

22/06/21 10:26:06 INFO mapred.LocalJobRunner: 1 / 1 copied.

22/06/21 10:26:06 INFO Configuration.deprecation: mapred.skip.on is deprecated. Instead, use mapreduce.job.skiprecords

22/06/21 10:26:06 INFO mapred.Task:

Task:attempt_local1195965365_0001_r_000000_0 is done. And is in the process of committing

22/06/21 10:26:06 INFO mapred.LocalJobRunner: 1 / 1 copied.

22/06/21 10:26:06 INFO mapred.Task: Task

attempt_local1195965365_0001_r_000000_0 is allowed to commit now

22/06/21 10:26:06 INFO output.FileOutputCommitter: Saved output of task

'attempt_local1195965365_0001_r_000000_0' to

hdfs://localhost:54310/output_1901/_temporary/0/task_local1195965365_0001_r_000000

22/06/21 10:26:06 INFO mapred.LocalJobRunner: reduce > reduce

22/06/21 10:26:06 INFO mapred.Task: Task
'attempt_local1195965365_0001_r_000000_0' done.

22/06/21 10:26:06 INFO mapred.LocalJobRunner: Finishing task:
attempt_local1195965365_0001_r_000000_0

22/06/21 10:26:06 INFO mapred.LocalJobRunner: reduce task executor complete.

22/06/21 10:26:06 INFO mapreduce.Job: Job job_local1195965365_0001 running
in uber mode : false

22/06/21 10:26:06 INFO mapreduce.Job: map 100% reduce 100%

22/06/21 10:26:06 INFO mapreduce.Job: Job job_local1195965365_0001
completed successfully

22/06/21 10:26:06 INFO mapreduce.Job: Counters: 38

File System Counters

FILE: Number of bytes read=152940

FILE: Number of bytes written=725372

FILE: Number of read operations=0

FILE: Number of large read operations=0

FILE: Number of write operations=0

HDFS: Number of bytes read=1776380

HDFS: Number of bytes written=8

HDFS: Number of read operations=13

HDFS: Number of large read operations=0

HDFS: Number of write operations=4

Map-Reduce Framework

Map input records=6565

Map output records=6564

Map output bytes=59076

Map output materialized bytes=72210

Input split bytes=110

Combine input records=0

Combine output records=0

Reduce input groups=1

Reduce shuffle bytes=72210

Reduce input records=6564

Reduce output records=1

Spilled Records=13128

Shuffled Maps =1

Failed Shuffles=0

Merged Map outputs=1

GC time elapsed (ms)=63

CPU time spent (ms)=0

Physical memory (bytes) snapshot=0

Virtual memory (bytes) snapshot=0

Total committed heap usage (bytes)=999292928

Shuffle Errors

BAD_ID=0

CONNECTION=0

IO_ERROR=0

WRONG_LENGTH=0

WRONG_MAP=0

WRONG_REDUCE=0

File Input Format Counters

Bytes Read=888190

File Output Format Counters

Bytes Written=8

```
hduser@bmsce-Precision-T1700:~$ hadoop fs -cat /output_1901/part-r-00000
```

```
1901 46
```

```
hduser@bmsce-Precision-T1700:~$
```

b) find the mean max temperature for every month

.

MeanMaxDriver.class

```
package meanmax;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;
```

```
import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class MeanMaxDriver {

    public static void main(String[] args) throws Exception {

        if (args.length != 2) {

            System.err.println("&quot;Please Enter the input and output  
parameters&quot;");

            System.exit(-1);

        }

        Job job = new Job();

        job.setJarByClass(MeanMaxDriver.class);

        job.setJobName("&quot;Max temperature&quot;");

        FileInputFormat.addInputPath(job, new Path(args[0]));

        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        job.setMapperClass(MeanMaxMapper.class);

        job.setReducerClass(MeanMaxReducer.class);

        job.setOutputKeyClass(Text.class);

        job.setOutputValueClass(IntWritable.class);

        System.exit(job.waitForCompletion(true) ? 0 : 1);

    }

}
```

```
}
```

```
MeanMaxMapper.class
```

```
package meanmax;
```

```
import java.io.IOException;
```

```
import org.apache.hadoop.io.IntWritable;
```

```
import org.apache.hadoop.io.LongWritable;
```

```
import org.apache.hadoop.io.Text;
```

```
import org.apache.hadoop.mapreduce.Mapper;
```

```
public class MeanMaxMapper extends Mapper<LongWritable, Text,
```

```
Text, IntWritable> {
```

```
    public static final int MISSING = 9999;
```

```
    public void map(LongWritable key, Text value,
```

```
Mapper<LongWritable, Text, Text, IntWritable>.Context context)
```

```
throws IOException, InterruptedException {
```

```
        int temperature;
```

```
        String line = value.toString();
```

```
        String month = line.substring(19, 21);
```

```
        if (line.charAt(87) == '+'&#39;+&#39;) {
```

```
            temperature = Integer.parseInt(line.substring(88, 92));
```

```
        } else {
```

```

temperature = Integer.parseInt(line.substring(87, 92));
}
String quality = line.substring(92, 93);
if (temperature != 9999 && quality.matches(""[01459]"))
context.write(new Text(month), new
IntWritable(temperature));
}
}

