

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



## LAB REPORT

on

## BIG DATA ANALYTICS

*Submitted by*

**NIKHIL ABRAHAM (1BM19CS101)**

*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 DATA ANALYTICS**" carried out by **NIKHIL ABRAHAM(1BM19CS101)**, who is bonafide student of **B. M. S. College of Engineering**. It is in partial fulfilment 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.

**Pallavi G.B**  
Assistant Professor  
Department of CSE  
BMSCE, Bengaluru

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

## INDEX

Sl. No.	Experiment Title
1.	Cassandra Lab Program 1: - Student Database
2.	Cassandra Lab Program 2: - Library Database
3.	MongoDB Lab Program 1 (CRUD Demonstration):
4.	HADOOP INSTALLATION
5.	HADOOP SAMPLE
6.	MAPREDUCE TEMPERATURE
7.	MAPREDUCE TOPN
8.	MAPREDUCE JOIN
9.	SCALA INSTALLATION
10.	SCALA WORDCOUNT

## 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 1

1. Create a key space by name Employee

```
cqlsh> create keyspace LAB1_Employee with replication = { 'class':'SimpleStrategy','replication_factor':1};  
cqlsh> use LAB1_Employee;  
cqlsh:lab1_employee> |
```

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:lab1_employee> create table Employee_info(Emp_id int ,Emp_name text ,Designation text  
,Date_of_joining timestamp,Salary double,Dept_name text,primary key(Emp_id));  
cqlsh:lab1_employee> |
```

3. Insert the values into the table in batch

```
cqlsh:lab1_employee> begin batch insert into employee_info(Emp_id,Emp_name,Designation,Date_of_joining,Salary,Dept_name)values(11,'Pankaj','Senior_Developer','2022-05-12',4500000,'Developing') insert into employee_info(Emp_id,Emp_name,Designation,Date_of_joining,Salary,Dept_name)values(12,'Preetham','Manager','2022-05-13',6500000,'Developing') insert into employee_info(Emp_id,Emp_name,Designation,Date_of_joining,Salary,Dept_name)values(13,'Prithvi','CEO','2012-05-13',8500000,'Overall') apply batch;
cqlsh:lab1_employee> select * from employee_info;

emp_id | date_of_joining | dept_name | designation | emp_name | salary
-----+-----------------+-----------+-------------+-----------+-----
 13 | 2012-05-12 18:30:00.000000+0000 | Overall | CEO | Prithvi | 8.5e+06
 11 | 2022-05-11 18:30:00.000000+0000 | Developing | Senior_Developer | Pankaj | 4.5e+06
 12 | 2022-05-12 18:30:00.000000+0000 | Developing | Manager | Preetham | 6.5e+06

(3 rows)
cqlsh:lab1_employee>
```

4. Update Employee name and Department of Emp-Id 121

```
cqlsh:lab1_employee> update employee_info set Emp_name='Puneeth' ,Dept_name='Sales' where Emp_id=13;
cqlsh:lab1_employee> select * from employee_info;

emp_id | date_of_joining | dept_name | designation | emp_name | salary
-----+-----------------+-----------+-------------+-----------+-----
 13 | 2012-05-12 18:30:00.000000+0000 | Sales | CEO | Puneeth | 8.5e+06
 11 | 2022-05-11 18:30:00.000000+0000 | Developing | Senior_Developer | Pankaj | 4.5e+06
 12 | 2022-05-12 18:30:00.000000+0000 | Developing | Manager | Preetham | 6.5e+06

(3 rows)
```

5. Sort the details of Employee records based on salary

```
cqlsh:lab1_employee> begin batch
    ... insert into emp(id,salary,name)values(5,45000,'Pankaj')
    ... insert into emp(id,salary,name)values(7,455000,'Preetham')
    ... insert into emp(id,salary,name)values(9,55000,'ram')
    ... apply batch;
cqlsh:lab1_employee> select * from emp;

id | salary | name
---+-----+---
 5 | 45000 | Pankaj
 7 | 4.55e+05 | Preetham
 9 | 55000 | ram

(3 rows)
cqlsh:lab1_employee> paging off;
Disabled Query paging.
cqlsh:lab1_employee> select * from emp where id in (5,7,9) order by salary;

id | salary | name
---+-----+---
 5 | 45000 | Pankaj
 9 | 55000 | ram
 7 | 4.55e+05 | Preetham

(3 rows)
```

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:lab1_employee> alter table employee_info add projects text;
cqlsh:lab1_employee> select * from employee_info;

emp_id | date_of_joining | dept_name | designation | emp_name | projects | salary
-----+-----------------+-----------+-------------+-----------+-----+-----
 13 | 2012-05-12 18:30:00.000000+0000 | Sales | CEO | Puneeth | null | 8.5e+06
 11 | 2022-05-11 18:30:00.000000+0000 | Developing | Senior_Developer | Pankaj | null | 4.5e+06
 12 | 2022-05-12 18:30:00.000000+0000 | Developing | Manager | Preetham | null | 6.5e+06

(3 rows)
```

