VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

BIG DATA ANALYTICS LAB

Submitted by

SAI PRANAV (1BM20CS138)

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-2023 to July-2023

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 LAB” carried out by **SAI PRANAV (1BM20CS138),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2023. The Lab report has been approved as it satisfies the academic requirements in respect of a **SAI PRANAV - (20CS6PEBDA)** work prescribed for the said degree.

Rajeshwari Madli
Assistant Professor
Department of CSE
BMSCE, Bengaluru

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

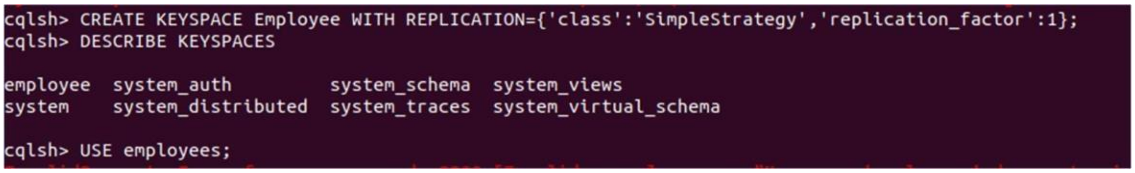
`

Index Sheet

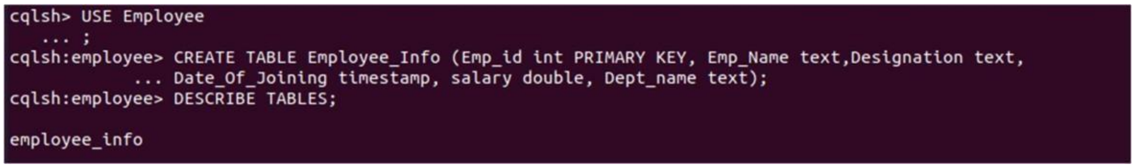
Course Outcome

CO1	Apply the concept of NoSQL, Hadoop or Spark for a given task
CO2	Apply the concept of NoSQL, Hadoop or Spark for a given task
CO3	Apply the concept of NoSQL, Hadoop or Spark for a given task

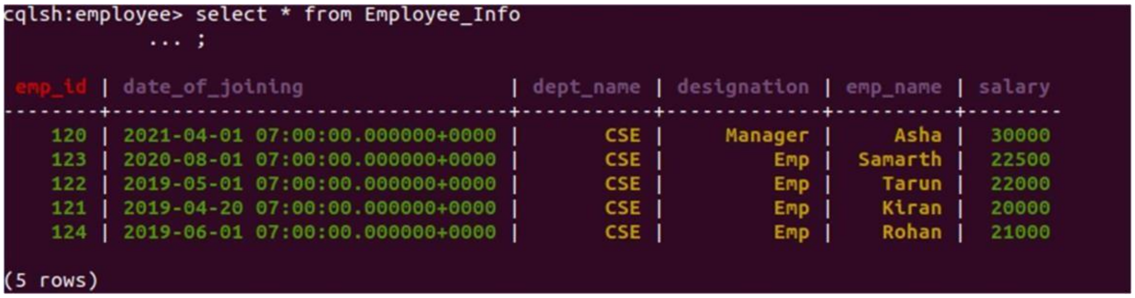
Program 1: Create a Data set either structured/Semi-Structured/Unstructured from twitter/Facebook etc. to perform various DB operations using Cassandra.



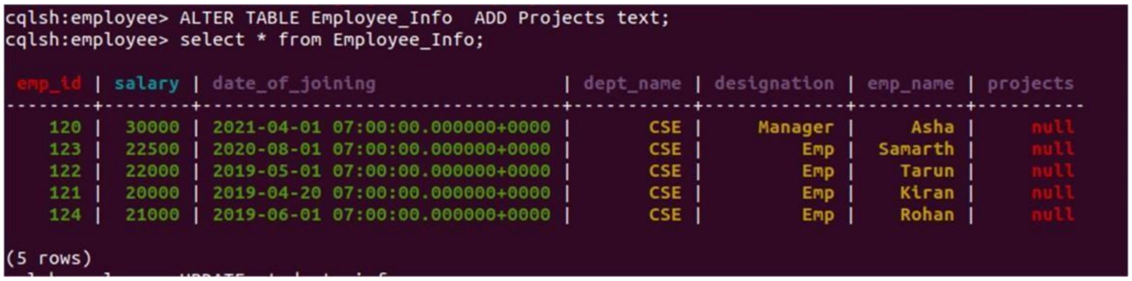
```
cqlsh> CREATE KEYSPACE Employee WITH REPLICATION={'class':'SimpleStrategy','replication_factor':1};
cqlsh> DESCRIBE KEYSPACES

employee  system_auth         system_schema  system_views
system    system_distributed  system_traces  system_virtual_schema

cqlsh> USE employees;
```



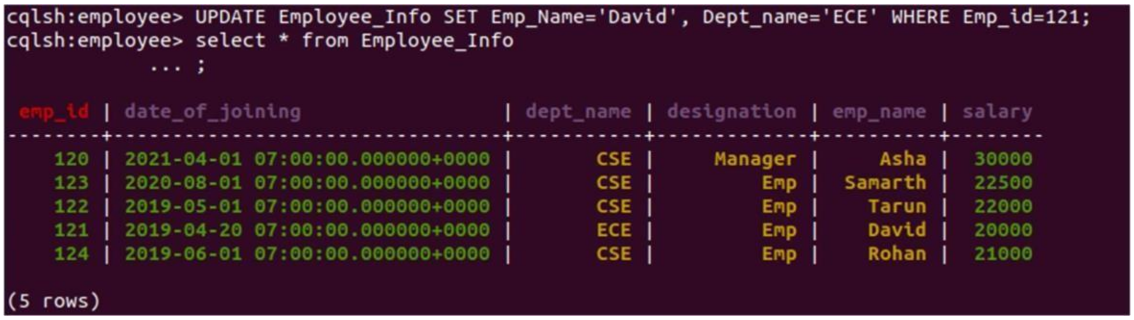
```
cqlsh> USE Employee
   ... ;
cqlsh:employee> CREATE TABLE Employee_Info (Emp_id int PRIMARY KEY, Emp_Name text,Designation text,
            ... Date_Of_Joining timestamp, salary double, Dept_name text);
cqlsh:employee> DESCRIBE TABLES;

employee_info
```



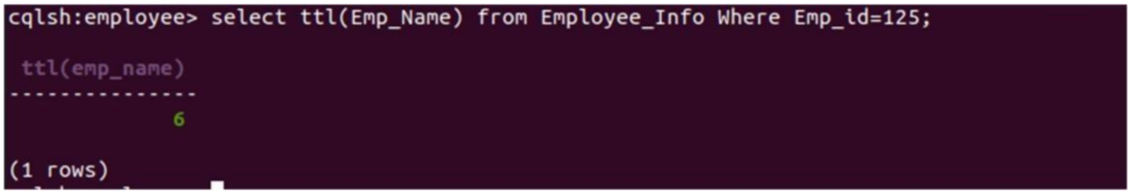
```
cqlsh:employee> select * from Employee_Info
            ... ;

 emp_id | date_of_joining                 | dept_name | designation | emp_name | salary
--------+---------------------------------+-----------+-------------+----------+--------
    120 | 2021-04-01 07:00:00.000000+0000 |       CSE |     Manager |     Asha |  30000
    123 | 2020-08-01 07:00:00.000000+0000 |       CSE |         Emp |  Samarth |  22500
    122 | 2019-05-01 07:00:00.000000+0000 |       CSE |         Emp |    Tarun |  22000
    121 | 2019-04-20 07:00:00.000000+0000 |       CSE |         Emp |    Kiran |  20000
    124 | 2019-06-01 07:00:00.000000+0000 |       CSE |         Emp |    Rohan |  21000

(5 rows)
```



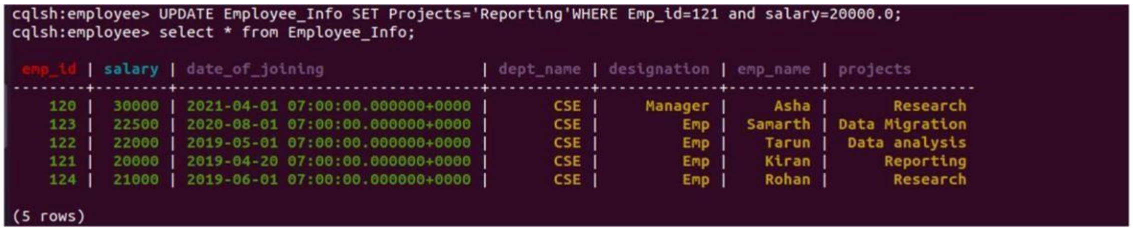
```
cqlsh:employee> ALTER TABLE Employee_Info  ADD Projects text;
cqlsh:employee> select * from Employee_Info;

 emp_id | salary | date_of_joining                 | dept_name | designation | emp_name | projects
--------+--------+---------------------------------+-----------+-------------+----------+----------
    120 |  30000 | 2021-04-01 07:00:00.000000+0000 |       CSE |     Manager |     Asha |     null
    123 |  22500 | 2020-08-01 07:00:00.000000+0000 |       CSE |         Emp |  Samarth |     null
    122 |  22000 | 2019-05-01 07:00:00.000000+0000 |       CSE |         Emp |    Tarun |     null
    121 |  20000 | 2019-04-20 07:00:00.000000+0000 |       CSE |         Emp |    Kiran |     null
    124 |  21000 | 2019-06-01 07:00:00.000000+0000 |       CSE |         Emp |    Rohan |     null

(5 rows)
```



```
cqlsh:employee> UPDATE Employee_Info SET Emp_Name='David', Dept_name='ECE' WHERE Emp_id=121;
cqlsh:employee> select * from Employee_Info
            ... ;

 emp_id | date_of_joining                 | dept_name | designation | emp_name | salary
--------+---------------------------------+-----------+-------------+----------+--------
    120 | 2021-04-01 07:00:00.000000+0000 |       CSE |     Manager |     Asha |  30000
    123 | 2020-08-01 07:00:00.000000+0000 |       CSE |         Emp |  Samarth |  22500
    122 | 2019-05-01 07:00:00.000000+0000 |       CSE |         Emp |    Tarun |  22000
    121 | 2019-04-20 07:00:00.000000+0000 |       ECE |         Emp |    David |  20000
    124 | 2019-06-01 07:00:00.000000+0000 |       CSE |         Emp |    Rohan |  21000

(5 rows)
```

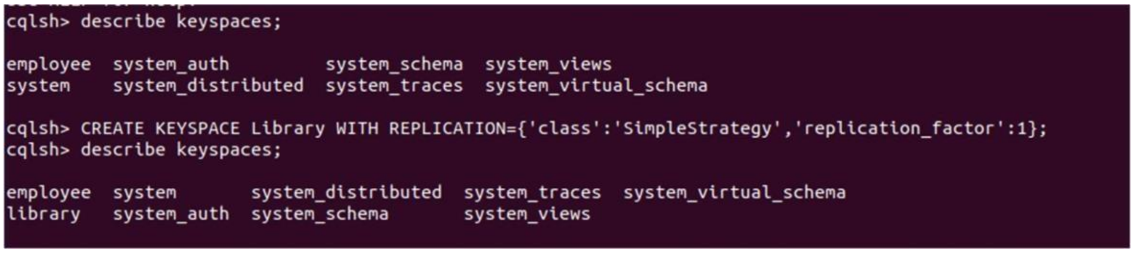


```
cqlsh:employee> select ttl(Emp_Name) from Employee_Info Where Emp_id=125;

 ttl(emp_name)
---------------
             6

(1 rows)
```

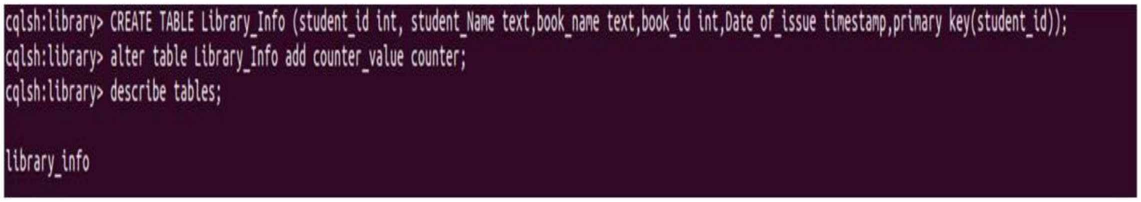


```
cqlsh:employee> UPDATE Employee_Info SET Projects='Reporting'WHERE Emp_id=121 and salary=20000.0;
cqlsh:employee> select * from Employee_Info;

 emp_id | salary | date_of_joining                 | dept_name | designation | emp_name | projects
--------+--------+---------------------------------+-----------+-------------+----------+---------------
    120 |  30000 | 2021-04-01 07:00:00.000000+0000 |       CSE |     Manager |     Asha |       Research
    123 |  22500 | 2020-08-01 07:00:00.000000+0000 |       CSE |         Emp |  Samarth | Data Migration
    122 |  22000 | 2019-05-01 07:00:00.000000+0000 |       CSE |         Emp |    Tarun |  Data analysis
    121 |  20000 | 2019-04-20 07:00:00.000000+0000 |       CSE |         Emp |    Kiran |      Reporting
    124 |  21000 | 2019-06-01 07:00:00.000000+0000 |       CSE |         Emp |    Rohan |       Research

(5 rows)
```

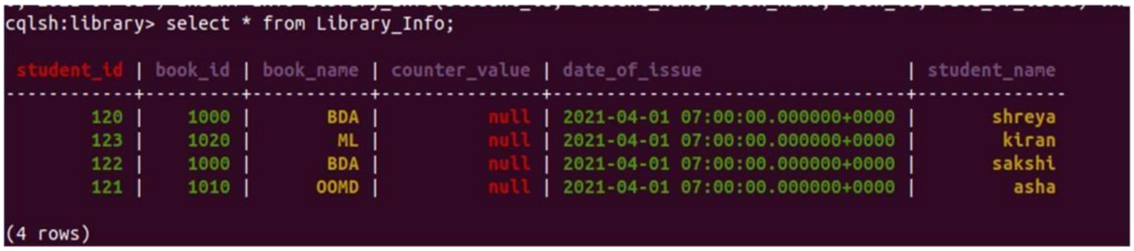
Program 2: Create a Data set either structured/Semi-Structured/Unstructured from twitter/Facebook etc. to perform various DB operations using Cassandra.