```

MeanMaxReducer.class

```

package meanmax;

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

public class MeanMaxReducer extends Reducer<Text, IntWritable,
Text, IntWritable> {

    public void reduce(Text key, Iterable<IntWritable> values,
        Reducer<Text, IntWritable, Text, IntWritable>.Context context)
        throws IOException, InterruptedException {

        int max_temp = 0;

        int total_temp = 0;
    }
}

```

```
int count = 0;

int days = 0;

for (IntWritable value : values) {

    int temp = value.get();

    if (temp > max_temp)

        max_temp = temp;

    count++;

    if (count == 3) {

        total_temp += max_temp;

        max_temp = 0;

        count = 0;

        days++;

    }

}

context.write(key, new IntWritable(total_temp / days));

}
```

- **Output**

```
hduser@bmsce-OptiPlex-3060:~$ hadoop jar  
/home/hduser/Desktop/mean_max_temp.jar meanmax.MeanMaxDriver  
/input_pranav/temp_1901.txt /avg_temp_output_meanmax_1901
```

22/06/21 10:17:01 INFO Configuration.deprecation: session.id is deprecated.
Instead, use dfs.metrics.session-id

22/06/21 10:17:01 INFO jvm.JvmMetrics: Initializing JVM Metrics with
processName=JobTracker, sessionId=

22/06/21 10:17:01 WARN mapreduce.JobSubmitter: Hadoop command-line
option parsing not performed. Implement the Tool interface and execute your
application with ToolRunner to remedy this.

22/06/21 10:17:01 INFO input.FileInputFormat: Total input paths to process : 1

22/06/21 10:17:01 INFO mapreduce.JobSubmitter: number of splits:1

22/06/21 10:17:01 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_local232634845_0001

22/06/21 10:17:01 INFO mapreduce.Job: The url to track the job:
<http://localhost:8080/>

22/06/21 10:17:01 INFO mapreduce.Job: Running job: job_local232634845_0001

22/06/21 10:17:01 INFO mapred.LocalJobRunner: OutputCommitter set in config
null

22/06/21 10:17:01 INFO mapred.LocalJobRunner: OutputCommitter is
org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter

22/06/21 10:17:01 INFO mapred.LocalJobRunner: Waiting for map tasks

22/06/21 10:17:01 INFO mapred.LocalJobRunner: Starting task:
attempt_local232634845_0001_m_000000_0

22/06/21 10:17:01 INFO mapred.Task: Using ResourceCalculatorProcessTree : []

22/06/21 10:17:01 INFO mapred.MapTask: Processing split:
hdfs://localhost:54310/input_pranav/temp_1901.txt:0+888190

22/06/21 10:17:01 INFO mapred.MapTask: (EQUATOR) 0 kvi
26214396(104857584)

22/06/21 10:17:01 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100

22/06/21 10:17:01 INFO mapred.MapTask: soft limit at 83886080

22/06/21 10:17:01 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600

22/06/21 10:17:01 INFO mapred.MapTask: kvstart = 26214396; length = 6553600

22/06/21 10:17:01 INFO mapred.MapTask: Map output collector class =
org.apache.hadoop.mapred.MapTask\$MapOutputBuffer

22/06/21 10:17:01 INFO mapred.LocalJobRunner:

22/06/21 10:17:01 INFO mapred.MapTask: Starting flush of map output

22/06/21 10:17:01 INFO mapred.MapTask: Spilling map output

22/06/21 10:17:01 INFO mapred.MapTask: bufstart = 0; bufend = 45948; bufvoid =
104857600

22/06/21 10:17:01 INFO mapred.MapTask: kvstart = 26214396(104857584);
kvend = 26188144(104752576); length = 26253/6553600

22/06/21 10:17:01 INFO mapred.MapTask: Finished spill 0

22/06/21 10:17:01 INFO mapred.Task:
Task:attempt_local232634845_0001_m_000000_0 is done. And is in the process
of committing

22/06/21 10:17:01 INFO mapred.LocalJobRunner: map

22/06/21 10:17:01 INFO mapred.Task: Task
'attempt_local232634845_0001_m_000000_0' done.