7. Update the altered table to add project names.

```
cqlsh:lab1_employee> update Employee_info set projects='Kubernetes' where Emp_id=11;
cqlsh:lab1_employee> update Employee_info set projects='node_js' where Emp_id=12;
cqlsh:lab1_employee> update Employee_info set projects='Mobile_app' where Emp_id=13;
cqlsh:lab1_employee> select * from employee_info;
```

emp_id	date_of_joining	dept_name	designation	emp_name	projects	salary
13	2012-05-12 18:30:00.000000+0000	Sales	CEO	Puneeth	Mobile_app	8.5e+06
11	2022-05-11 18:30:00.000000+0000	Developing	Senior_Developer	Pankaj	Kubernetes	4.5e+06
12	2022-05-12 18:30:00.000000+0000	Developing	Manager	Preetham	node_js	6.5e+06

(3 rows)

8 Create a TTL of 15 seconds to display the values of Employees.

```
cqlsh:lab1_employee> insert into Employee_info (Emp_id,Emp_name,Designation,Date_of_joining,Salary,Dept_name)values(19,'Prithvi','Senior_Developer','2022-08-12',400000,'Developing') using TTL 50;
cqlsh:lab1_employee> select TTL(emp_name) from Employee_info where Emp_id=19;
```

```
ttl(emp_name)
```

```
45
```

## LAB 2

1 Create a key space by name Library

```
cqlsh> create keyspace lab2_library with replication={'class':'SimpleStrategy','replication_factor':1};
cqlsh> use lab2_library;
cqlsh:lab2_library>
```

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:lab2_library> create table library_info(stud_id int,counter_value counter,stud_name text,book_id int
,date_of_issue timestamp,primary key(stud_id,stud_name,book_id,date_of_issue));
cqlsh:lab2_library> A
```

3. Insert the values into the table in batch

```
cqlsh:lab2_library> update library_info set counter_value=counter_value + 2 where stud_id=2 and stud_name='Pankaj' and book_id=145 and date_of_issue='2022-08-04';
cqlsh:lab2_library> select * from library_info;
```

stud_id	stud_name	book_id	date_of_issue	counter_value
2	Pankaj	145	2022-08-03 18:30:00.000000+0000	4

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

```
cqlsh:lab2_library> update library_info set counter_value=counter_value + 2 where stud_id=2 and stud_name='Pankaj' and book_id=145 and date_of_issue='2022-08-04';
cqlsh:lab2_library> select * from library_info;
```

stud_id	stud_name	book_id	date_of_issue	counter_value
2	Pankaj	145	2022-08-03 18:30:00.000000+0000	2

5. Write a query to show that a student with id 112 has taken a book “BDA” 2 times.

```
cqlsh:lab2_library> update library_info set counter_value=counter_value + 2 where stud_id=112 and stud_name='Preetham' and book_id=145 and date_of_issue='2022-08-04';
cqlsh:lab2_library> select counter_value from library_info where stud_id=112;
```

counter_value
2

5. Write a query to show that a student with id 112 has taken a book "BDA" 2 times.

```
cqlsh:lab2_library> update library_info set counter_value=counter_value + 2 where stud_id=112 and stud_name='Preetham' and book_id=145 and date_of_issue='2022-08-04';
cqlsh:lab2_library> select counter_value from library_info where stud_id=112;

  counter_value
  -----
    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_value].
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:lab2_library> create table library_info2(stud_id int,counter_value counter,stud_name text,book_id int,date_of_issue timestamp,primary key(stud_id,stud_name,book_id,date_of_issue));
cqlsh:lab2_library> copy library_info2(stud_id,stud_name,book_id,date_of_issue,counter_value)from 'lib.csv';
Using 7 child processes

Starting copy of lab2_library.library_info2 with columns [stud_id, stud_name, book_id, date_of_issue, counter_value].
Processed: 2 rows; Rate:      4 rows/s; Avg. rate:      6 rows/s
2 rows imported from 1 files in 0.356 seconds (0 skipped). ■
cqlsh:lab2_library> select * from library_info;

  stud_id | stud_name | book_id | date_of_issue           | counter_value
  -----+-----+-----+-----+-----+
    2   | Pankaj   | 145   | 2022-08-03 18:30:00.000000+0000 |        4
  112  | Preetham | 145   | 2022-08-03 18:30:00.000000+0000 |        2