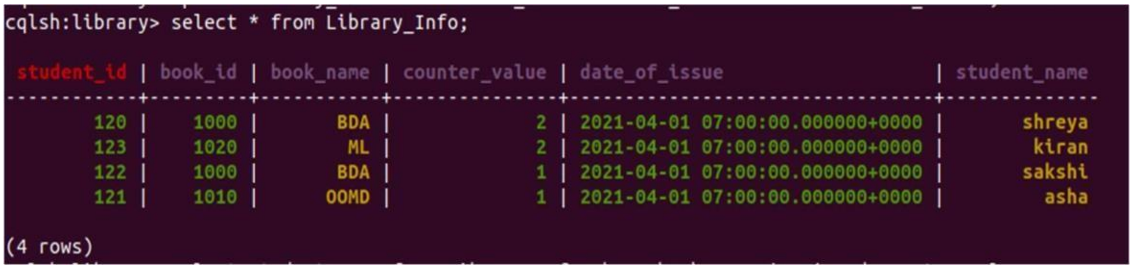
```
cqlsh> describe keyspaces;

employee  system_auth         system_schema  system_views
system    system_distributed  system_traces  system_virtual_schema

cqlsh> CREATE KEYSPACE Library WITH REPLICATION={'class':'SimpleStrategy','replication_factor':1};
cqlsh> describe keyspaces;

employee  system       system_distributed  system_traces  system_virtual_schema
library   system_auth  system_schema       system_views
```



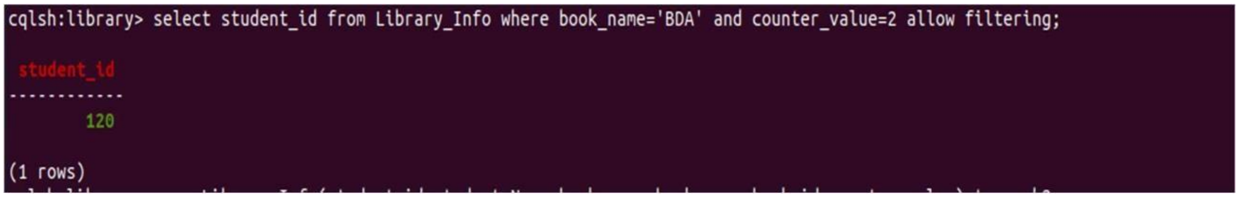
```
cqlsh:library> CREATE TABLE Library_Info (student_id int, student_Name text,book_name text,book_id int,Date_of_issue timestamp,primary key(student_id));
cqlsh:library> alter table Library_Info add counter_value counter;
cqlsh:library> describe tables;

library_info
```



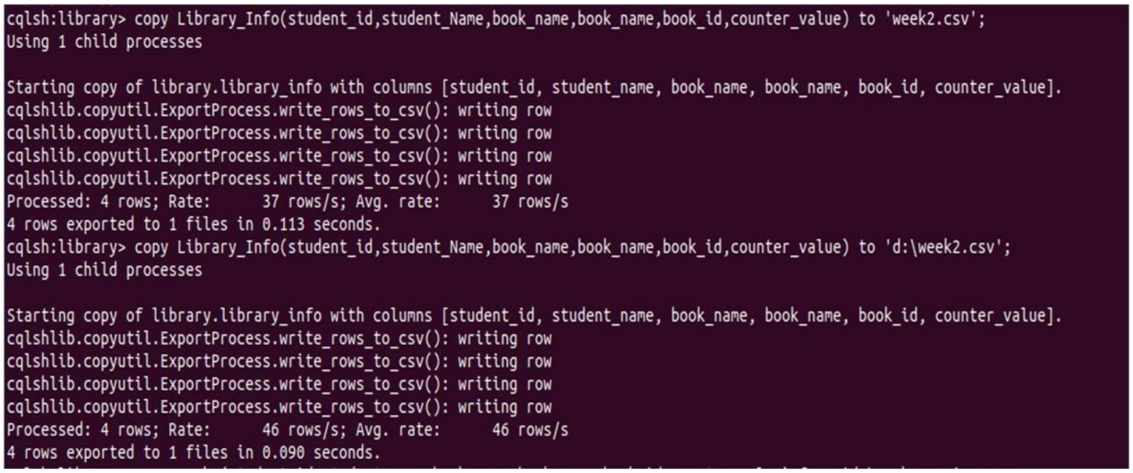
```
cqlsh:library> select * from Library_Info;

 student_id | book_id | book_name | counter_value | date_of_issue                   | student_name
------------+---------+-----------+---------------+---------------------------------+--------------
        120 |    1000 |       BDA |          null | 2021-04-01 07:00:00.000000+0000 |       shreya
        123 |    1020 |        ML |          null | 2021-04-01 07:00:00.000000+0000 |        kiran
        122 |    1000 |       BDA |          null | 2021-04-01 07:00:00.000000+0000 |       sakshi
        121 |    1010 |      OOMD |          null | 2021-04-01 07:00:00.000000+0000 |         asha

(4 rows)
```



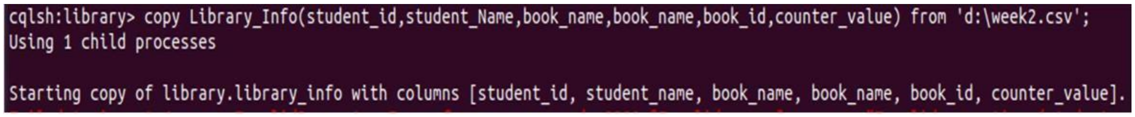
```
cqlsh:library> select * from Library_Info;

 student_id | book_id | book_name | counter_value | date_of_issue                   | student_name
------------+---------+-----------+---------------+---------------------------------+--------------
        120 |    1000 |       BDA |             2 | 2021-04-01 07:00:00.000000+0000 |       shreya
        123 |    1020 |        ML |             2 | 2021-04-01 07:00:00.000000+0000 |        kiran
        122 |    1000 |       BDA |             1 | 2021-04-01 07:00:00.000000+0000 |       sakshi
        121 |    1010 |      OOMD |             1 | 2021-04-01 07:00:00.000000+0000 |         asha

(4 rows)
```



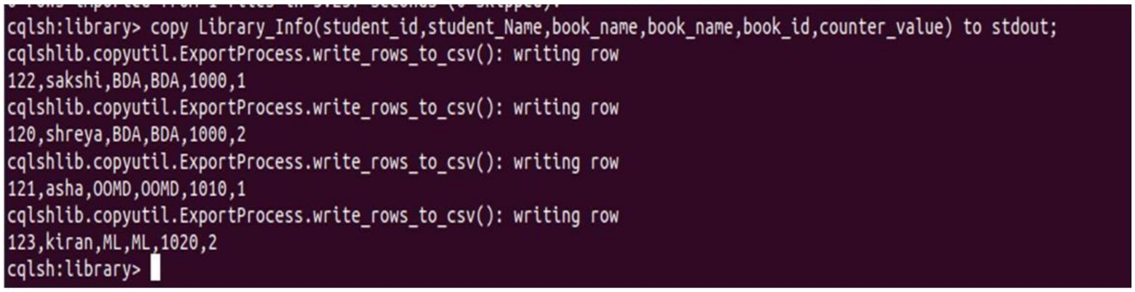
```
cqlsh:library> select student_id from Library_Info where book_name='BDA' and counter_value=2 allow filtering;

 student_id
------------
        120

(1 rows)
```



```
cqlsh:library> copy Library_Info(student_id,student_Name,book_name,book_name,book_id,counter_value) to 'week2.csv';
Using 1 child processes

Starting copy of library.library_info with columns [student_id, student_name, book_name, book_name, book_id, counter_value].
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
Processed: 4 rows; Rate:      37 rows/s; Avg. rate:      37 rows/s
4 rows exported to 1 files in 0.113 seconds.
cqlsh:library> copy Library_Info(student_id,student_Name,book_name,book_name,book_id,counter_value) to 'd:\week2.csv';
Using 1 child processes

Starting copy of library.library_info with columns [student_id, student_name, book_name, book_name, book_id, counter_value].
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
Processed: 4 rows; Rate:      46 rows/s; Avg. rate:      46 rows/s
4 rows exported to 1 files in 0.090 seconds.
```



```
cqlsh:library> copy Library_Info(student_id,student_Name,book_name,book_name,book_id,counter_value) from 'd:\week2.csv';
Using 1 child processes

Starting copy of library.library_info with columns [student_id, student_name, book_name, book_name, book_id, counter_value].
```



```
cqlsh:library> copy Library_Info(student_id,student_Name,book_name,book_name,book_id,counter_value) to stdout;
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
122,sakshi,BDA,BDA,1000,1
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
120,shreya,BDA,BDA,1000,2
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
121,asha,OOMD,OOMD,1010,1
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
123,kiran,ML,ML,1020,2
cqlsh:library> 
```

Program 3: Mongo DB CRUD Operations

CREATE DATBASE IN MONGODB:

```
bmsce@bmsce-Precision-T1700:~$ mongo sh
    MongoDB shell version v3.6.8
    connecting to: mongodb://127.0.0.1:27017/sh
    Implicit session: session { "id" : UUID("1875dd28-6f10-4e6f-ae5c-4c2b351e2abe") }
    MongoDB server version: 3.6.8
    Server has startup warnings:
    2023-04-01T15:22:28.307+0530 I STORAGE  [initandlisten]
    2023-04-01T15:22:28.307+0530 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
    2023-04-01T15:22:28.307+0530 I STORAGE  [initandlisten] **        See http://dochub.mongodb.org/core/prodnotes-filesystem
    2023-04-01T15:22:35.278+0530 I CONTROL  [initandlisten]
    2023-04-01T15:22:35.278+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
    2023-04-01T15:22:35.278+0530 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
    2023-04-01T15:22:35.278+0530 I CONTROL  [initandlisten]
> use yathri_db
    switched to db yathri_db
> db
    yathri_db
> show dbs
    Neha        0.000GB
    Niharika_db  0.000GB
    abcd        0.000GB
    admin        0.000GB
    config       0.000GB
    local       0.000GB
```

```
        myDB        0.000GB
        sec         0.000GB
        student     0.000GB
        test        0.000GB
```

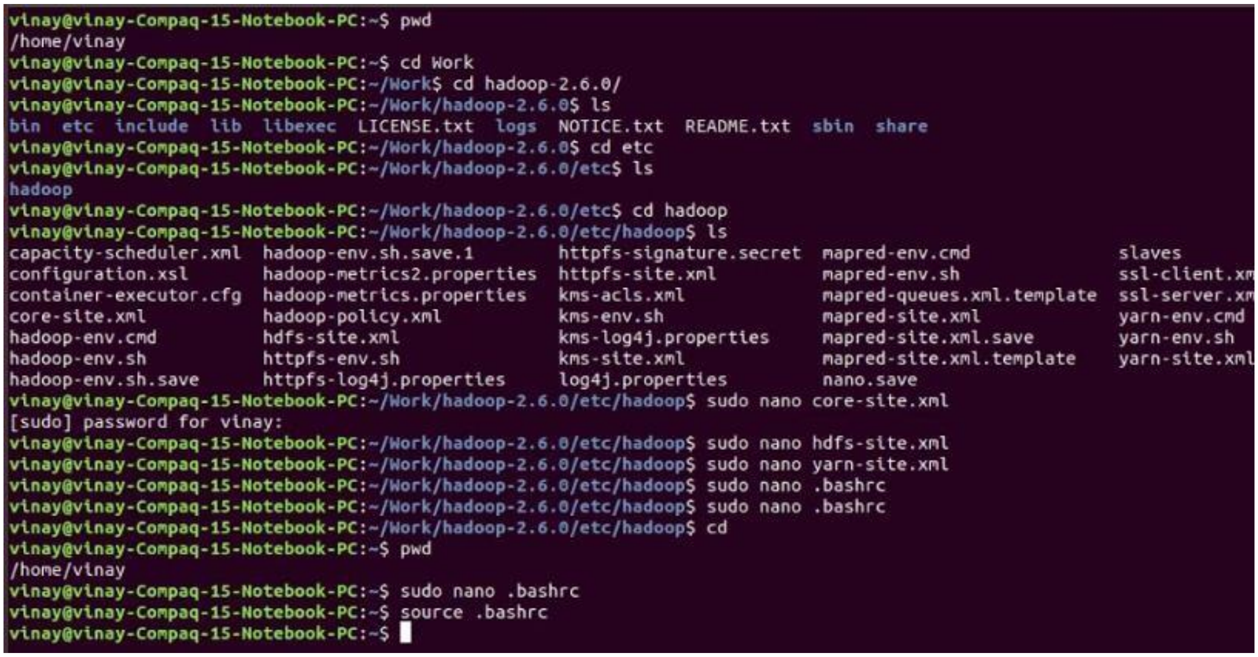
CRUD OPERATION:

```
> db.createCollection("Student")
        { "ok" : 1 }
> db.Student.drop()
        true
> show collections
> db.createCollection("Student")
        { "ok" : 1 }
>  show collections
        Student
> db.Student.insert({_id:1,Student_name:"AryaDavid",Grade:"VII",Hobbies:"InternetSurfing"})
        WriteResult({ "nInserted" : 1 })
> db.Student.find()
        { "_id" : 1, "Student_name" : "AryaDavid", "Grade" : "VII", "Hobbies" : "InternetSurfing"
}
>
db.Student.update({_id:1,Student_name:"AryaDavid",Grade:"VII"},{$set:{Hobbies:"Chess"}},{
upsert:true})
        WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Student.find()
        { "_id" : 1, "Student_name" : "AryaDavid", "Grade" : "VII", "Hobbies" : "Chess" }
> db.Student.find({Student_name: "AryaDavid"})
        { "_id" : 1, "Student_name" : "AryaDavid", "Grade" : "VII", "Hobbies" : "Chess" }
> db.Student.find({},{_id:0,Student_name:1,Grade:1})
        { "Student_name" : "AryaDavid", "Grade" : "VII" }
> db.Student.find({Grade:{$eq:"VII"}}).pretty()
```