22/06/21 10:17:01 INFO mapred.LocalJobRunner: Finishing task:
attempt_local232634845_0001_m_000000_0

22/06/21 10:17:01 INFO mapred.LocalJobRunner: map task executor complete.

22/06/21 10:17:01 INFO mapred.LocalJobRunner: Waiting for reduce tasks

22/06/21 10:17:01 INFO mapred.LocalJobRunner: Starting task:
attempt_local232634845_0001_r_000000_0

22/06/21 10:17:01 INFO mapred.Task: Using ResourceCalculatorProcessTree : []

22/06/21 10:17:01 INFO mapred.ReduceTask: Using ShuffleConsumerPlugin:
org.apache.hadoop.mapreduce.task.reduce.Shuffle@1a055244

22/06/21 10:17:01 INFO reduce.MergeManagerImpl: MergerManager:
memoryLimit=349752512, maxSingleShuffleLimit=87438128,
mergeThreshold=230836672, ioSortFactor=10,
memToMemMergeOutputsThreshold=10

22/06/21 10:17:01 INFO reduce.EventFetcher:
attempt_local232634845_0001_r_000000_0 Thread started: EventFetcher for
fetching Map Completion Events

22/06/21 10:17:01 INFO reduce.LocalFetcher: localfetcher#1 about to shuffle
output of map attempt_local232634845_0001_m_000000_0 decomp: 59078 len:
59082 to MEMORY

22/06/21 10:17:01 INFO reduce.InMemoryMapOutput: Read 59078 bytes from
map-output for attempt_local232634845_0001_m_000000_0

22/06/21 10:17:01 INFO reduce.MergeManagerImpl: closeInMemoryFile ->
map-output of size: 59078, inMemoryMapOutputs.size() -> 1, commitMemory ->
0, usedMemory ->59078

22/06/21 10:17:01 INFO reduce.EventFetcher: EventFetcher is interrupted..
Returning

22/06/21 10:17:01 INFO mapred.LocalJobRunner: 1 / 1 copied.

22/06/21 10:17:01 INFO reduce.MergeManagerImpl: finalMerge called with 1 in-memory map-outputs and 0 on-disk map-outputs

22/06/21 10:17:01 INFO mapred.Merger: Merging 1 sorted segments

22/06/21 10:17:01 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of total size: 59073 bytes

22/06/21 10:17:01 INFO reduce.MergeManagerImpl: Merged 1 segments, 59078 bytes to disk to satisfy reduce memory limit

22/06/21 10:17:01 INFO reduce.MergeManagerImpl: Merging 1 files, 59082 bytes from disk

22/06/21 10:17:01 INFO reduce.MergeManagerImpl: Merging 0 segments, 0 bytes from memory into reduce

22/06/21 10:17:01 INFO mapred.Merger: Merging 1 sorted segments

22/06/21 10:17:01 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of total size: 59073 bytes

22/06/21 10:17:01 INFO mapred.LocalJobRunner: 1 / 1 copied.

22/06/21 10:17:01 INFO Configuration.deprecation: mapred.skip.on is deprecated. Instead, use mapreduce.job.skiprecords

22/06/21 10:17:01 INFO mapred.Task:

Task:attempt_local232634845_0001_r_000000_0 is done. And is in the process of committing

22/06/21 10:17:01 INFO mapred.LocalJobRunner: 1 / 1 copied.

22/06/21 10:17:01 INFO mapred.Task: Task

attempt_local232634845_0001_r_000000_0 is allowed to commit now

22/06/21 10:17:01 INFO output.FileOutputCommitter: Saved output of task

'attempt_local232634845_0001_r_000000_0' to

hdfs://localhost:54310/avg_temp_output_meanmax_1901/_temporary/0/task_local232634845_0001_r_000000

22/06/21 10:17:01 INFO mapred.LocalJobRunner: reduce > reduce

22/06/21 10:17:01 INFO mapred.Task: Task
'attempt_local232634845_0001_r_000000_0' done.

22/06/21 10:17:01 INFO mapred.LocalJobRunner: Finishing task:
attempt_local232634845_0001_r_000000_0

22/06/21 10:17:01 INFO mapred.LocalJobRunner: reduce task executor complete.