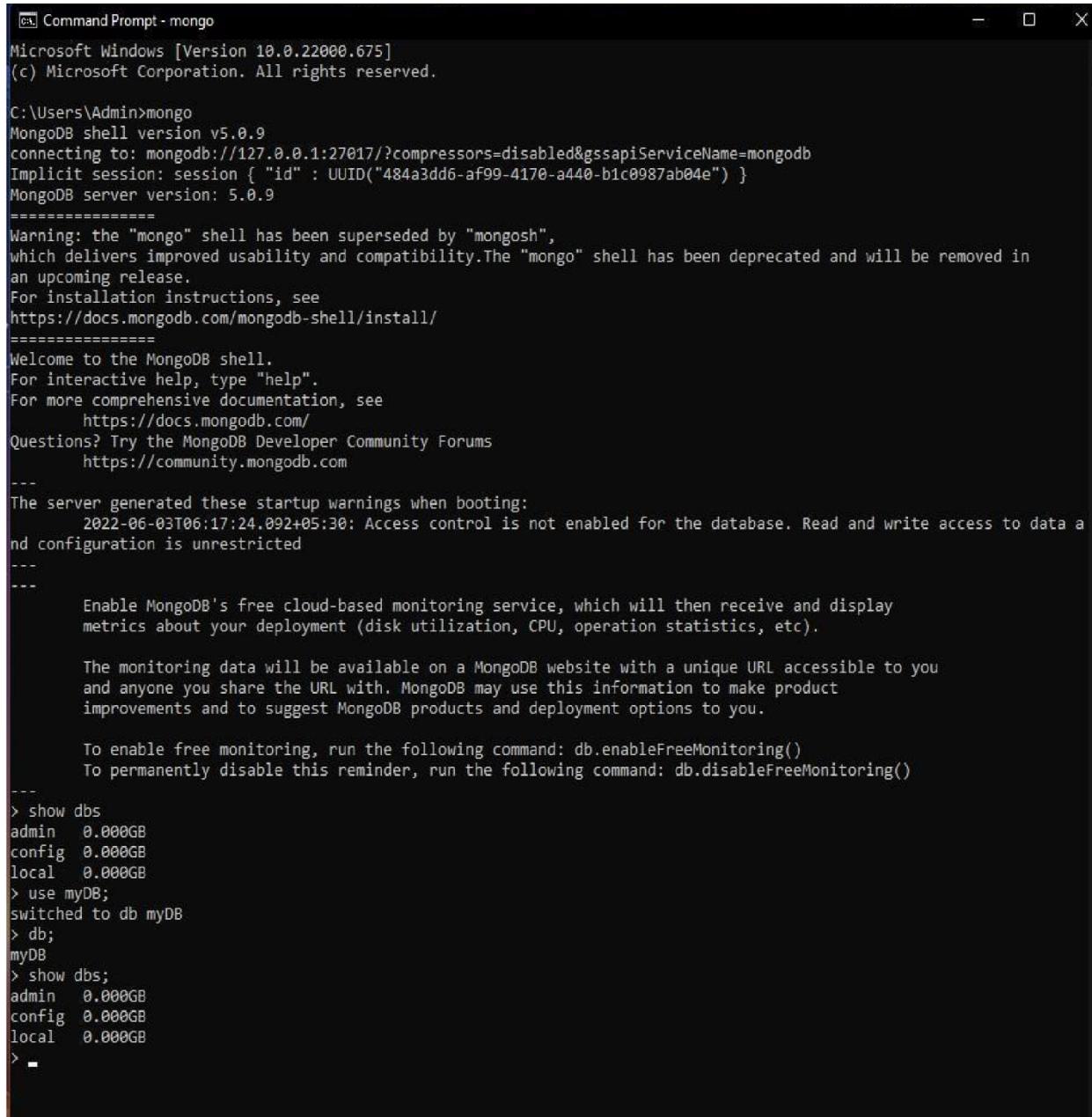
(2 rows)
cqlsh:lab2_library> select * from library_info2;

  stud_id | stud_name | book_id | date_of_issue           | counter_value
  -----+-----+-----+-----+-----+
    2   | Pankaj   | 145   | 2022-08-03 18:30:00.000000+0000 |        4
  112  | Preetham | 145   | 2022-08-03 18:30:00.000000+0000 |        2
```



## LAB 3

- I. CREATE DATABASE IN MONGODB. use myDB; db; (Confirm the existence of your database)  
show dbs; (To list all databases)



```
Command Prompt - mongo
Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>mongo
MongoDB shell version v5.0.9
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("484a3dd6-af99-4170-a440-b1c0987ab04e") }
MongoDB server version: 5.0.9
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
    https://docs.mongodb.com/
Questions? Try the MongoDB Developer Community Forums
    https://community.mongodb.com
---
The server generated these startup warnings when booting:
    2022-06-03T06:17:24.092+05:30: Access control is not enabled for the database. Read and write access to data a
nd configuration is unrestricted
---
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> show dbs
admin 0.000GB
config 0.000GB
local 0.000GB
> use myDB;
switched to db myDB
> db;
myDB
> show dbs;
admin 0.000GB
config 0.000GB
local 0.000GB
> -
```

## II.CRUD (CREATE, READ, UPDATE, DELETE) OPERATIONS

1. To create a collection by the name “Student”. Let us take a look at the collection list prior to the creation of the new collection “Student”.

```
db.createCollection("Student"); > sql equivalent CREATE TABLE
STUDENT(...);
```

2. To drop a collection by the name “Student”.