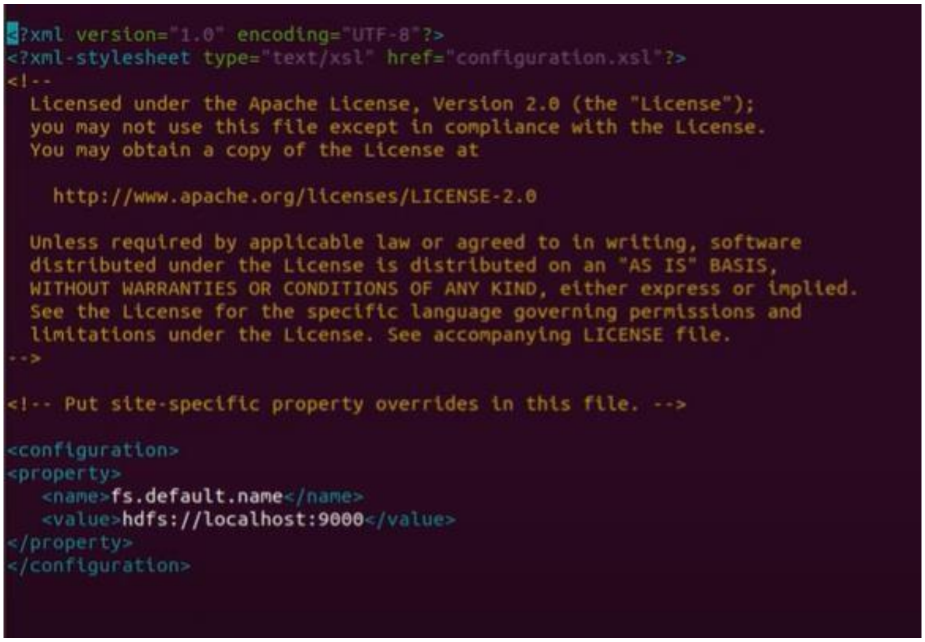
```
        {
                "_id" : 1,
                "Student_name" : "AryaDavid",
                "Grade" : "VII",
                "Hobbies" : "Chess"
        }
> db.Student.find({Hobbies:{$in:["Chess","Skating"]}}).pretty()
        {
                "_id" : 1,
                "Student_name" : "AryaDavid",
                "Grade" : "VII",
                "Hobbies" : "Chess"
        }
> db.Student.find({Student_name:/^M/}).pretty()
> db.Student.find({Student_name:/^A/}).pretty()
        {
                "_id" : 1,
                "Student_name" : "AryaDavid",
                "Grade" : "VII",
                "Hobbies" : "Chess"
        }
> db.Student.find({Student_name:/e/}).pretty()
> db.Student.find({Student_name:/i/}).pretty()
        {
                "_id" : 1,
                "Student_name" : "AryaDavid",
                "Grade" : "VII",
                "Hobbies" : "Chess"
        }
> db.Student.find().sort({Student_name: -1}).pretty()
        {
```

```
        "_id" : 1,
        "Student_name" : "AryaDavid",
        "Grade" : "VII",
        "Hobbies" : "Chess"
}
{
        "_id" : 2,
        "Student_name" : "Anu",
        "Grade" : "VI",
        "Hobbies" : "InternetSurfing"
}
```

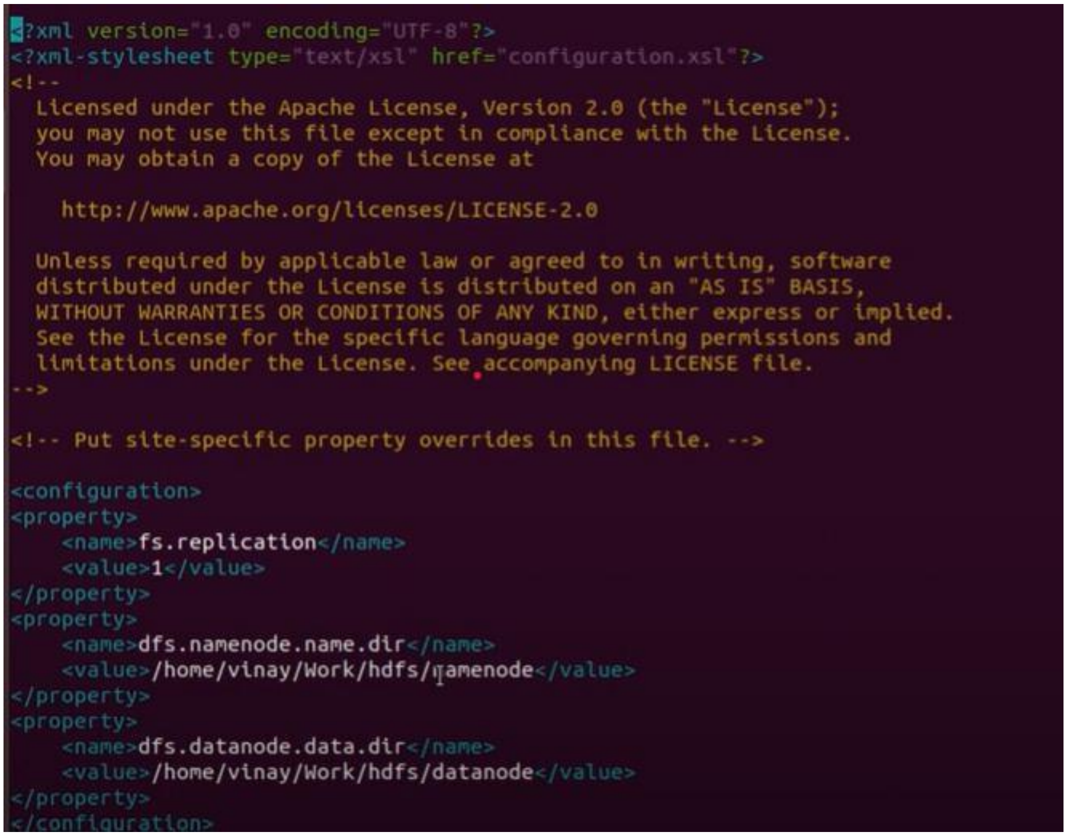
Program 4: Hadoop Installation



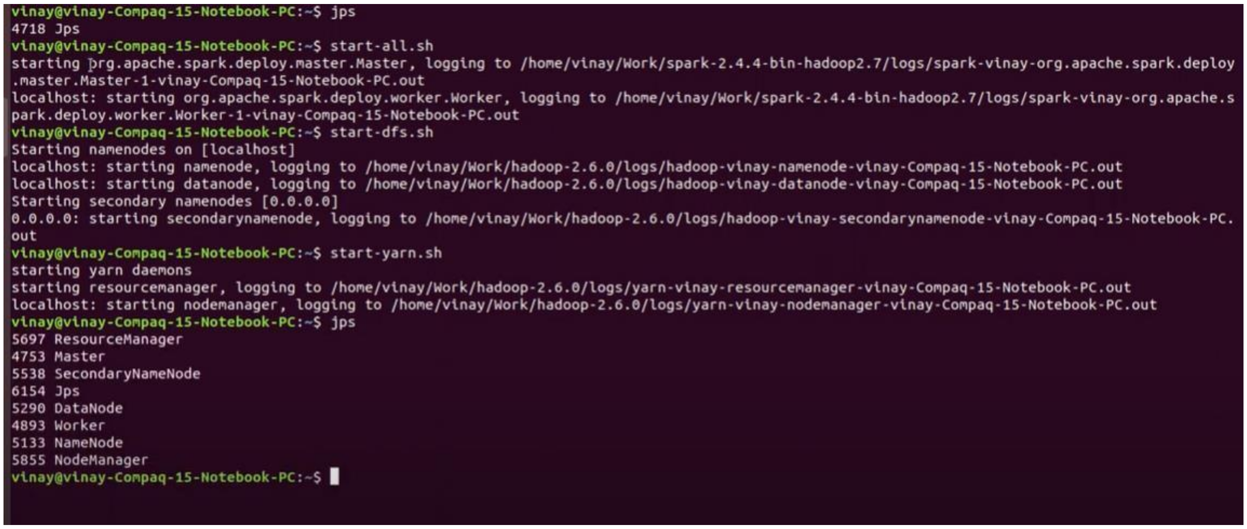
```
vinay@vinay-Compaq-15-Notebook-PC:~$ pwd
/home/vinay
vinay@vinay-Compaq-15-Notebook-PC:~$ cd Work
vinay@vinay-Compaq-15-Notebook-PC:~/Work$ cd hadoop-2.6.0/
vinay@vinay-Compaq-15-Notebook-PC:~/Work/hadoop-2.6.0$ ls
bin  etc  include  lib  libexec  LICENSE.txt  logs  NOTICE.txt  README.txt  sbin  share
vinay@vinay-Compaq-15-Notebook-PC:~/Work/hadoop-2.6.0$ cd etc
vinay@vinay-Compaq-15-Notebook-PC:~/Work/hadoop-2.6.0/etc$ ls
hadoop
vinay@vinay-Compaq-15-Notebook-PC:~/Work/hadoop-2.6.0/etc$ cd hadoop
vinay@vinay-Compaq-15-Notebook-PC:~/Work/hadoop-2.6.0/etc/hadoop$ ls
capacity-scheduler.xml  hadoop-env.sh.save.1        httpfs-signature.secret  mapred-env.cmd              slaves
configuration.xsl       hadoop-metrics2.properties  httpfs-site.xml          mapred-env.sh               ssl-client.xm
container-executor.cfg  hadoop-metrics.properties   kms-acls.xml             mapred-queues.xml.template  ssl-server.xm
core-site.xml           hadoop-policy.xml           kms-env.sh               mapred-site.xml             yarn-env.cmd
hadoop-env.cmd          hdfs-site.xml               kms-log4j.properties     mapred-site.xml.save        yarn-env.sh
hadoop-env.sh           httpfs-env.sh               kms-site.xml             mapred-site.xml.template    yarn-site.xml
hadoop-env.sh.save      httpfs-log4j.properties     log4j.properties         nano.save
vinay@vinay-Compaq-15-Notebook-PC:~/Work/hadoop-2.6.0/etc/hadoop$ sudo nano core-site.xml
[sudo] password for vinay:
vinay@vinay-Compaq-15-Notebook-PC:~/Work/hadoop-2.6.0/etc/hadoop$ sudo nano hdfs-site.xml
vinay@vinay-Compaq-15-Notebook-PC:~/Work/hadoop-2.6.0/etc/hadoop$ sudo nano yarn-site.xml
vinay@vinay-Compaq-15-Notebook-PC:~/Work/hadoop-2.6.0/etc/hadoop$ sudo nano .bashrc
vinay@vinay-Compaq-15-Notebook-PC:~/Work/hadoop-2.6.0/etc/hadoop$ sudo nano .bashrc
vinay@vinay-Compaq-15-Notebook-PC:~/Work/hadoop-2.6.0/etc/hadoop$ cd
vinay@vinay-Compaq-15-Notebook-PC:~$ pwd
/home/vinay
vinay@vinay-Compaq-15-Notebook-PC:~$ sudo nano .bashrc
vinay@vinay-Compaq-15-Notebook-PC:~$ source .bashrc
vinay@vinay-Compaq-15-Notebook-PC:~$ 
```



```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
   <name>fs.default.name</name>
   <value>hdfs://localhost:9000</value>
</property>
</configuration>
```

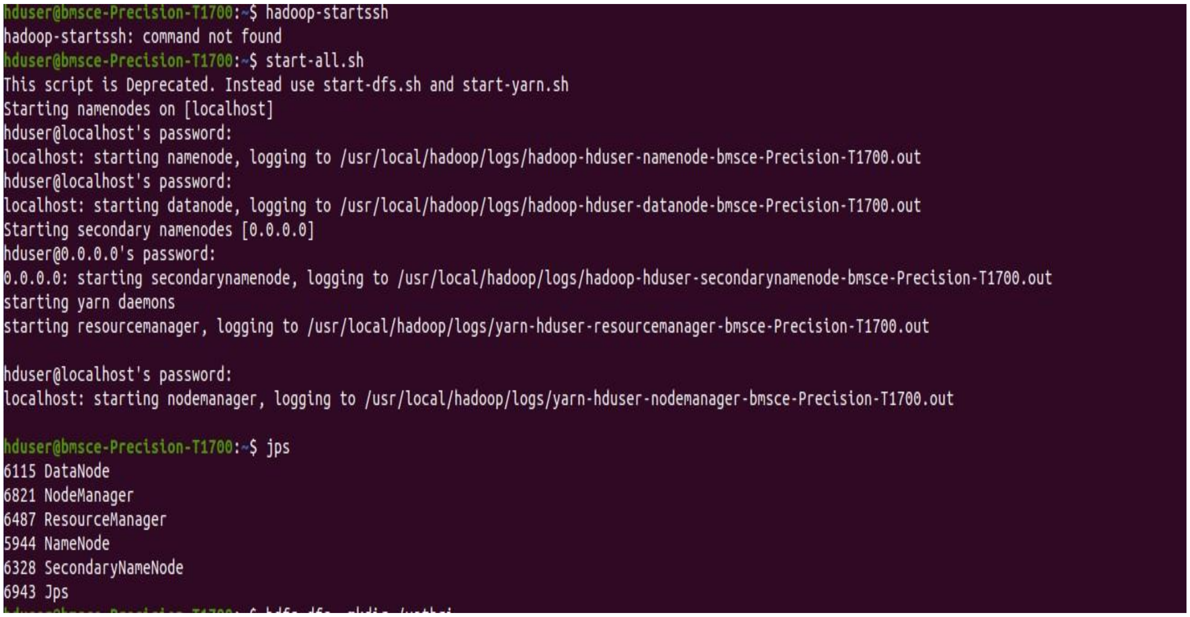


```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
    <name>fs.replication</name>
    <value>1</value>
</property>
<property>
    <name>dfs.namenode.name.dir</name>
    <value>/home/vinay/Work/hdfs/namenode</value>
</property>
<property>
    <name>dfs.datanode.data.dir</name>
    <value>/home/vinay/Work/hdfs/datanode</value>
</property>
</configuration>
```