22/06/21 10:17:02 INFO mapreduce.Job: Job job_local232634845_0001 running
in uber mode : false

22/06/21 10:17:02 INFO mapreduce.Job: map 100% reduce 100%

22/06/21 10:17:02 INFO mapreduce.Job: Job job_local232634845_0001
completed successfully

22/06/21 10:17:02 INFO mapreduce.Job: Counters: 38

File System Counters

FILE: Number of bytes read=125588

FILE: Number of bytes written=682332

FILE: Number of read operations=0

FILE: Number of large read operations=0

FILE: Number of write operations=0

HDFS: Number of bytes read=1776380

HDFS: Number of bytes written=74

HDFS: Number of read operations=13

HDFS: Number of large read operations=0

HDFS: Number of write operations=4

Map-Reduce Framework

Map input records=6565

Map output records=6564

Map output bytes=45948

Map output materialized bytes=59082

Input split bytes=114

Combine input records=0

Combine output records=0

Reduce input groups=12

Reduce shuffle bytes=59082

Reduce input records=6564

Reduce output records=12

Spilled Records=13128

Shuffled Maps =1

Failed Shuffles=0

Merged Map outputs=1

GC time elapsed (ms)=54

CPU time spent (ms)=0

Physical memory (bytes) snapshot=0

Virtual memory (bytes) snapshot=0

Total committed heap usage (bytes)=999292928

Shuffle Errors

BAD_ID=0

CONNECTION=0

IO_ERROR=0

WRONG_LENGTH=0

WRONG_MAP=0

WRONG_REDUCE=0

File Input Format Counters

Bytes Read=888190

File Output Format Counters

Bytes Written=74

```
hduser@bmsce-OptiPlex-3060:~$ hdfs dfs -ls /avg_temp_meanmax_output
```

```
ls: `/avg_temp_meanmax_output': No such file or directory
```

```
hduser@bmsce-OptiPlex-3060:~$ hdfs dfs -ls /avg_temp_output_meanmax_1901
```

```
Found 2 items
```

```
-rw-r--r--  1 hduser supergroup      0 2022-06-21 10:17
```

```
/avg_temp_output_meanmax_1901/_SUCCESS
```

```
-rw-r--r--  1 hduser supergroup    74 2022-06-21 10:17
```

```
/avg_temp_output_meanmax_1901/part-r-00000
```

```
hduser@bmsce-OptiPlex-3060:~$ hdfs dfs -cat
```

```
/avg_temp_output_meanmax/part-r-00000
```

```
cat: `/avg_temp_output_meanmax/part-r-00000': No such file or directory
```

```
hduser@bmsce-OptiPlex-3060:~$ hdfs dfs -cat  
/avg_temp_output_meanmax_1901/part-r-00000
```

01	4
02	0
03	7
04	44
05	100
06	168
07	219
08	198
09	141
10	100
11	19
12	3

LAB 7

For a given Text file, create a Map Reduce program to sort the content in an alphabetic order listing only top 'n' maximum occurrence of words.

// TopN.java package sortWords;

```
import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.fs.Path;
import
```

```
org.apache.hadoop.io.IntWritable; import org.apache.hadoop.io.Text; import
```

```
org.apache.hadoop.mapreduce.Job; import
```

```
org.apache.hadoop.mapreduce.Mapper; import
```

```
org.apache.hadoop.mapreduce.Reducer; import
```

```
org.apache.hadoop.mapreduce.lib.input.FileInputFormat; import
```

```
org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; import
```

```
org.apache.hadoop.util.GenericOptionsParser; import utils.MiscUtils;
```

```
import java.io.IOException; import java.util.*;
```

```
public class TopN {
```

```
    public static void main(String[] args) throws Exception {
```

```
        Configuration conf = new Configuration();
```

```
        String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs(); if
```

```
(otherArgs.length != 2) {
```

```
    System.err.println("Usage: TopN <in> <out>");
```

```
    System.exit(2);
```

```
}
```

```
    Job job = Job.getInstance(conf); job.setJobName("Top N");
```

```
    job.setJarByClass(TopN.class);
```

```
    job.setMapperClass(TopNMapper.class);
```

```
    //job.setCombinerClass(TopNReducer.class);
```

```

job.setReducerClass(TopNReducer.class); job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(job, new Path(otherArgs[0]));
FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));
System.exit(job.waitForCompletion(true) ? 0 : 1);
}

/**
 * The mapper reads one line at the time, splits it into an array of single words and
 * emits every
 * word to the reducers with the value of 1.
 */

public static class TopNMapper extends Mapper<Object, Text, Text, IntWritable> {
    private final static IntWritable one = new IntWritable(1); private Text word = new
    Text();

    private String tokens = "[_|$#<>\\^=\\[\\]\\*\\/\\\\\\.,;\\.\\-:()?!\\\"'"]";

    @Override
    17
    public void map(Object key, Text value, Context context) throws IOException,
    InterruptedException {

        String cleanLine = value.toString().toLowerCase().replaceAll(tokens, " ");
        StringTokenizer itr
        = new StringTokenizer(cleanLine); while (itr.hasMoreTokens()) {
            word.set(itr.nextToken().trim()); context.write(word, one);
        }
    }
}