```
db.Student.drop();
```

3. Create a collection by the name “Students” and store the following data in it.

```
db.Student.insert({_id:1,StudName:"MichelleJacintha",Grade:"VII"&quot;,Hobbies:&quot;InternetSurfing&quot;});
```

4. Insert the document for “AryanDavid” 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 from “Skating” to “Chess”. ) 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:3,StudName:"AryanDavid",Grade:"VII"},{$set:{Hobbies:"Skating"}},{upsert:true});
```

```
> db.createCollection("Student");
{ "ok" : 1 }
> db.Student.drop();
true
> db.createCollection("Student");
{ "ok" : 1 }
> db.Student.insert({_id:1, StudName:"MichelleJacintha", Grade:"VII", Hobbies:"InternetSurfing"});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:1, StudName:"MichelleJacintha", Grade:"VII", Hobbies:"InternetSurfing"});
WriteResult({
    "nInserted" : 0,
    "writeError" : {
        "code" : 11000,
        "errmsg" : "E11000 duplicate key error collection: myDB.Student index: _id_ dup key: { _id: 1.0 }"
    }
})
> db.Student.updateElseInsert({_id:3, StudName:"AryanDavid", Grade:"VII"},{$set:{Hobbies:"Skating"}},{upsert:true});
uncaught exception: TypeError: db.Student.updateElseInsert is not a function :
@(shell):1:1
> db.Student.update({_id:3, StudName:"AryanDavid", Grade:"VII"},{$set:{Hobbies:"Skating"}},{upsert:true});
WriteResult({ "nMatched" : 0, "nUpserted" : 1, "nModified" : 0, "_id" : 3 })
>
```

## 5. FIND METHOD

A. To search for documents from the “Students” collection based on certain search criteria.

```
db.Student.find({StudName:"Aryan David"});
({cond..},{columns.. column:1, columnname:0})
```

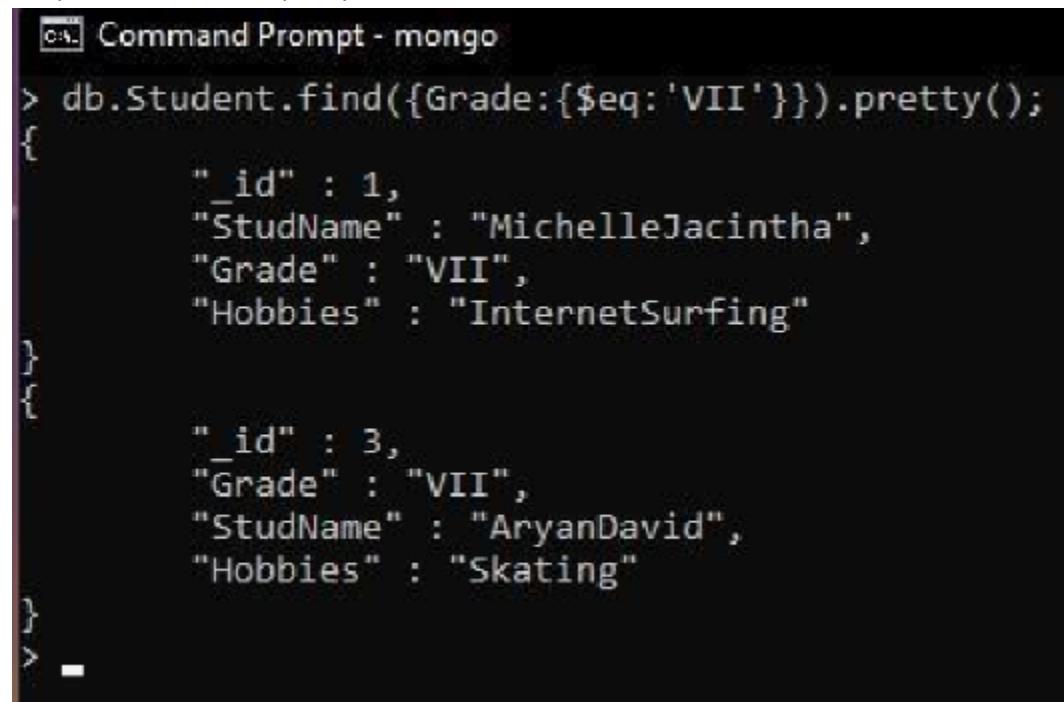
```
> db.Student.find({StudName:"AryanDavid"});
{ "_id" : 3, "Grade" : "VII", "StudName" : "AryanDavid", "Hobbies" : "Skating" }
```