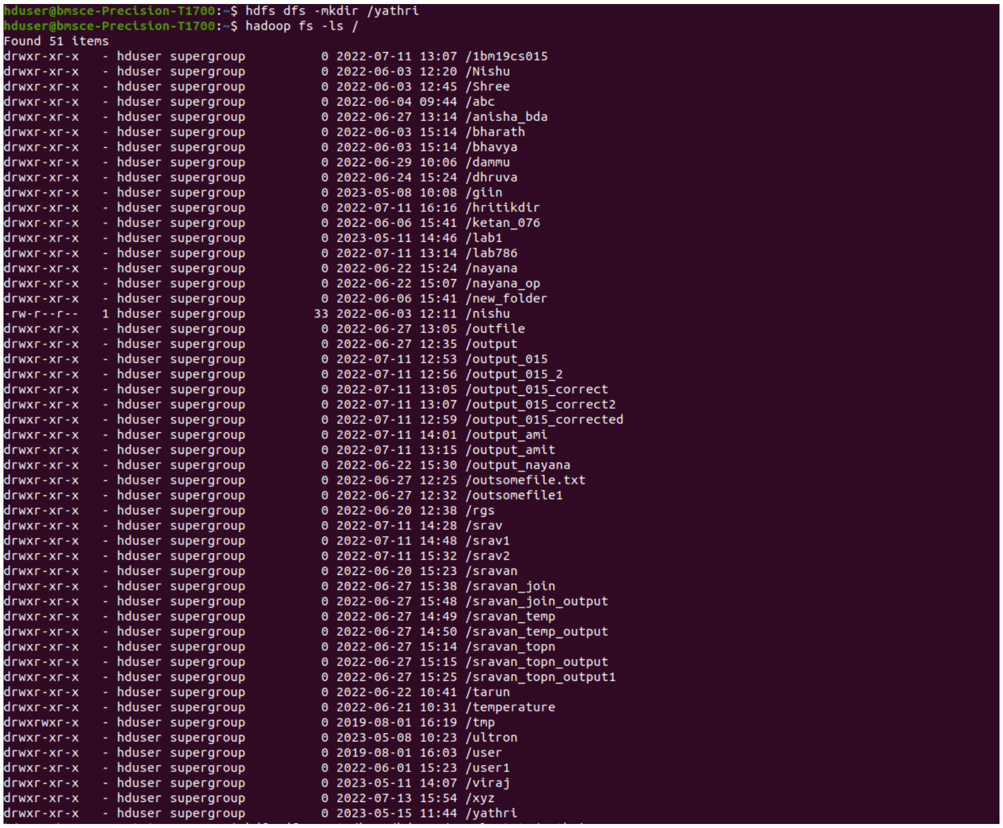


```
vinay@vinay-Compaq-15-Notebook-PC:~$ jps
4718 Jps
vinay@vinay-Compaq-15-Notebook-PC:~$ start-all.sh
starting org.apache.spark.deploy.master.Master, logging to /home/vinay/Work/spark-2.4.4-bin-hadoop2.7/logs/spark-vinay-org.apache.spark.deploy
.master.Master-1-vinay-Compaq-15-Notebook-PC.out
localhost: starting org.apache.spark.deploy.worker.Worker, logging to /home/vinay/Work/spark-2.4.4-bin-hadoop2.7/logs/spark-vinay-org.apache.s
park.deploy.worker.Worker-1-vinay-Compaq-15-Notebook-PC.out
vinay@vinay-Compaq-15-Notebook-PC:~$ start-dfs.sh
Starting namenodes on [localhost]
localhost: starting namenode, logging to /home/vinay/Work/hadoop-2.6.0/logs/hadoop-vinay-namenode-vinay-Compaq-15-Notebook-PC.out
localhost: starting datanode, logging to /home/vinay/Work/hadoop-2.6.0/logs/hadoop-vinay-datanode-vinay-Compaq-15-Notebook-PC.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /home/vinay/Work/hadoop-2.6.0/logs/hadoop-vinay-secondarynamenode-vinay-Compaq-15-Notebook-PC.
out
vinay@vinay-Compaq-15-Notebook-PC:~$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /home/vinay/Work/hadoop-2.6.0/logs/yarn-vinay-resourcemanager-vinay-Compaq-15-Notebook-PC.out
localhost: starting nodemanager, logging to /home/vinay/Work/hadoop-2.6.0/logs/yarn-vinay-nodemanager-vinay-Compaq-15-Notebook-PC.out
vinay@vinay-Compaq-15-Notebook-PC:~$ jps
5697 ResourceManager
4753 Master
5538 SecondaryNameNode
6154 Jps
5290 DataNode
4893 Worker
5133 NameNode
5855 NodeManager
vinay@vinay-Compaq-15-Notebook-PC:~$
```

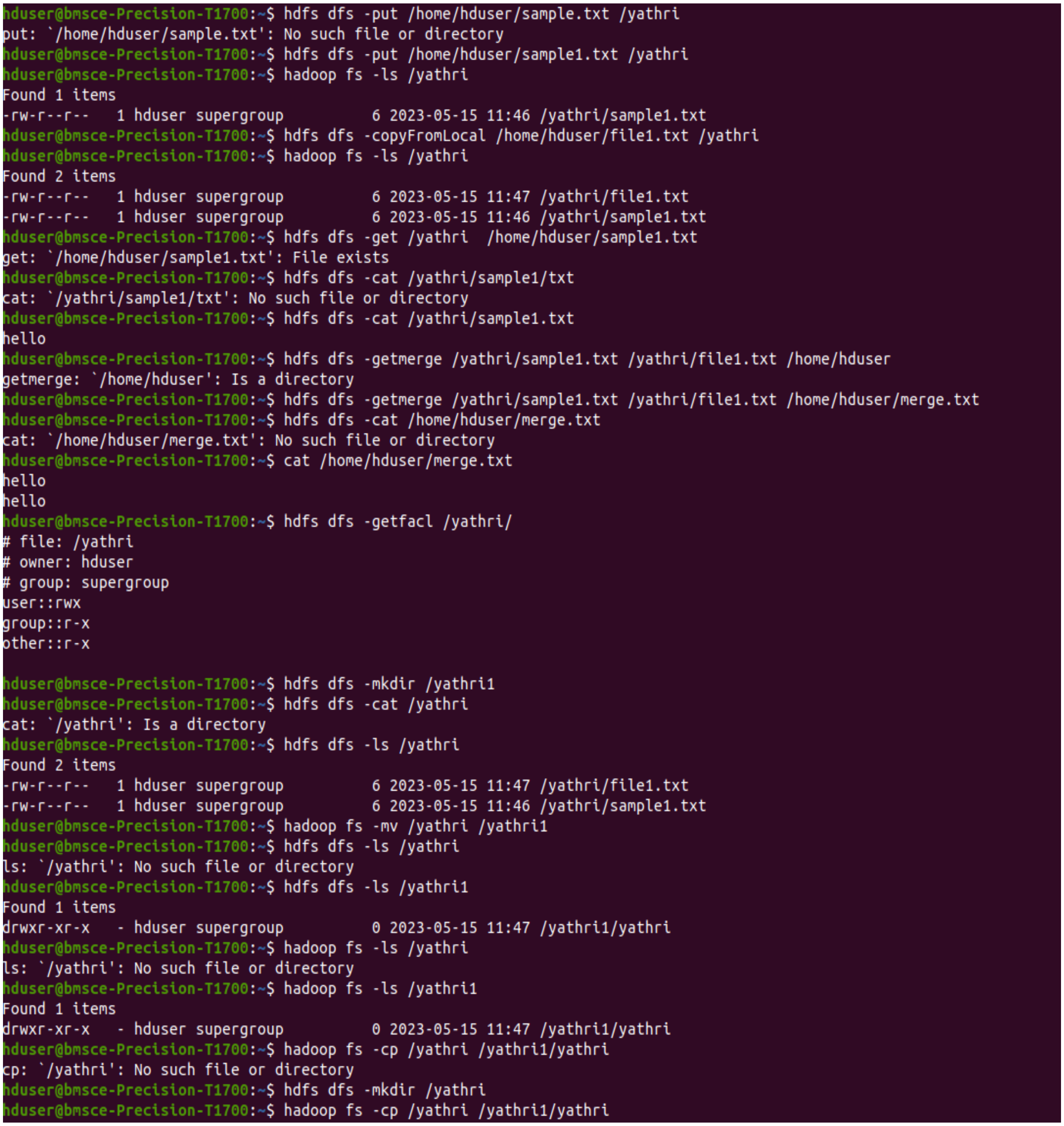
Program 5: Execution of HDFS Commands for interaction with Hadoop Environment.



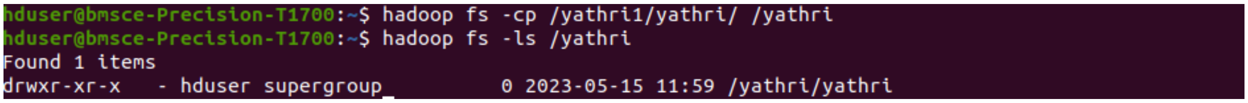
```
hduser@bmsce-Precision-T1700:~$ hadoop-startssh
hadoop-startssh: command not found
hduser@bmsce-Precision-T1700:~$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [localhost]
hduser@localhost's password:
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hduser-namenode-bmsce-Precision-T1700.out
hduser@localhost's password:
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hduser-datanode-bmsce-Precision-T1700.out
Starting secondary namenodes [0.0.0.0]
hduser@0.0.0.0's password:
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-hduser-secondarynamenode-bmsce-Precision-T1700.out
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hduser-resourcemanager-bmsce-Precision-T1700.out

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

hduser@bmsce-Precision-T1700:~$ jps
6115 DataNode
6821 NodeManager
6487 ResourceManager
5944 NameNode
6328 SecondaryNameNode
6943 Jps
```



```
hduser@bmsce-Precision-T1700:~$ hdfs dfs -mkdir /yathri
hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /
Found 51 items
drwxr-xr-x   - hduser supergroup          0 2022-07-11 13:07 /1bm19cs015
drwxr-xr-x   - hduser supergroup          0 2022-06-03 12:20 /Nishu
drwxr-xr-x   - hduser supergroup          0 2022-06-03 12:45 /Shree
drwxr-xr-x   - hduser supergroup          0 2022-06-04 09:44 /abc
drwxr-xr-x   - hduser supergroup          0 2022-06-27 13:14 /anisha_bda
drwxr-xr-x   - hduser supergroup          0 2022-06-03 15:14 /bharath
drwxr-xr-x   - hduser supergroup          0 2022-06-03 15:14 /bhavya
drwxr-xr-x   - hduser supergroup          0 2022-06-29 10:06 /dammu
drwxr-xr-x   - hduser supergroup          0 2022-06-24 15:24 /dhruva
drwxr-xr-x   - hduser supergroup          0 2023-05-08 10:08 /giin
drwxr-xr-x   - hduser supergroup          0 2022-07-11 16:16 /hritikdir
drwxr-xr-x   - hduser supergroup          0 2022-06-06 15:41 /ketan_076
drwxr-xr-x   - hduser supergroup          0 2023-05-11 14:46 /lab1
drwxr-xr-x   - hduser supergroup          0 2022-07-11 13:14 /lab786
drwxr-xr-x   - hduser supergroup          0 2022-06-22 15:24 /nayana
drwxr-xr-x   - hduser supergroup          0 2022-06-22 15:07 /nayana_op
drwxr-xr-x   - hduser supergroup          0 2022-06-06 15:41 /new_folder
-rw-r--r--   1 hduser supergroup         33 2022-06-03 12:11 /nishu
drwxr-xr-x   - hduser supergroup          0 2022-06-27 13:05 /outfile
drwxr-xr-x   - hduser supergroup          0 2022-06-27 12:35 /output
drwxr-xr-x   - hduser supergroup          0 2022-07-11 12:53 /output_015
drwxr-xr-x   - hduser supergroup          0 2022-07-11 12:56 /output_015_2
drwxr-xr-x   - hduser supergroup          0 2022-07-11 13:05 /output_015_correct
drwxr-xr-x   - hduser supergroup          0 2022-07-11 13:07 /output_015_correct2
drwxr-xr-x   - hduser supergroup          0 2022-07-11 12:59 /output_015_corrected
drwxr-xr-x   - hduser supergroup          0 2022-07-11 14:01 /output_ami
drwxr-xr-x   - hduser supergroup          0 2022-07-11 13:15 /output_amit
drwxr-xr-x   - hduser supergroup          0 2022-06-22 15:30 /output_nayana
drwxr-xr-x   - hduser supergroup          0 2022-06-27 12:25 /outsomefile.txt
drwxr-xr-x   - hduser supergroup          0 2022-06-27 12:32 /outsomefile1
drwxr-xr-x   - hduser supergroup          0 2022-06-20 12:38 /rgs
drwxr-xr-x   - hduser supergroup          0 2022-07-11 14:28 /srav
drwxr-xr-x   - hduser supergroup          0 2022-07-11 14:48 /srav1
drwxr-xr-x   - hduser supergroup          0 2022-07-11 15:32 /srav2
drwxr-xr-x   - hduser supergroup          0 2022-06-20 15:23 /sravan
drwxr-xr-x   - hduser supergroup          0 2022-06-27 15:38 /sravan_join
drwxr-xr-x   - hduser supergroup          0 2022-06-27 15:48 /sravan_join_output
drwxr-xr-x   - hduser supergroup          0 2022-06-27 14:49 /sravan_temp
drwxr-xr-x   - hduser supergroup          0 2022-06-27 14:50 /sravan_temp_output
drwxr-xr-x   - hduser supergroup          0 2022-06-27 15:14 /sravan_topn
drwxr-xr-x   - hduser supergroup          0 2022-06-27 15:15 /sravan_topn_output
drwxr-xr-x   - hduser supergroup          0 2022-06-27 15:25 /sravan_topn_output1
drwxr-xr-x   - hduser supergroup          0 2022-06-22 10:41 /tarun
drwxr-xr-x   - hduser supergroup          0 2022-06-21 10:31 /temperature
drwxrwxr-x   - hduser supergroup          0 2019-08-01 16:19 /tmp
drwxr-xr-x   - hduser supergroup          0 2023-05-08 10:23 /ultron
drwxr-xr-x   - hduser supergroup          0 2019-08-01 16:03 /user
drwxr-xr-x   - hduser supergroup          0 2022-06-01 15:23 /user1
drwxr-xr-x   - hduser supergroup          0 2023-05-11 14:07 /viraj
drwxr-xr-x   - hduser supergroup          0 2022-07-13 15:54 /xyz
drwxr-xr-x   - hduser supergroup          0 2023-05-15 11:44 /yathri
```



```
hduser@bmsce-Precision-T1700:~$ hdfs dfs -put /home/hduser/sample.txt /yathri
put: `/home/hduser/sample.txt': No such file or directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs -put /home/hduser/sample1.txt /yathri
hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /yathri
Found 1 items
-rw-r--r--   1 hduser supergroup          6 2023-05-15 11:46 /yathri/sample1.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs -copyFromLocal /home/hduser/file1.txt /yathri
hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /yathri
Found 2 items
-rw-r--r--   1 hduser supergroup          6 2023-05-15 11:47 /yathri/file1.txt
-rw-r--r--   1 hduser supergroup          6 2023-05-15 11:46 /yathri/sample1.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs -get /yathri  /home/hduser/sample1.txt
get: `/home/hduser/sample1.txt': File exists
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /yathri/sample1/txt
cat: `/yathri/sample1/txt': No such file or directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /yathri/sample1.txt
hello
hduser@bmsce-Precision-T1700:~$ hdfs dfs -getmerge /yathri/sample1.txt /yathri/file1.txt /home/hduser
getmerge: `/home/hduser': Is a directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs -getmerge /yathri/sample1.txt /yathri/file1.txt /home/hduser/merge.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /home/hduser/merge.txt
cat: `/home/hduser/merge.txt': No such file or directory
hduser@bmsce-Precision-T1700:~$ cat /home/hduser/merge.txt
hello
hello
hduser@bmsce-Precision-T1700:~$ hdfs dfs -getfacl /yathri/
# file: /yathri
# owner: hduser
# group: supergroup
user::rwx
group::r-x
other::r-x

hduser@bmsce-Precision-T1700:~$ hdfs dfs -mkdir /yathri1
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /yathri
cat: `/yathri': Is a directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /yathri
Found 2 items
-rw-r--r--   1 hduser supergroup          6 2023-05-15 11:47 /yathri/file1.txt
-rw-r--r--   1 hduser supergroup          6 2023-05-15 11:46 /yathri/sample1.txt
hduser@bmsce-Precision-T1700:~$ hadoop fs -mv /yathri /yathri1
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /yathri
ls: `/yathri': No such file or directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /yathri1
Found 1 items
drwxr-xr-x   - hduser supergroup          0 2023-05-15 11:47 /yathri1/yathri
hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /yathri
ls: `/yathri': No such file or directory
hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /yathri1
Found 1 items
drwxr-xr-x   - hduser supergroup          0 2023-05-15 11:47 /yathri1/yathri
hduser@bmsce-Precision-T1700:~$ hadoop fs -cp /yathri /yathri1/yathri
cp: `/yathri': No such file or directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs -mkdir /yathri
hduser@bmsce-Precision-T1700:~$ hadoop fs -cp /yathri /yathri1/yathri
```



```
hduser@bmsce-Precision-T1700:~$ hadoop fs -cp /yathri1/yathri/ /yathri
hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /yathri
Found 1 items
drwxr-xr-x   - hduser supergroup          0 2023-05-15 11:59 /yathri/yathri
```