```

```
}
```

```
/**
```

* The reducer retrieves every word and puts it into a Map: if the word already exists in the * map,

increments its value, otherwise sets it to 1.

```
*/
```

```
public static class TopNReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
```

```
    private Map<Text, IntWritable> countMap = new HashMap<>();
```

```
    @Override
```

```
    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException,
```

```
    InterruptedException {
```

```
        // computes the number of occurrences of a single word int sum = 0; for
```

```
        (IntWritable val : values) { sum += val.get();
```

```
    }
```

```
    // puts the number of occurrences of this word into the map.
```

```
    // We need to create another Text object because the Text instance
```

```
    // we receive is the same for all the words countMap.put(new Text(key), new IntWritable(sum));
```

```
}
```

```
@Override
```

```
    protected void cleanup(Context context) throws IOException, InterruptedException {
```

```
        Map<Text, IntWritable> sortedMap = MiscUtils.sortByValues(countMap);
```

```
        int counter = 0; for (Text key : sortedMap.keySet()) { if (counter++ == 3) {
```



```

break;
}
context.write(key, sortedMap.get(key));
}
}
}
/**
 * The combiner retrieves every word and puts it into a Map: if the word already
 * exists in the *
 * map, increments its value, otherwise sets it to 1.
 */
public static class TopNCombiner extends Reducer<Text, IntWritable, Text,
IntWritable> {
18
    @Override
    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws
IOException,
InterruptedException {
    // computes the number of occurrences of a single word
    int sum = 0;
    for (IntWritable val : values) {
        sum += val.get();
    }
    context.write(key, new IntWritable(sum));
}
}
}
}

```

```

// MiscUtils.java package utils;

import java.util.*;

public class MiscUtils {

    /**
    sorts the map by values. Taken from:
    http://javarevisited.blogspot.it/2012/12/how-to-sort-hashmap-java-by-key-and-value.html
    */

    public static <K extends Comparable, V extends Comparable> Map<K, V>
    sortByValues(Map<K, V>
    map) {

        List<Map.Entry<K, V>> entries = new LinkedList<Map.Entry<K,
        V>>(map.entrySet());

        Collections.sort(entries, new Comparator<Map.Entry<K, V>>() {

            @Override public int compare(Map.Entry<K, V> o1, Map.Entry<K, V> o2) { return
            o2.getValue().compareTo(o1.getValue());

            }

        });

        //LinkedHashMap will keep the keys in the order they are inserted
        //which is currently sorted on natural ordering

        Map<K, V> sortedMap = new LinkedHashMap<K, V>();

        for (Map.Entry<K, V> entry : entries) {

            sortedMap.put(entry.getKey(), entry.getValue());

        }
    }
}

```

```
return sortedMap;
```

```
}
```

```
}
```

```
C:\hadoop_new\share\hadoop\mapreduce>hdfs dfs -cat \sortwordsOutput\part-r-00000  
car      7  
deer     6  
bear     3
```

LAB 8:-Create a Hadoop Map Reduce program to combine information from the users file along with Information from the posts file by using the concept of join and display user_id, Reputation and Score.

```
// JoinDriver.java import org.apache.hadoop.conf.Configured; import
org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.Text; import org.apache.hadoop.mapred.*; import
org.apache.hadoop.mapred.lib.MultipleInputs; import org.apache.hadoop.util.*;

public class JoinDriver extends Configured implements Tool {

    public static class KeyPartitioner implements Partitioner<TextPair, Text> {

        @Override

        public void configure(JobConf job) {}

        @Override

        public int getPartition(TextPair key, Text value, int numPartitions) { return
(key.getFirst().hashCode() & Integer.MAX_VALUE) % numPartitions;

        }

        }

        @Override public int run(String[] args) throws Exception { if (args.length != 3) {
System.out.println("Usage: <Department Emp Strength input>
<Department Name input> <output>");

return -1;

        }

        JobConf conf = new JobConf(getConf(), getClass()); conf.setJobName("Join
'Department Emp Strength input' with 'Department Name input'");

        Path AInputPath = new Path(args[0]);
```

```
Path BInputPath = new Path(args[1]);
Path outputPath = new Path(args[2]);
MultipleInputs.addInputPath(conf, AInputPath, TextInputFormat.class,
Posts.class);
MultipleInputs.addInputPath(conf, BInputPath, TextInputFormat.class,
User.class);
FileOutputFormat.setOutputPath(conf, outputPath);
conf.setPartitionerClass(KeyPartitioner.class);
conf.setOutputValueGroupingComparator(TextPair.FirstComparator.class);

conf.setMapOutputKeyClass(TextPair.class);
```