B. 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({}, {StudName:1,Grade:1,_id:0});
```

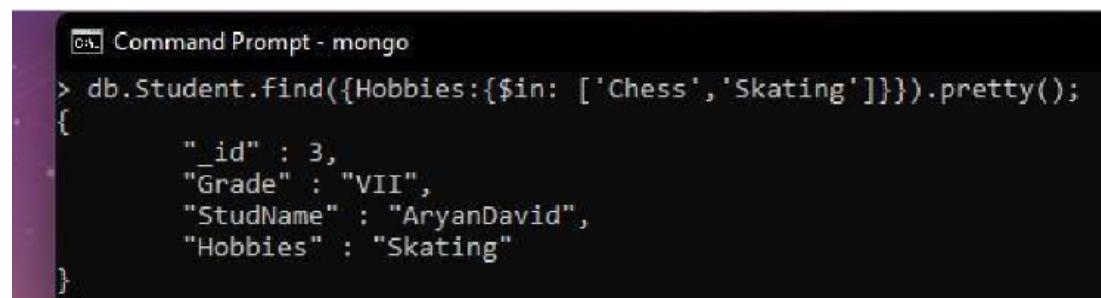
```
Command Prompt - mongo
> db.Student.find({}, {StudName:1,Grade:1,_id:0});
{ "StudName" : "MichelleJacintha", "Grade" : "VII" }
{ "Grade" : "VII", "StudName" : "AryanDavid" }
>
```

C. To find those documents where the Grade is set to ‘VII’ db.Student.find({Grade: {\$eq:'VII'}}).pretty();



```
> db.Student.find({Grade: {$eq: 'VII'}}).pretty();
{
    "_id" : 1,
    "StudName" : "MichelleJacintha",
    "Grade" : "VII",
    "Hobbies" : "InternetSurfing"
}
{
    "_id" : 3,
    "Grade" : "VII",
    "StudName" : "AryanDavid",
    "Hobbies" : "Skating"
}
>
```

D. 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:[‘Chess’,‘Skating’]} }).pretty () ;



```
> db.Student.find({Hobbies:{ $in: ['Chess', 'Skating']}}).pretty();
{
    "_id" : 3,
    "Grade" : "VII",
    "StudName" : "AryanDavid",
    "Hobbies" : "Skating"
}
```

E. To find documents from the Students collection where the StudName begins with “M”.  
db.Student.find( {StudName:/^M/} ).pretty();



```
> db.Student.find({StudName:/^M/}).pretty();
{
    "_id" : 1,
    "StudName" : "MichelleJacintha",
    "Grade" : "VII",
    "Hobbies" : "InternetSurfing"
}
>
```

F. To find documents from the Students collection where the StudName has an “e” in any position.  
db.Student.find({StudName:/e/}).pretty();

```
Command Prompt - mongo
> db.Student.find({StudName:/e/}).pretty();
{
    "_id" : 1,
    "StudName" : "MichelleJacintha",
    "Grade" : "VII",
    "Hobbies" : "InternetSurfing"
}
>
```

G. To find the number of documents in the Students collection. db.Student.count();

```
Command Prompt - mongo
> db.Student.count();
2
>
```

H. To sort the documents from the Students collection in the descending order of StudName.  
db.Student.find().sort({StudName:-1}).pretty();

```
Command Prompt - mongo
> db.Student.find().sort({StudName:-1}).pretty();
{
    "_id" : 1,
    "StudName" : "MichelleJacintha",
    "Grade" : "VII",
    "Hobbies" : "InternetSurfing"
}
{
    "_id" : 3,
    "Grade" : "VII",
    "StudName" : "AryanDavid",
    "Hobbies" : "Skating"
}
>
```

III. Import data from a CSV file  
Given a CSV file “sample.txt” in the D:drive, import the file into the MongoDB collection, “SampleJSON”. The collection is in the database “test”.  
mongoimport --db Student --collection airlines --type csv --headerline --file /home/hduser/Desktop/airline.csv

```
C:\Program Files\MongoDB\Server\5.0\bin>mongoimport --db Student --collection airlines --type csv --file "C:\Program Files\MongoDB\airline.csv" --headerline
2022-06-03T08:24:18.366+0530    connected to: mongodb://localhost/
2022-06-03T08:24:18.395+0530    6 document(s) imported successfully. 0 document(s) failed to import.

C:\Program Files\MongoDB\Server\5.0\bin>
```

#### IV. Export data to a CSV file

This command used at the command prompt exports MongoDB JSON documents from

“Customers” collection in the “test” database into a CSV file “Output.txt” in the D:drive.

```
mongoexport --host localhost --db Student --collection airlines --csv --out /home/hduser/Desktop/output.txt --fields "Year","Quarter"
```

```
C:\Program Files\MongoDB\Server\5.0\bin>mongoexport --host localhost --db Student --collection airlines --csv --out "C:\home\hduser\Desktop\output.txt" --fields "Year","Quarter"
2022-06-03T08:28:58.325+0530    csv flag is deprecated; please use --type=csv instead
2022-06-03T08:28:58.946+0530    connected to: mongodb://localhost/
2022-06-03T08:28:58.972+0530    exported 6 records

C:\Program Files\MongoDB\Server\5.0\bin>_
```

#### V. Save Method :

Save() method will insert a new document, if the document with the \_id does not exist. If it exists it will replace the existing document.