Program 6: Create a Map Reduce program to

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

AverageMapper:

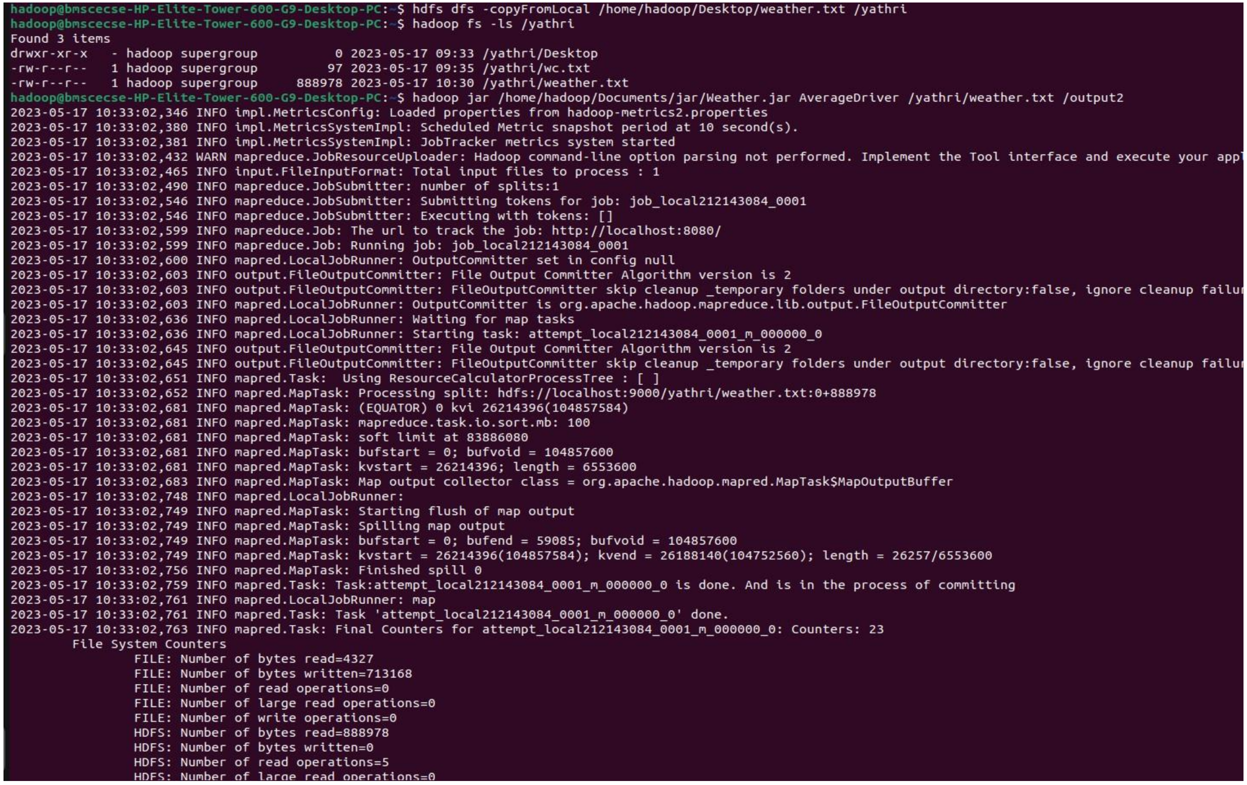
```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

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

  public void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text, IntWritable>.Context
context) throws IOException, InterruptedException {
    int temperature;
    String line = value.toString();
    String year = line.substring(15, 19);
    if (line.charAt(87) == '+') {
      temperature = Integer.parseInt(line.substring(88, 92));
    } else {
      temperature = Integer.parseInt(line.substring(87, 92));
    }
    String quality = line.substring(92, 93);
    if (temperature != 9999 && quality.matches("[01459]"))
      context.write(new Text(year), new IntWritable(temperature));
  }
}
```

AverageReducer:

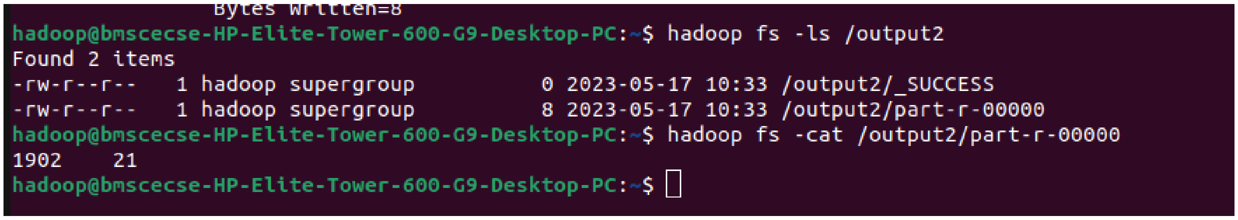
```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

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

AverageDriver:

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

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



```
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hdfs dfs -copyFromLocal /home/hadoop/Desktop/weather.txt /yathri
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -ls /yathri
Found 3 items
drwxr-xr-x   - hadoop supergroup          0 2023-05-17 09:33 /yathri/Desktop
-rw-r--r--   1 hadoop supergroup         97 2023-05-17 09:35 /yathri/wc.txt
-rw-r--r--   1 hadoop supergroup     888978 2023-05-17 10:30 /yathri/weather.txt
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop jar /home/hadoop/Documents/jar/Weather.jar AverageDriver /yathri/weather.txt /output2
2023-05-17 10:33:02,346 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2023-05-17 10:33:02,380 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2023-05-17 10:33:02,381 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2023-05-17 10:33:02,432 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your appl
2023-05-17 10:33:02,465 INFO input.FileInputFormat: Total input files to process : 1
2023-05-17 10:33:02,490 INFO mapreduce.JobSubmitter: number of splits:1
2023-05-17 10:33:02,546 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local212143084_0001
2023-05-17 10:33:02,546 INFO mapreduce.JobSubmitter: Executing with tokens: []
2023-05-17 10:33:02,599 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2023-05-17 10:33:02,599 INFO mapreduce.Job: Running job: job_local212143084_0001
2023-05-17 10:33:02,600 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2023-05-17 10:33:02,603 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2023-05-17 10:33:02,603 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failu
2023-05-17 10:33:02,603 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2023-05-17 10:33:02,636 INFO mapred.LocalJobRunner: Waiting for map tasks
2023-05-17 10:33:02,636 INFO mapred.LocalJobRunner: Starting task: attempt_local212143084_0001_m_000000_0
2023-05-17 10:33:02,645 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2023-05-17 10:33:02,645 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failu
2023-05-17 10:33:02,651 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2023-05-17 10:33:02,652 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/yathri/weather.txt:0+888978
2023-05-17 10:33:02,681 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2023-05-17 10:33:02,681 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2023-05-17 10:33:02,681 INFO mapred.MapTask: soft limit at 83886080
2023-05-17 10:33:02,681 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2023-05-17 10:33:02,681 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2023-05-17 10:33:02,683 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
2023-05-17 10:33:02,748 INFO mapred.LocalJobRunner:
2023-05-17 10:33:02,749 INFO mapred.MapTask: Starting flush of map output
2023-05-17 10:33:02,749 INFO mapred.MapTask: Spilling map output
2023-05-17 10:33:02,749 INFO mapred.MapTask: bufstart = 0; bufend = 59085; bufvoid = 104857600
2023-05-17 10:33:02,749 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 26188140(104752560); length = 26257/6553600
2023-05-17 10:33:02,756 INFO mapred.MapTask: Finished spill 0
2023-05-17 10:33:02,759 INFO mapred.Task: Task:attempt_local212143084_0001_m_000000_0 is done. And is in the process of committing
2023-05-17 10:33:02,761 INFO mapred.LocalJobRunner: map
2023-05-17 10:33:02,761 INFO mapred.Task: Task 'attempt_local212143084_0001_m_000000_0' done.
2023-05-17 10:33:02,763 INFO mapred.Task: Final Counters for attempt_local212143084_0001_m_000000_0: Counters: 23
        File System Counters
                FILE: Number of bytes read=4327
                FILE: Number of bytes written=713168
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=888978
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=5
                HDFS: Number of large read operations=0
```



```
                Bytes Written=8
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -ls /output2
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2023-05-17 10:33 /output2/_SUCCESS
-rw-r--r--   1 hadoop supergroup          8 2023-05-17 10:33 /output2/part-r-00000
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -cat /output2/part-r-00000
1902    21
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ 
```

b) find the mean max temperature for every month

MeanMaxMapper:

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

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

  public void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text,
IntWritable>.Context context) throws IOException, InterruptedException {
    int temperature;
    String line = value.toString();
    String month = line.substring(19, 21);
    if (line.charAt(87) == '+') {
      temperature = Integer.parseInt(line.substring(88, 92));
    } else {
      temperature = Integer.parseInt(line.substring(87, 92));
    }
    String quality = line.substring(92, 93);
    if (temperature != 9999 && quality.matches("[01459]"))
      context.write(new Text(month), new IntWritable(temperature));
  }
}
```

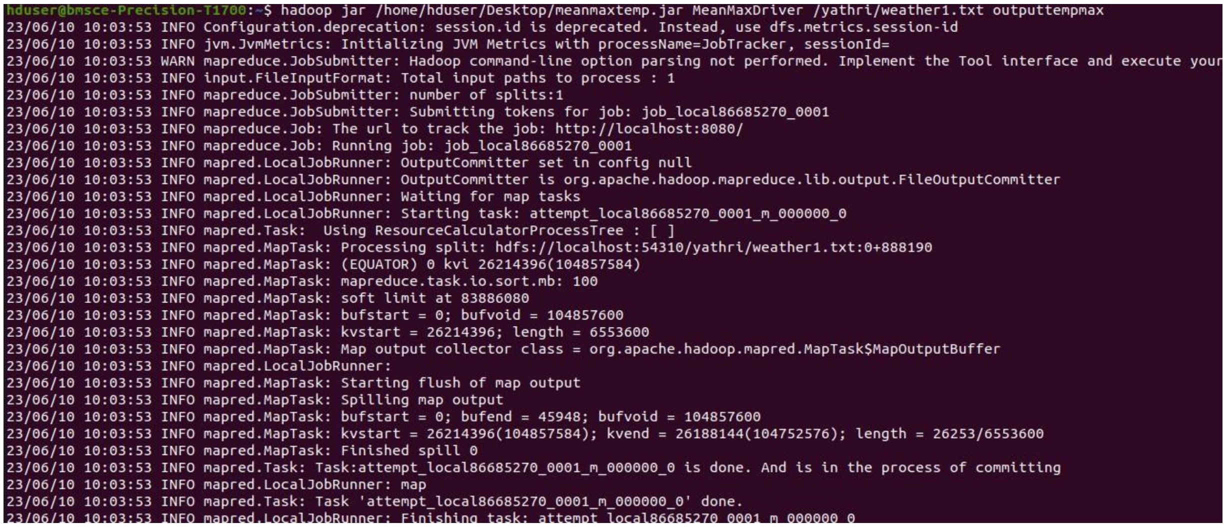
MeanMaxReducer:

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class MeanMaxReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
public void reduce(Text key, Iterable<IntWritable> values, Reducer<Text, IntWritable, Text,
IntWritable>.Context context) throws IOException, InterruptedException {
    int max_temp = 0;
    int total_temp = 0;
    int count = 0;
    int days = 0;
    for (IntWritable value : values) {
      int temp = value.get();
      if (temp > max_temp)
```

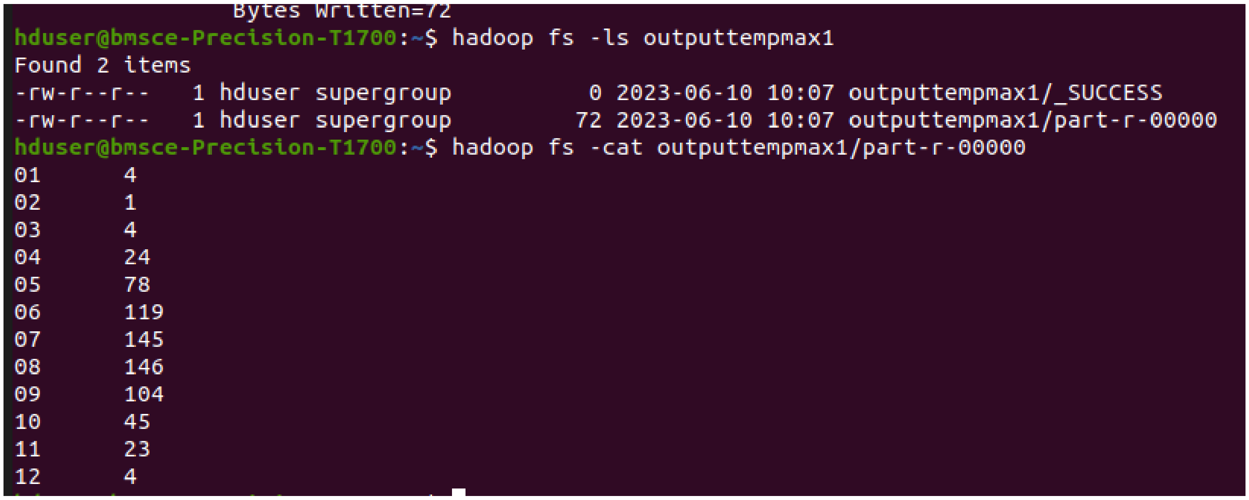
```
        max_temp = temp;
      count++;
      if (count == 3) {
        total_temp += max_temp;
        max_temp = 0;
        count = 0;
        days++;
      }
    }
    context.write(key, new IntWritable(total_temp / days));
  }
}
```

MeanMaxDriver:

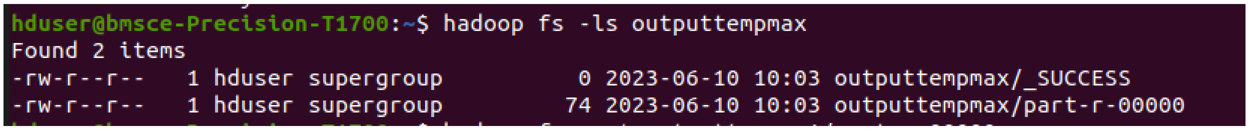
```
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class MeanMaxDriver {
  public static void main(String[] args) throws Exception {
    if (args.length != 2) {
      System.err.println("Please Enter the input and output parameters");
      System.exit(-1);
    }
    Job = new Job();
    job.setJarByClass(MeanMaxDriver.class);
    job.setJobName("Max temperature");
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    job.setMapperClass(MeanMaxMapper.class);
    job.setReducerClass(MeanMaxReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```



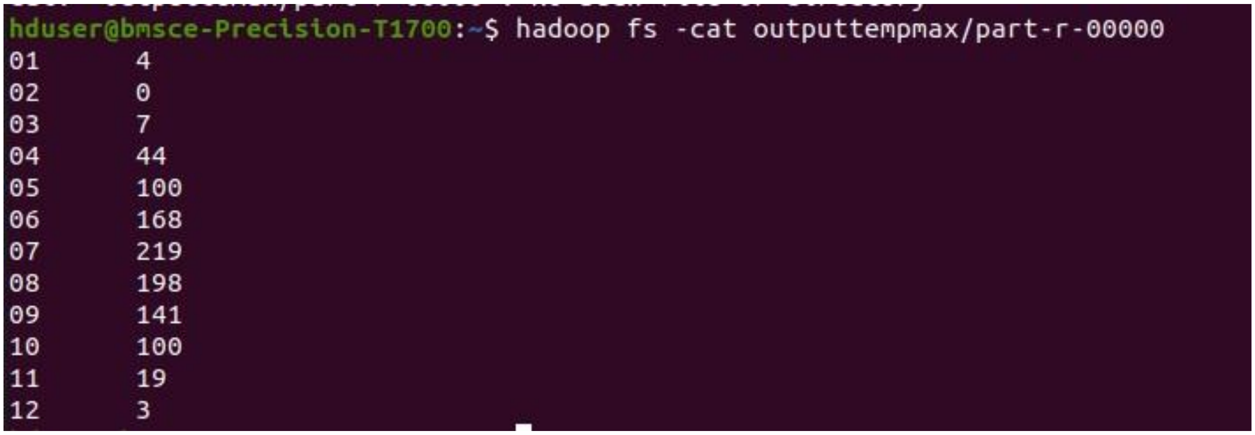
```
hduser@bmsce-Precision-T1700:~$ hadoop jar /home/hduser/Desktop/meanmaxtemp.jar MeanMaxDriver /yathri/weather1.txt outputtempmax
23/06/10 10:03:53 INFO Configuration.deprecation: session.id is deprecated. Instead, use dfs.metrics.session-id
23/06/10 10:03:53 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker, sessionId=
23/06/10 10:03:53 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your
23/06/10 10:03:53 INFO input.FileInputFormat: Total input paths to process : 1
23/06/10 10:03:53 INFO mapreduce.JobSubmitter: number of splits:1
23/06/10 10:03:53 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local86685270_0001
23/06/10 10:03:53 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
23/06/10 10:03:53 INFO mapreduce.Job: Running job: job_local86685270_0001
23/06/10 10:03:53 INFO mapred.LocalJobRunner: OutputCommitter set in config null
23/06/10 10:03:53 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
23/06/10 10:03:53 INFO mapred.LocalJobRunner: Waiting for map tasks
23/06/10 10:03:53 INFO mapred.LocalJobRunner: Starting task: attempt_local86685270_0001_m_000000_0
23/06/10 10:03:53 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
23/06/10 10:03:53 INFO mapred.MapTask: Processing split: hdfs://localhost:54310/yathri/weather1.txt:0+888190
23/06/10 10:03:53 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
23/06/10 10:03:53 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
23/06/10 10:03:53 INFO mapred.MapTask: soft limit at 83886080
23/06/10 10:03:53 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
23/06/10 10:03:53 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
23/06/10 10:03:53 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
23/06/10 10:03:53 INFO mapred.LocalJobRunner:
23/06/10 10:03:53 INFO mapred.MapTask: Starting flush of map output
23/06/10 10:03:53 INFO mapred.MapTask: Spilling map output
23/06/10 10:03:53 INFO mapred.MapTask: bufstart = 0; bufend = 45948; bufvoid = 104857600
23/06/10 10:03:53 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 26188144(104752576); length = 26253/6553600
23/06/10 10:03:53 INFO mapred.MapTask: Finished spill 0
23/06/10 10:03:53 INFO mapred.Task: Task:attempt_local86685270_0001_m_000000_0 is done. And is in the process of committing
23/06/10 10:03:53 INFO mapred.LocalJobRunner: map
23/06/10 10:03:53 INFO mapred.Task: Task 'attempt_local86685270_0001_m_000000_0' done.
23/06/10 10:03:53 INFO mapred.LocalJobRunner: Finishing task: attempt_local86685270_0001_m_000000_0
```



```
                Bytes Written=72
hduser@bmsce-Precision-T1700:~$ hadoop fs -ls outputtempmax1
Found 2 items
-rw-r--r--   1 hduser supergroup          0 2023-06-10 10:07 outputtempmax1/_SUCCESS
-rw-r--r--   1 hduser supergroup         72 2023-06-10 10:07 outputtempmax1/part-r-00000
hduser@bmsce-Precision-T1700:~$ hadoop fs -cat outputtempmax1/part-r-00000
01      4
02      1
03      4
04      24
05      78
06      119
07      145
08      146
09      104
10      45
11      23
12      4
```



```
hduser@bmsce-Precision-T1700:~$ hadoop fs -ls outputtempmax
Found 2 items
-rw-r--r--   1 hduser supergroup          0 2023-06-10 10:03 outputtempmax/_SUCCESS
-rw-r--r--   1 hduser supergroup         74 2023-06-10 10:03 outputtempmax/part-r-00000
```



```
hduser@bmsce-Precision-T1700:~$ hadoop fs -cat outputtempmax/part-r-00000
01      4
02      0
03      7
04      44
05      100
06      168
07      219
08      198
09      141
10      100
11      19
12      3
```

Program 7: Create a Map Reduce program to sort the content in an alphabetic order listing only top 10 maximum occurrences of words.

TopNMapper:

```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;


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

  private Text word = new Text();

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

  public void map(Object key, Text value, Mapper<Object, Text, Text, IntWritable>.Context
context) throws IOException, InterruptedException {
    String cleanLine = value.toString().toLowerCase().replaceAll(this.tokens, " ");
    StringTokenizer itr = new StringTokenizer(cleanLine);
    while (itr.hasMoreTokens()) {
      this.word.set(itr.nextToken().trim());
      context.write(this.word, one);
    }
  }
}
```

TopNReducer:

```
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
import utils.MiscUtils;


public class TopNReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
  private Map<Text, IntWritable> countMap = new HashMap<>();

  public void reduce(Text key, Iterable<IntWritable> values, Reducer<Text, IntWritable, Text,
IntWritable>.Context context) throws IOException, InterruptedException {
    int sum = 0;
    for (IntWritable val : values)
      sum += val.get();
    this.countMap.put(new Text(key), new IntWritable(sum));
  }

  protected void cleanup(Reducer<Text, IntWritable, Text, IntWritable>.Context context)
throws IOException, InterruptedException {
    Map<Text, IntWritable> sortedMap = MiscUtils.sortByValues(this.countMap);
    int counter = 0;
    for (Text key : sortedMap.keySet()) {
      if (counter++ == 20)
        break;
      context.write(key, sortedMap.get(key));
    }
  }
}
```

TopnNDriver:

```
import java.io.IOException;
import java.util.StringTokenizer;
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.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;

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.getInstance(conf);
    job.setJobName("Top N");
    job.setJarByClass(TopN.class);
    job.setMapperClass(TopNMapper.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);
  }

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

    private Text word = new Text();

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

    public void map(Object key, Text value, Mapper<Object, Text, Text, IntWritable>.Context
context) throws IOException, InterruptedException {
      String cleanLine = value.toString().toLowerCase().replaceAll(this.tokens, " ");
      StringTokenizer itr = new StringTokenizer(cleanLine);
      while (itr.hasMoreTokens()) {
        this.word.set(itr.nextToken().trim());
        context.write(this.word, one);
      }
    }
  }
}
```

TopNCombiner:

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;


public class TopNCombiner extends Reducer<Text, IntWritable, Text, IntWritable> {
  public void reduce(Text key, Iterable<IntWritable> values, Reducer<Text, IntWritable, Text,
IntWritable>.Context context) throws IOException, InterruptedException {
    int sum = 0;
    for (IntWritable val : values)
      sum += val.get();
    context.write(key, new IntWritable(sum));
  }
}
```

Package util:

```
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());
}
});
Map<K, V> sortedMap = new LinkedHashMap<K, V>();
for (Map.Entry<K, V> entry : entries) {
sortedMap.put(entry.getKey(), entry.getValue());
}
return sortedMap;
}
}
```

Test.txt:

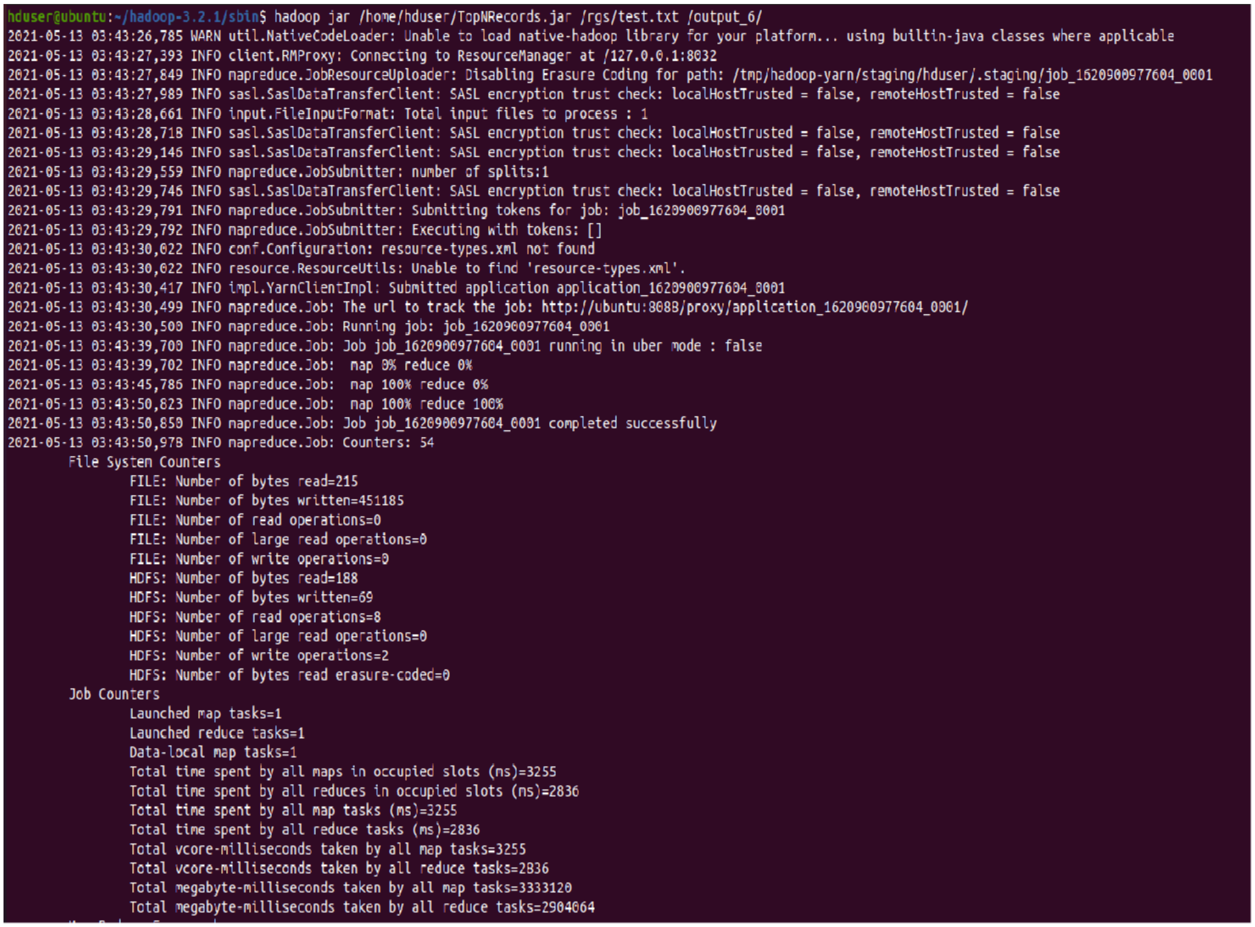
hi how are you

how is your job

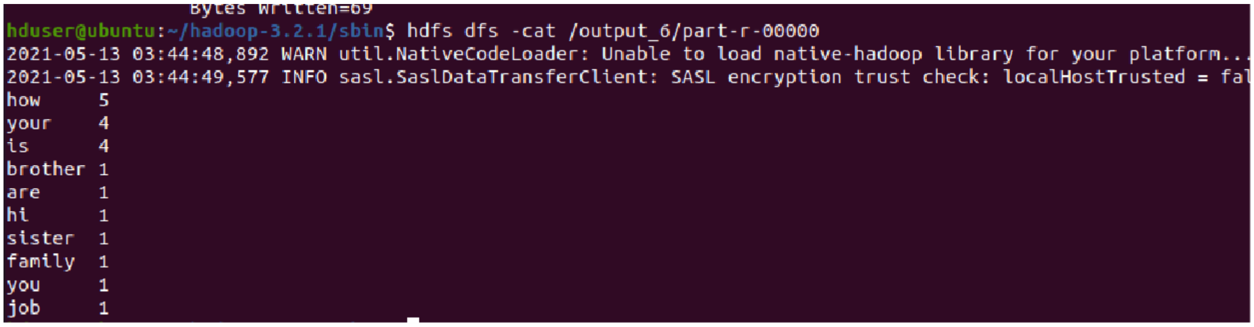
how is your family

how is your brother

how is your sister



```
hduser@ubuntu:~/hadoop-3.2.1/sbin$ hadoop jar /home/hduser/TopNRecords.jar /rgs/test.txt /output_6/
2021-05-13 03:43:26,785 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
2021-05-13 03:43:27,393 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0.1:8032
2021-05-13 03:43:27,849 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hduser/.staging/job_1620900977604_0001
2021-05-13 03:43:27,989 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-05-13 03:43:28,661 INFO input.FileInputFormat: Total input files to process : 1
2021-05-13 03:43:28,718 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-05-13 03:43:29,146 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-05-13 03:43:29,559 INFO mapreduce.JobSubmitter: number of splits:1
2021-05-13 03:43:29,746 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-05-13 03:43:29,791 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1620900977604_0001
2021-05-13 03:43:29,792 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-05-13 03:43:30,022 INFO conf.Configuration: resource-types.xml not found
2021-05-13 03:43:30,022 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-05-13 03:43:30,417 INFO impl.YarnClientImpl: Submitted application application_1620900977604_0001
2021-05-13 03:43:30,499 INFO mapreduce.Job: The url to track the job: http://ubuntu:8088/proxy/application_1620900977604_0001/
2021-05-13 03:43:30,500 INFO mapreduce.Job: Running job: job_1620900977604_0001
2021-05-13 03:43:39,700 INFO mapreduce.Job: Job job_1620900977604_0001 running in uber mode : false
2021-05-13 03:43:39,702 INFO mapreduce.Job:  map 0% reduce 0%
2021-05-13 03:43:45,786 INFO mapreduce.Job:  map 100% reduce 0%
2021-05-13 03:43:50,823 INFO mapreduce.Job:  map 100% reduce 100%
2021-05-13 03:43:50,850 INFO mapreduce.Job: Job job_1620900977604_0001 completed successfully
2021-05-13 03:43:50,978 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=215
                FILE: Number of bytes written=451185
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=188
                HDFS: Number of bytes written=69
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=3255
                Total time spent by all reduces in occupied slots (ms)=2836
                Total time spent by all map tasks (ms)=3255
                Total time spent by all reduce tasks (ms)=2836
                Total vcore-milliseconds taken by all map tasks=3255
                Total vcore-milliseconds taken by all reduce tasks=2836
                Total megabyte-milliseconds taken by all map tasks=3333120
                Total megabyte-milliseconds taken by all reduce tasks=2904064
```



```
                Bytes Written=69
hduser@ubuntu:~/hadoop-3.2.1/sbin$ hdfs dfs -cat /output_6/part-r-00000
2021-05-13 03:44:48,892 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform...
2021-05-13 03:44:49,577 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = fal
how     5
your    4
is      4
brother 1
are     1
hi      1
sister  1
family  1
you     1
job     1
```

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

JoinDriver.java:

```
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.*;
import org.apache.hadoop.mapred.lib.MultipleInputs;
import org.apache.hadoop.util.*;
public class JoinDriver extends Configured implements Tool {
public static class KeyPartitioner implements Partitioner<TextPair, Text> {
@Override
public void configure(JobConf job) {}
@Override
public int getPartition(TextPair key, Text value, int numPartitions) {
return (key.getFirst().hashCode() & Integer.MAX_VALUE) %
numPartitions;
}
}
@Override
public int run(String[] args) throws Exception {
if (args.length != 3) {
System.out.println("Usage: <Department Emp Strength input>
<Department Name input> <output>");
return -1;
}
JobConf conf = new JobConf(getConf(), getClass());
conf.setJobName("Join 'Department Emp Strength input' with 'Department Name input'");
Path AInputPath = new Path(args[0]);
Path BInputPath = new Path(args[1]);
Path outputPath = new Path(args[2]);
MultipleInputs.addInputPath(conf, AInputPath, TextInputFormat.class, Posts.class);
MultipleInputs.addInputPath(conf, BInputPath, TextInputFormat.class,User.class);
FileOutputFormat.setOutputPath(conf, outputPath);
conf.setPartitionerClass(KeyPartitioner.class);
conf.setOutputValueGroupingComparator(TextPair.FirstComparator.class);
conf.setMapOutputKeyClass(TextPair.class);
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,
Reporter reporter)
throws IOException
{
```

```
String valueString = value.toString();
String[] SingleNodeData = valueString.split("\t");
output.collect(new TextPair(SingleNodeData[0], "1"), new Text(SingleNodeData[1]));
}
}
```

Posts.java:

```
import java.io.IOException;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.*;
public class Posts extends MapReduceBase implements Mapper<LongWritable, Text, TextPair,
Text> {
@Override
public void map(LongWritable key, Text value, OutputCollector<TextPair, Text> output,
Reporter reporter) throws IOException
{
String valueString = value.toString();
String[] SingleNodeData = valueString.split("\t");
output.collect(new TextPair(SingleNodeData[3], "0"), new Text(SingleNodeData[9]));
}
}
```

TextPair.java:

```
import java.io.*;
import org.apache.hadoop.io.*;
public class TextPair implements WritableComparable<TextPair> {
private Text first;
private Text second;
public TextPair() {
set(new Text(), new Text());
```

```
}
public TextPair(String first, String second) { set(new Text(first), new Text(second)); }
public TextPair(Text first, Text second) {
set(first, second);
}
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, b2, s2, firstL2);
if (cmp != 0) {
return cmp;
}
return TEXT_COMPARATOR.compare(b1, s1 + firstL1, l1 - firstL1,
b2, s2 + firstL2, l2 - firstL2);
} catch (IOException e) {
throw new IllegalArgumentException(e);
}
}
}
static {
WritableComparator.define(TextPair.class, new Comparator());
}
public static class FirstComparator extends WritableComparator {
private static final Text.Comparator TEXT_COMPARATOR = new Text.Comparator();
public FirstComparator() {
super(TextPair.class);
}
```

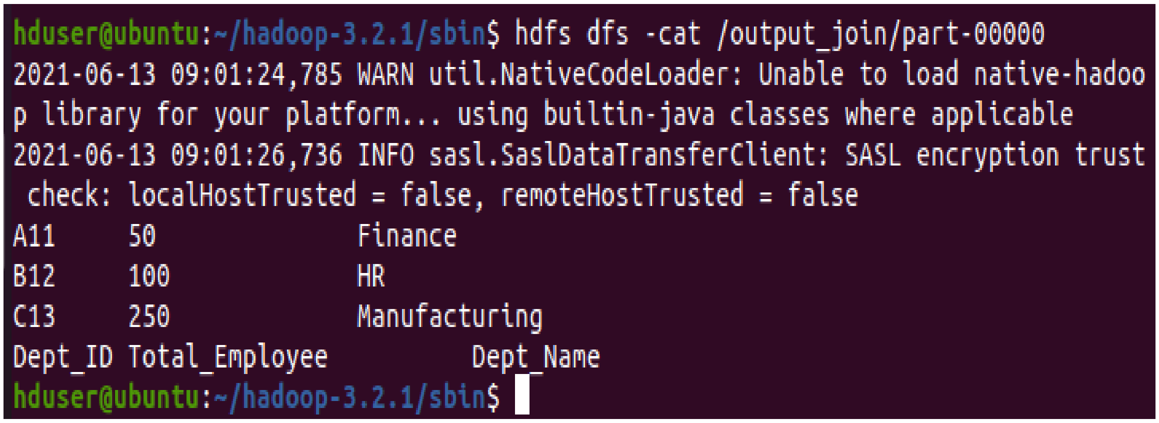
```
@Override
public int compare(byte[] b1, int s1, int l1,
byte[] b2, int s2, int l2) {
try {
int firstL1 = WritableUtils.decodeVIntSize(b1[s1]) + readVInt(b1, s1);
int firstL2 = WritableUtils.decodeVIntSize(b2[s2]) + readVInt(b2, s2);
return TEXT_COMPARATOR.compare(b1, s1, firstL1, b2, s2, firstL2);
} catch (IOException e) {
throw new IllegalArgumentException(e);
}
}
@Override
public int compare(WritableComparable a, WritableComparable b) {
if (a instanceof TextPair && b instanceof TextPair) {
return ((TextPair) a).first.compareTo(((TextPair) b).first);
}
return super.compare(a, b);
}
} }
```

DeptName.txt:

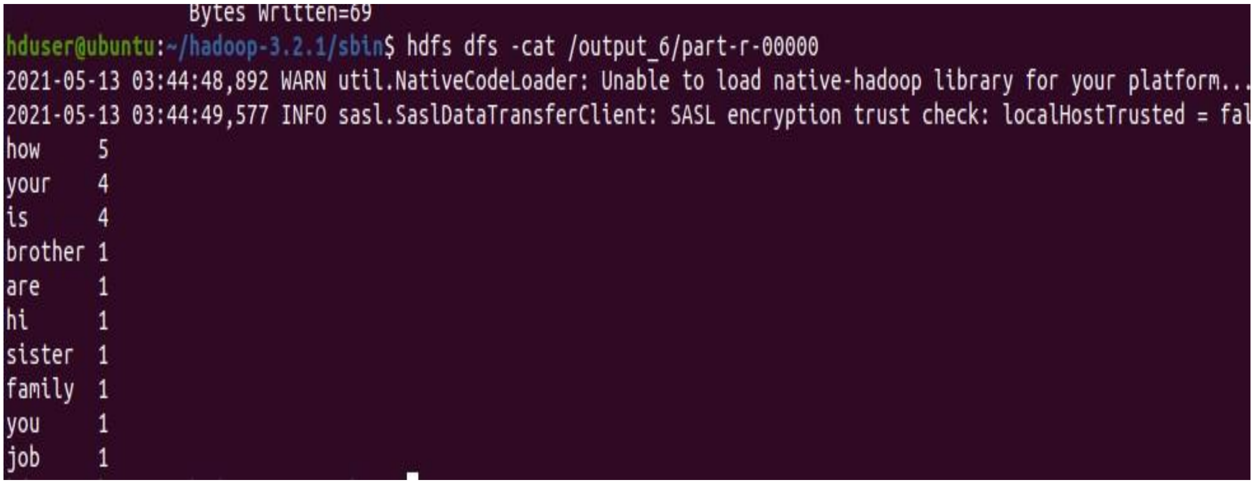
Dept_ID	Dept_Name
A11	Finance
B12	HR
C13	Manufacturing

DeptStrength:

Dept_ID	Total_Employee
A11	50
B12	100
C13	250

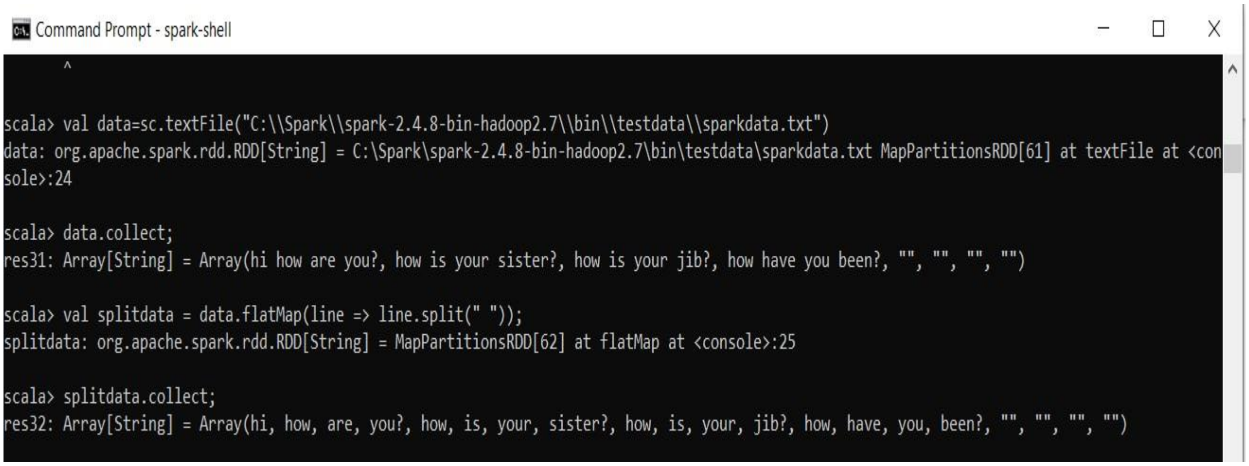