21

```
conf.setReducerClass(JoinReducer.class);
conf.setOutputKeyClass(Text.class);

JobClient.runJob(conf);
return 0;
}

public static void main(String[] args) throws Exception {
int exitCode = ToolRunner.run(new JoinDriver(), args);
System.exit(exitCode);
}
}

// JoinReducer.java import java.io.IOException; import java.util.Iterator;
```

```

import org.apache.hadoop.io.Text; import org.apache.hadoop.mapred.*;

public class JoinReducer extends MapReduceBase implements Reducer<TextPair,
Text, Text, Text> {

@Override

public void reduce (TextPair key, Iterator<Text> values, OutputCollector<Text,
Text> output,

Reporter reporter)

throws IOException

{

Text nodeId = new Text(values.next()); while (values.hasNext()) {

Text node = values.next();

Text outValue = new Text(nodeId.toString() + "\t\t" + node.toString());

output.collect(key.getFirst(), outValue);

}

}

}

// User.java import java.io.IOException; import java.util.Iterator; import

org.apache.hadoop.conf.Configuration; import

org.apache.hadoop.fs.FSDataInputStream; import

org.apache.hadoop.fs.FSDataOutputStream; import

org.apache.hadoop.fs.FileSystem; import

org.apache.hadoop.fs.Path; import org.apache.hadoop.io.LongWritable; import

org.apache.hadoop.io.Text; import org.apache.hadoop.mapred.*;

import org.apache.hadoop.io.IntWritable;

public class User extends MapReduceBase implements Mapper<LongWritable,

Text, TextPair, Text> {

```

22

@Override

```
public void map(LongWritable key, Text value, OutputCollector<TextPair, Text>  
output, Reporter
```

```
reporter)
```

```
throws IOException
```

```
{
```

```
String valueString = value.toString();
```

```
String[] SingleNodeData = valueString.split("\t");
```

```
output.collect(new TextPair(SingleNodeData[0], "1"), new  
Text(SingleNodeData[1]));
```

```
}
```

```
}
```

```
//Posts.java import java.io.IOException;
```

```
import org.apache.hadoop.io.*; import org.apache.hadoop.mapred.*;
```

```
public class Posts extends MapReduceBase implements Mapper<LongWritable,  
Text, TextPair, Text> {
```

```
@Override
```

```
public void map(LongWritable key, Text value, OutputCollector<TextPair, Text>  
output, Reporter
```

```
reporter)
```

```
throws IOException
```

```
{
```

```
String valueString = value.toString();
```

```
String[] SingleNodeData = valueString.split("\t"); output.collect(new
```

```

    TextPair(SingleNodeData[3], "0"), new
    Text(SingleNodeData[9]));
}
}

// TextPair.java import java.io.*;
import org.apache.hadoop.io.*;

public class TextPair implements WritableComparable<TextPair> {
    private Text first; private Text second;

    public TextPair() { set(new Text(), new Text());
    }

    public TextPair(String first, String second) { set(new Text(first), new Text(second));
    }

    public TextPair(Text first, Text second) { set(first, second);
    }

23
    public void set(Text first, Text second) { this.first = first; this.second = second;
    }

    public Text getFirst() { return first;
    }

    public Text getSecond() { return second;