```
db.Students.save({StudName:"Vamsi", Grade:"VI"})
```

```
switched to db Student
> db.Students.save({StudName:"Vamsi", Grade:"VI"})
WriteResult({ "nInserted" : 1 })
> _
```

#### VI. Add a new field to existing Document:

```
db.Students.update({_id:4},{$set:{Location:"Network"}})
```

```
> db.Students.update({_id:4},{$set:{Location:"Network"}})
WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })
> _
```

#### VII.. Count the number of documents in Student Collections with grade :VII

```
db.Students.count({Grade:"VII"}) retrieve first 3 documents
```

```
db.Students.find({Grade:"VII"}).limit(3).pretty(); Sort the
```

document in Ascending order

```
db.Students.find().sort({StudName:1}).pretty(); Note: for
```

desending order : db.Students.find().sort({StudName:-

```
1}).pretty(); to Skip the 1 st two documents from the
```

```
Students Collections db.Students.find().skip(2).pretty()
```

```
> db.Students.find().sort({StudName:1}).pretty();
{
  "_id" : ObjectId("629979944de3211e43081306"),
  "StudName" : "Vamsi",
  "Grade" : "VII"
}>
```

## LAB 4

## SCREENSHOT OF HADOOP INSTALLATION

LAB 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 \tempc:  
\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.txthello 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 \tempFound 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/
```

tempc:\hadoop\_new\sbin>hdfs dfs -mv \lab1 \temp

```
c:\hadoop_new\sbin>hdfs dfs -ls  
\tempFound 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 21:12 /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.txtDeleted /temp/sample.txt  
c:\hadoop_new\sbin>hdfs dfs -ls  
\tempFound 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 \tempFound 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
```

SCREENSHOTS –

```

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 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

```

## LAB 6

Map reduce Temperature

For the given file, Create a Map Reduce program to

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

```

// AverageDriver.java
package temperature;
import org.apache.hadoop.io.*;
import org.apache.hadoop.fs.*;
import org.apache.hadoop.mapreduce.*;

```

```
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("Please Enter the input and output parameters");
            System.exit(-1);
        }

        Job job = new Job(); job.setJarByClass(AverageDriver.class);
        job.setJobName("Max temperature");
        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.java
package temperature;

import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import java.io.IOException;

public class AverageMapper extends Mapper <LongWritable, Text, Text, IntWritable>
{
    public static final int MISSING = 9999;

    public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException
    {
        String line = value.toString(); String
        year = line.substring(15,19);int
        temperature;
        if (line.charAt(87)=='+')
            temperature = Integer.parseInt(line.substring(88, 92));
        else
            temperature = Integer.parseInt(line.substring(87, 92));
        String quality = line.substring(92, 93);
        if(temperature != MISSING && quality.matches("[01459]"))
            context.write(new Text(year),new IntWritable(temperature));
    }
}

//AverageReducer.java
package temperature;

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

```

```

org.apache.hadoop.mapreduce.*; import
java.io.IOException;

public class AverageReducer extends Reducer <Text, IntWritable, Text, IntWritable>
{
    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws
IOException,InterruptedException
    {
        int max_temp = 0;int count = 0;
        for (IntWritable value : values)
        {
            max_temp += value.get();count+=1;
        }
        context.write(key, new IntWritable(max_temp/count));
    }
}

```

SCREENSHOT -

```

c:\hadoop_new\sbin>hdfs dfs -cat /tempAverageOutput/part-r-00000
1901    46
1949    94
1950    3

```

b) Find the mean max temperature for every month.

```

//TempDriver.java

package temperatureMax;

import
org.apache.hadoop.io.*;import
org.apache.hadoop.fs.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class TempDriver
{

```

```

public static void main (String[] args) throws Exception
{
    if (args.length != 2)
    {
        System.err.println("Please Enter the input and output
                           parameters");System.exit(-1);
    }

    Job job = new Job();
    job.setJarByClass(TempDriver.class);
    job.setJobName("Max temperature");
    FileInputFormat.addInputPath(job,new Path(args[0]));
    FileOutputFormat.setOutputPath(job,new Path (args[1]));

    job.setMapperClass(TempMapper.class);
    job.setReducerClass(TempReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    System.exit(job.waitForCompletion(true)?0:1);

}

//TempMapper.java
package temperatureMax;

import org.apache.hadoop.io.*;
import
org.apache.hadoop.mapreduce.*;import
java.io.IOException;

public class TempMapper extends Mapper <LongWritable, Text, Text, IntWritable>
{
    public static final int MISSING = 9999;
}