```
hduser@ubuntu:~/hadoop-3.2.1/sbin$ hdfs dfs -cat /output_join/part-00000
2021-06-13 09:01:24,785 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
2021-06-13 09:01:26,736 INFO sasl.SaslDataTransferClient: SASL encryption trust
 check: localHostTrusted = false, remoteHostTrusted = false
A11     50              Finance
B12     100             HR
C13     250             Manufacturing
Dept_ID Total_Employee          Dept_Name
hduser@ubuntu:~/hadoop-3.2.1/sbin$ 
```

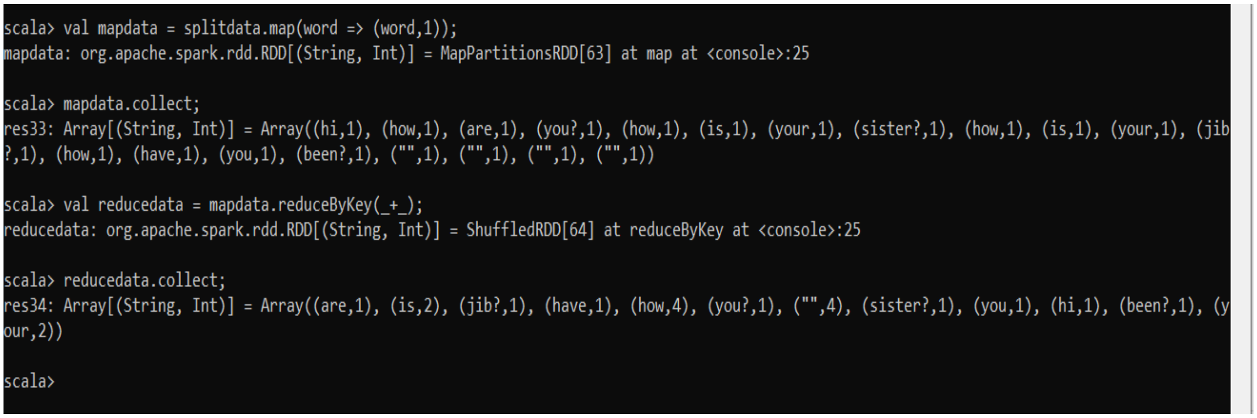


```
                Bytes Written=69
hduser@ubuntu:~/hadoop-3.2.1/sbin$ hdfs dfs -cat /output_6/part-r-00000
2021-05-13 03:44:48,892 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform...
2021-05-13 03:44:49,577 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = fal
how     5
your    4
is      4
brother 1
are     1
hi      1
sister  1
family  1
you     1
job     1
```

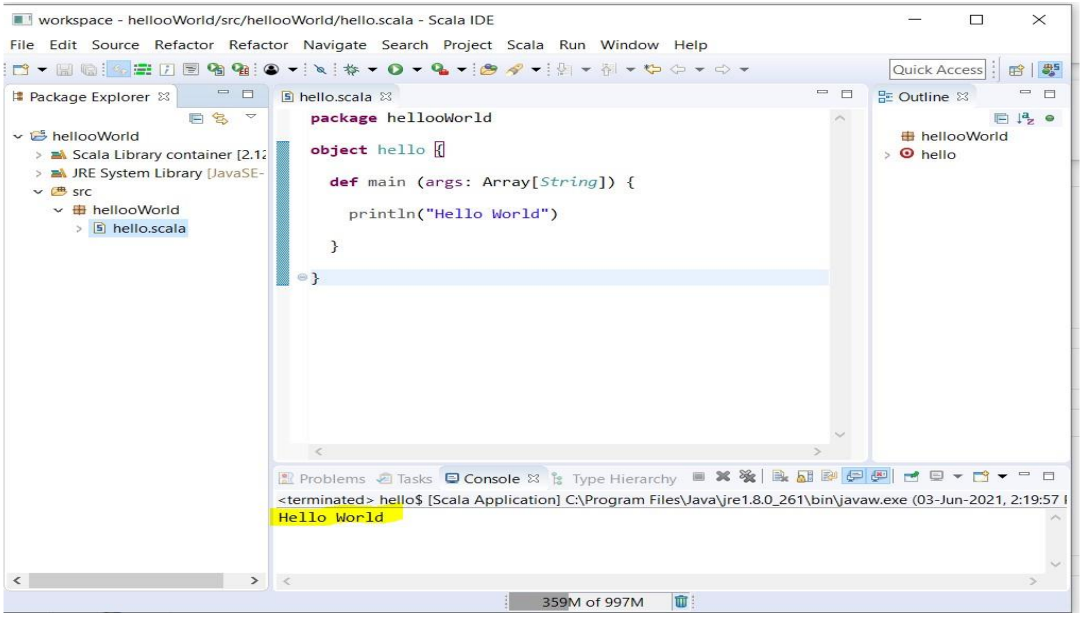
Program 9: Program to print word count on scala shell and print “Hello world” on scala IDE



```
      ^


scala> val data=sc.textFile("C:\\Spark\\spark-2.4.8-bin-hadoop2.7\\bin\\testdata\\sparkdata.txt")
data: org.apache.spark.rdd.RDD[String] = C:\Spark\spark-2.4.8-bin-hadoop2.7\bin\testdata\sparkdata.txt MapPartitionsRDD[61] at textFile at <con
sole>:24

scala> data.collect;
res31: Array[String] = Array(hi how are you?, how is your sister?, how is your jib?, how have you been?, "", "", "", "")

scala> val splitdata = data.flatMap(line => line.split(" "));
splitdata: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[62] at flatMap at <console>:25

scala> splitdata.collect;
res32: Array[String] = Array(hi, how, are, you?, how, is, your, sister?, how, is, your, jib?, how, have, you, been?, "", "", "", "")
```



```
scala> val mapdata = splitdata.map(word => (word,1));
mapdata: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[63] at map at <console>:25

scala> mapdata.collect;
res33: Array[(String, Int)] = Array((hi,1), (how,1), (are,1), (you?,1), (how,1), (is,1), (your,1), (sister?,1), (how,1), (is,1), (your,1), (jib
?,1), (how,1), (have,1), (you,1), (been?,1), ("",1), ("",1), ("",1), ("",1))

scala> val reducedata = mapdata.reduceByKey(_+_);
reducedata: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[64] at reduceByKey at <console>:25

scala> reducedata.collect;
res34: Array[(String, Int)] = Array((are,1), (is,2), (jib?,1), (have,1), (how,4), (you?,1), ("",4), (sister?,1), (you,1), (hi,1), (been?,1), (y
our,2))

scala>
```

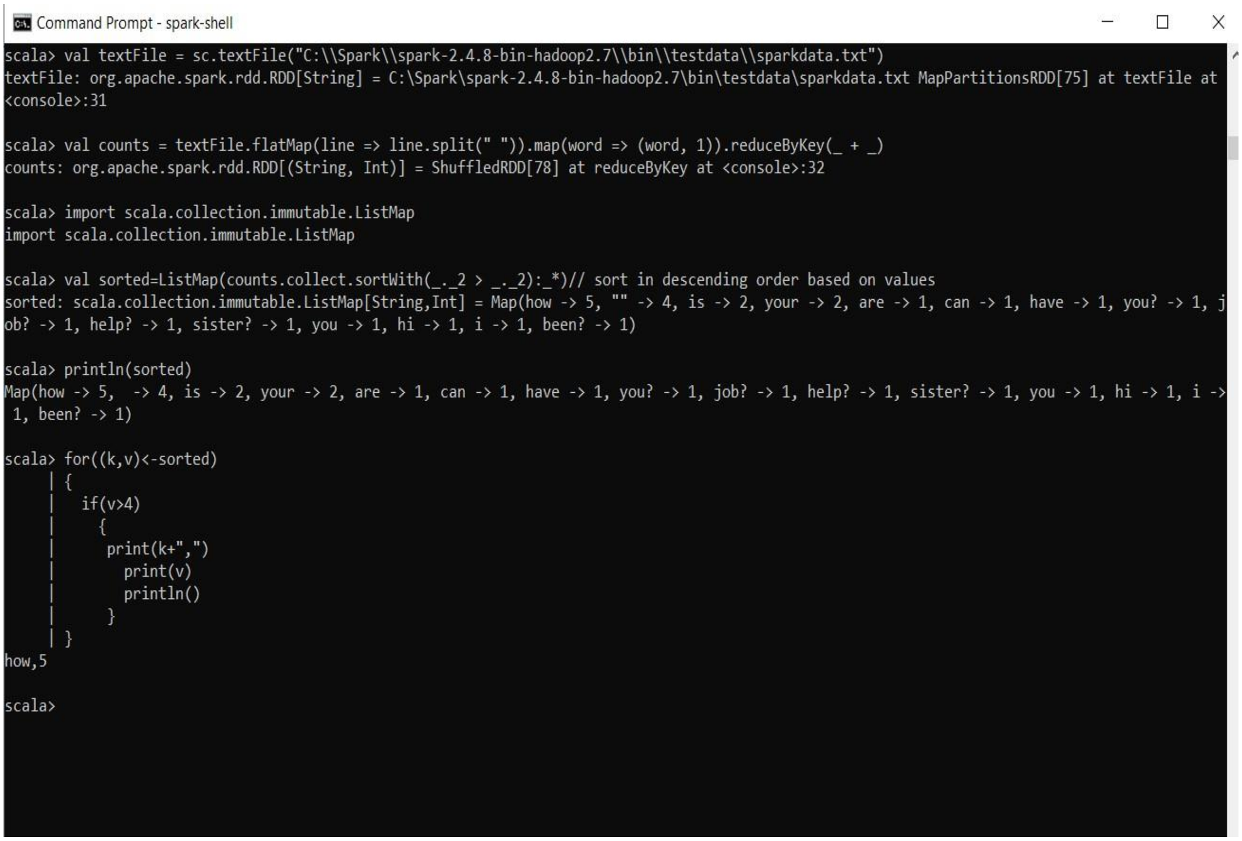

workspace - hellooWorld/src/hellooWorld/hello.scala - Scala IDE
File Edit Source Refactor Refactor Navigate Search Project Scala Run Window Help
Quick Access
Package Explorer
hellooWorld
Scala Library container [2.12
JRE System Library [JavaSE-
src
hellooWorld
hello.scala
Outline
hellooWorld
hello
Problems Tasks Console Type Hierarchy
359M of 997M

hello.scala

```
package hellooWorld

object hello {

  def main (args: Array[String]) {

    println("Hello World")

  }

}
```

```
<terminated> hello$ [Scala Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (03-Jun-2021, 2:19:57 I
Hello World
```

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



```
scala> val textFile = sc.textFile("C:\\Spark\\spark-2.4.8-bin-hadoop2.7\\bin\\testdata\\sparkdata.txt")
textFile: org.apache.spark.rdd.RDD[String] = C:\Spark\spark-2.4.8-bin-hadoop2.7\bin\testdata\sparkdata.txt MapPartitionsRDD[75] at textFile at
<console>:31

scala> val counts = textFile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(_ + _)
counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[78] at reduceByKey at <console>:32

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

scala> val sorted=ListMap(counts.collect.sortWith(_._2 > _._2):_*)// sort in descending order based on values
sorted: scala.collection.immutable.ListMap[String,Int] = Map(how -> 5, "" -> 4, is -> 2, your -> 2, are -> 1, can -> 1, have -> 1, you? -> 1, j
ob? -> 1, help? -> 1, sister? -> 1, you -> 1, hi -> 1, i -> 1, been? -> 1)

scala> println(sorted)
Map(how -> 5,  -> 4, is -> 2, your -> 2, are -> 1, can -> 1, have -> 1, you? -> 1, job? -> 1, help? -> 1, sister? -> 1, you -> 1, hi -> 1, i ->
 1, been? -> 1)

scala> for((k,v)<-sorted)
     | {
     |   if(v>4)
     |     {
     |      print(k+",")
     |        print(v)
     |        println()
     |      }
     | }
how,5

scala>
```