```



```
}
```

```
@Override
```

```
public void write(DataOutput out) throws IOException { first.write(out);  
second.write(out);
```

```
}
```

```
@Override public void readFields(DataInput in) throws IOException {  
first.readFields(in);
```

```
second.readFields(in);
```

```
}
```

```
@Override public int hashCode() { return first.hashCode() * 163 +  
second.hashCode();
```

```
}
```

```
@Override public boolean equals(Object o) { if (o instanceof TextPair) { TextPair tp  
= (TextPair) o;
```

```
return first.equals(tp.first) && second.equals(tp.second);
```

```
} return false;
```

```
}
```

```
@Override public String toString() { return first + "\t" + second;
```

```
}
```

```
@Override
```

```
public int compareTo(TextPair tp) { int cmp = first.compareTo(tp.first); if (cmp != 0)  
{ return
```

```
cmp;
```

```

    }
    return second.compareTo(tp.second);
}

// ^^ TextPair

// vv TextPairComparator public static class Comparator extends
WritableComparator {

    private static final Text.Comparator TEXT_COMPARATOR = new Text.Comparator();

    public Comparator() { super(TextPair.class);
    }

    @Override public int compare(byte[] b1, int s1, int l1, byte[] b2, int s2, int l2) {
    try {
24
        int firstL1 = WritableUtils.decodeVIntSize(b1[s1]) + readVInt(b1, s1); int firstL2 =
        WritableUtils.decodeVIntSize(b2[s2]) + readVInt(b2, s2); int cmp =
        TEXT_COMPARATOR.compare(b1, s1, firstL1, b2, s2, firstL2); if (cmp != 0) { return
        cmp;
        }

        return TEXT_COMPARATOR.compare(b1, s1 + firstL1, l1 - firstL1,
        b2, s2 + firstL2, l2 - firstL2);
    } catch (IOException e) { throw new IllegalArgumentException(e);
    }
    }
}

```

```

}
static {
    WritableComparator.define(TextPair.class, new Comparator());
}
public static class FirstComparator extends WritableComparator {

    private static final Text.Comparator TEXT_COMPARATOR = new Text.Comparator();

    public FirstComparator() { super(TextPair.class);
    }

    @Override public int compare(byte[] b1, int s1, int l1, byte[] b2, int s2, int l2) {
        try {
            int firstL1 = WritableUtils.decodeVIntSize(b1[s1]) + readVInt(b1, s1); int firstL2 =
            WritableUtils.decodeVIntSize(b2[s2]) + readVInt(b2, s2); return
            TEXT_COMPARATOR.compare(b1,
            s1, firstL1, b2, s2, firstL2);
        } catch (IOException e) { throw new IllegalArgumentException(e);
        }
    }

    @Override
    public int compare(WritableComparable a, WritableComparable b) { if (a
    instanceof TextPair && b
    instanceof TextPair) { return ((TextPair) a).first.compareTo(((TextPair) b).first);
    }
}

```

```
return super.compare(a, b);
```

```
}
```

```
}
```

```
}
```

```
c:\hadoop_new\share\hadoop\mapreduce>hdfs dfs -cat \joinOutput\part-00000
"100005361"      "2"      "36134"
"100018705"      "2"      "76"
"100022094"      "0"      "6354"
```

LAB 9 Program to print word count on scala shell and print “Hello world” on scala IDE

```
scala> println("Hello World!");
```

Hello World!

```
val data=sc.textFile("sparkdata.txt")
```

```
data.collect;
```

```
val splitdata = data.flatMap(line => line.split(" "));
```