```

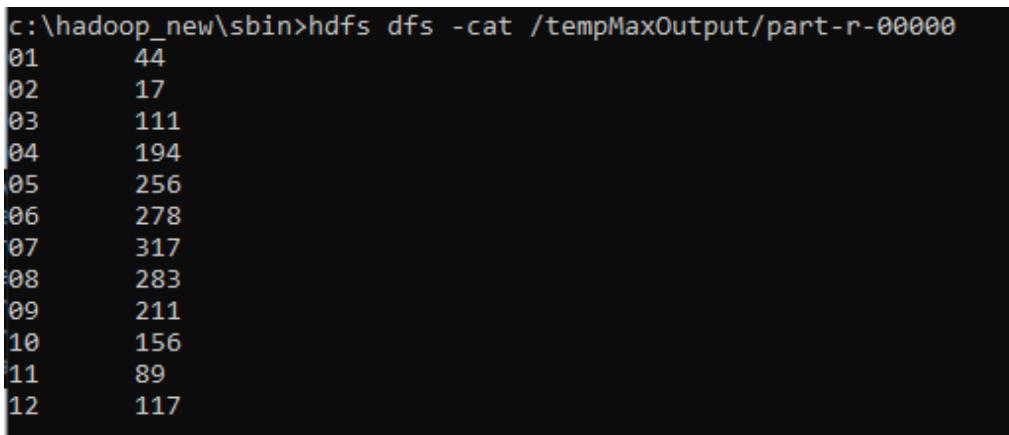
```

public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException
{
    String line = value.toString();
    String month =
    line.substring(19,21);int
    temperature;
    if (line.charAt(87)=='+')
        temperature = Integer.parseInt(line.substring(88, 92));
    else
        temperature = Integer.parseInt(line.substring(87, 92));
    String quality = line.substring(92, 93);
    if(temperature != MISSING && quality.matches("[01459]"))
        context.write(new Text(month),new IntWritable(temperature));

}
}

```

#### SCREENSHOT -



```

c:\hadoop_new\sbin>hdfs dfs -cat /tempMaxOutput/part-r-00000
01      44
02      17
03     111
04     194
05     256
06     278
07     317
08     283
09     211
10     156
11      89
12     117

```

## LAB 7

### Map reduce TOPN

For a given Text file, create a Map Reduce program to sort the content in an alphabetic orderlisting 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
    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
        wordint 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> {

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

        // computes the number of occurrences of a single
        wordint 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;
}

```

#### SCREENSHOTS –

```
C:\hadoop_new\share\hadoop\mapreduce>hdfs dfs -cat \sortwords\input.txt
deer bear river
car car river
deer car bear
car deer deer
car car car
bear deer deer
```

```
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 andScore.

```
// 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,
Posts.class)
TextInputFormat.class, MultipleInputs.addInputPath(conf, BInputPath,
;User.class);
TextInputFormat.class,


FileOutputFormat.setOutputPath(conf, outputPath);

conf.setPartitionerClass(KeyPartitioner.class);
conf.setOutputValueGroupingComparator(TextPair.FirstComparator.class);

conf.setMapOutputKeyClass(TextPair.class);

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> {

    @Override
    public void map(LongWritable key, Text value, OutputCollector<TextPair, Text> output,
Reportер reporter)
        throws IOException
    {

        String valueString = value.toString();

```

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

```

        output.collect(new TextPair(SingleNodeData[0], "1"),
newText(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);
    }

    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 {
            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, l1 - firstL1, b2, s2 + firstL2, l2 - 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);  
}  
}  
}  
}
```

## SCREENSHOTS –

```
c:\hadoop_new\share\hadoop\mapreduce>hdfs dfs -cat \posts\sampleposts.tsv
"2312" "Feedback on Audio Quality" "cs101 production audio" "100005361" "<p>We are looking for feedback on the audio in our videos. Tell us what you think and try to be as <em>specific</em> as possible.</p>" "question"
"\N" "\N" "2012-02-23 00:28:02.321344+00" "2" "" "\N" "201398145" "2014-01-14 17:18:35.613939+00" "2960" "\N" "\N" "524" "f"
"2014856" "" "cs101" "100022094" "<p>I also would like to know the answer to this question. An 'open exam' sounds great, but on the other hand it also seems pretty easy to cheat now: solutions have been posted and anybody only interested in a certificate wouldn't have much of a problem getting the highest distinction. So where is the catch??</p>" "answer" "2014706" "2014706" "2012-07-01 10:32:36.302782+00" "0" "" "\N"
"100022094" "2012-07-01 10:32:36.302782+00" "2020501" "\N" "\N" "0" "f"
"2004004" "" "cs101" "100018705" "<p>But then why even the new variable q? Why not just modify the variable p?</p>" "comment" "2003997"
"2003993" "2012-05-03 21:07:52.028935+00" "2" ""
"\N" "100018705" "2012-05-03 21:07:52.028935+00" "2005150" "\N" "\N" "0" "f"

c:\hadoop_new\share\hadoop\mapreduce>hdfs dfs -cat \users\sampleusers.tsv
"100006402" "18" "0" "0" "0"
"100022094" "6354" "4" "12" "50"
"100018705" "76" "0" "3" "4"
"100005361" "36134" "73" "220" "333"
```

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

## LAB 9:

Scala installation screenshot

```
sam@ubuntu:~$ start-master.sh
starting org.apache.spark.deploy.master.Master, logging to /opt/spark/logs/spark-sam-org.apache.spark.deploy.master.Master-1-ubuntu.out
sam@ubuntu:~$ start-slave.sh spark://ubuntu:7077
This script is deprecated, use start-worker.sh
starting org.apache.spark.deploy.worker.Worker, logging to /opt/spark/logs/spark-sam-org.apache.spark.deploy.worker.Worker-1-ubuntu.out
```

```
sam@ubuntu:~$ spark-shell
21/06/13 07:19:08 WARN Utils: Your hostname, ubuntu resolves to a loopback address: 127.0.1.1; using 192.168.18.128 instead (on interface ens33)
21/06/13 07:19:08 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/opt/spark/jars/spark-unsafe_2.12-3.1.1.jar) to constructor java.nio.DirectByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
21/06/13 07:19:10 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
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://192.168.18.128:4040
Spark context available as 'sc' (master = local[*], app id = local-1623593969342).
Spark session available as 'spark'.
Welcome to

   __|  __ \
  .  \_| \_ \
  /  _ \| /  _ \
 /  \_ \| /  \_ \
   \_ \| \_ \_ \
    \_ \| \_ \_ \
     \_ \| \_ \_ \
      \_ \| \_ \_ \
       \_ \| \_ \_ \
        \_ \| \_ \_ \
         \_ \| \_ \_ \
          \_ \| \_ \_ \
           \_ \| \_ \_ \
            \_ \| \_ \_ \
             \_ \| \_ \_ \
              \_ \| \_ \_ \
               \_ \| \_ \_ \
                \_ \| \_ \_ \
                 \_ \| \_ \_ \
                  \_ \| \_ \_ \
                   \_ \| \_ \_ \
                    \_ \| \_ \_ \
                     \_ \| \_ \_ \
                      \_ \| \_ \_ \
                       \_ \| \_ \_ \
                        \_ \| \_ \_ \
                         \_ \| \_ \_ \
                          \_ \| \_ \_ \
                           \_ \| \_ \_ \
                            \_ \| \_ \_ \
                             \_ \| \_ \_ \
                              \_ \| \_ \_ \
                               \_ \| \_ \_ \
                                \_ \| \_ \_ \
                                 \_ \| \_ \_ \
                                  \_ \| \_ \_ \
                                   \_ \| \_ \_ \
                                    \_ \| \_ \_ \
                                     \_ \| \_ \_ \
                                      \_ \| \_ \_ \
                                       \_ \| \_ \_ \
                                        \_ \| \_ \_ \
                                         \_ \| \_ \_ \
                                          \_ \| \_ \_ \
                                           \_ \| \_ \_ \
                                            \_ \| \_ \_ \
                                             \_ \| \_ \_ \
                                              \_ \| \_ \_ \
                                               \_ \| \_ \_ \
                                                \_ \| \_ \_ \
                                                 \_ \| \_ \_ \
                                                  \_ \| \_ \_ \
                                                   \_ \| \_ \_ \
                                                    \_ \| \_ \_ \
                                                     \_ \| \_ \_ \
                                                      \_ \| \_ \_ \
                                                       \_ \| \_ \_ \
                                                        \_ \| \_ \_ \
                                                         \_ \| \_ \_ \
                                                          \_ \| \_ \_ \
                                                           \_ \| \_ \_ \
                                                            \_ \| \_ \_ \
                                                             \_ \| \_ \_ \
                                                              \_ \| \_ \_ \
                                                               \_ \| \_ \_ \
                                                                \_ \| \_ \_ \
                                                                 \_ \| \_ \_ \
                                                                  \_ \| \_ \_ \
                                                                   \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
                                                                    \_ \| \_ \_ \
................................................................
```

// scala shell

```
scala> val textfile = sc.textFile("/home/sam/Desktop/abc.txt")
```

```
textfile: org.apache.spark.rdd.RDD[String] = /home/sam/Desktop/abc.txt MapPartitionsRDD[1] at textFile at <console>:24
```

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

```
counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at
<console>:25
```

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

```
scala> val sorted = ListMap(counts.collect.sortWith(_.2>.2):*)
```

```
scala> println(sorted)
ListMap(car -> 7, deer -> 5, bear -> 3, river -> 3, -> 1)
```

```
scala> for((k,v)<-sorted)
| {
| if(v>4)
| {
|   println(k+"-"+v)
| }
| } car-7 deer-5
```

```
scala> val textfile = sc.textFile("/home/sam/Desktop/abc.txt")
textfile: org.apache.spark.rdd.RDD[String] = /home/sam/Desktop/abc.txt MapPartitionsRDD[8] at textFile at <console>:25

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>:26

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] = ListMap(hello -> 3, apple -> 2, unicorn -> 1, world -> 1)

scala> println(sorted)
ListMap(hello -> 3, apple -> 2, unicorn -> 1, world -> 1)
```

```
scala> for((k,v)<-sorted)
| {
| if(v>2)
| {
|   println(k+"-"+v)
| }
| }
hello-3
```