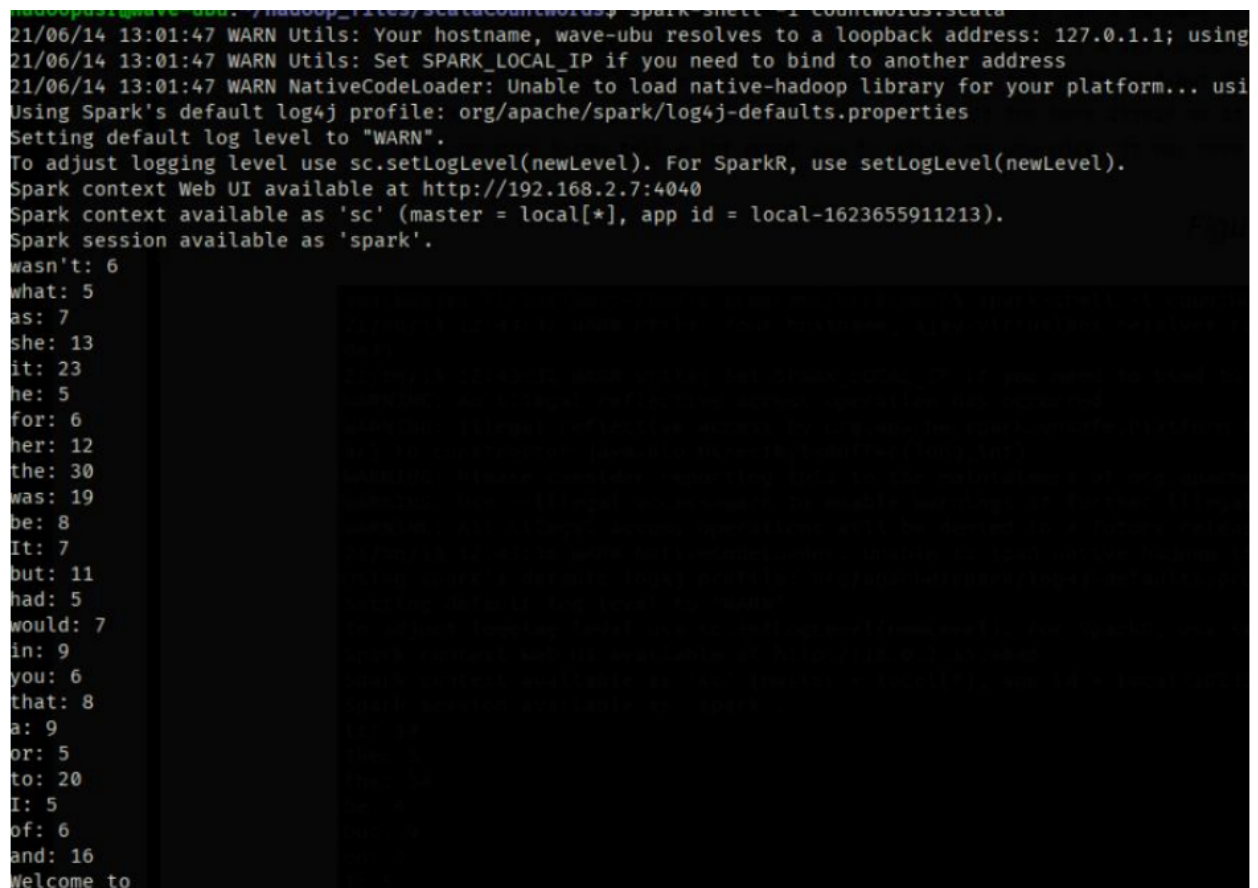
```
splitdata.collect;
```

```
val mapdata = splitdata.map(word => (word,1));
```

```
mapdata.collect;
```

```
val reducedata = mapdata.reduceByKey(_+_);
```

```
reducedata.collect;
```



The screenshot shows a Scala shell terminal window with the following output:

```
21/06/14 13:01:47 WARN Utils: Your hostname, wave-ubu resolves to a loopback address: 127.0.1.1; using
21/06/14 13:01:47 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
21/06/14 13:01:47 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... usi
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://192.168.2.7:4040
Spark context available as 'sc' (master = local[*], app id = local-1623655911213).
Spark session available as 'spark'.
wasn't: 6
what: 5
as: 7
she: 13
it: 23
he: 5
for: 6
her: 12
the: 30
was: 19
be: 8
It: 7
but: 11
had: 5
would: 7
in: 9
you: 6
that: 8
a: 9
or: 5
to: 20
I: 5
of: 6
and: 16
Welcome to
```

LAB-10 :-Using RDD and FlatMap count how many times each word appears in a file and write out a list of words whose count is strictly greater than 4 using Spark

- commands and output:

```
cala> val textFile=sc.textFile("/home/hduser/Desktop/sample.txt");
```

```
textFile: org.apache.spark.rdd.RDD[String] = /home/hduser/Desktop/sample.txt  
MapPartitionsRDD[8] at textFile at <console>:24
```

```
scala> val counts=textFile.flatMap(line=>line.split(" "  
"))).map(word=>(word,1)).reduceByKey(_=_)
```

```
<console>:25: error: reassignment to val
```

```
    val counts=textFile.flatMap(line=>line.split(" "  
"))).map(word=>(word,1)).reduceByKey(_=_)
```

^

```
scala> val counts=textFile.flatMap(line=>line.split("
")).map(word=>(word,1)).reduceByKey(_+_)
```

counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[11] at reduceByKey at <console>:25

```
scala> import scala.collection.immutable.ListMap

import scala.collection.immutable.ListMap
```

```
scala> val sorted=ListMap(counts.collect.sortWith(_._2>_._2):_*)
```

sorted: scala.collection.immutable.ListMap[String,Int] = Map(is -> 4, how -> 4, your -> 4, are -> 1, brother -> 1, sister -> 1, family -> 1, ypu -> 1, job -> 1, hi -> 1, hw -> 1)

```
scala> println(sorted)
```

Map(is -> 4, how -> 4, your -> 4, are -> 1, brother -> 1, sister -> 1, family -> 1, ypu -> 1, job -> 1, hi -> 1, hw -> 1)

```
scala> for((k,v)<-sorted)
```

```
  | {
  |   if(v>4)
  |   {
  |     print(k+",")
  |     print(v)
  |     println()
  |   }
```

