VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Big Data Analytics (23CS6PCBDA)

Submitted by

Vagisha Ajay (1BM22CS346)

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
Feb-2025 to June-2025

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 (23CS6PCBDA)**" carried out by **Vagisha Ajay (1BM22CS346),** 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 2025. The Lab report has been approved as it satisfies the academic requirements in respect of a **Big Data Analytics - (23CS6PCBDA)** work prescribed for the said degree.

Spoorthi D M
Assistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. Kavitha Sooda
Professor and Head
Department of CSE
BMSCE, Bengaluru

Index Sheet

10	Write a simple streaming program in Spark to receive text data streams on a particular port, perform basic text cleaning (like white space removal, stop words removal, lemmatization, etc.), and print the cleaned text on the screen. (Open Ended Question).	41-43

Course Outcome

CO1	Apply the concept of NoSQL, Hadoop or Spark for a given task
CO2	Analyze big data analytics mechanisms that can be applied to obtain solution for a given problem.
CO3	Design and implement solutions using data analytics mechanisms for a given problem.

Lab 1

Q) MongoDB- CRUD Operations Demonstration (Practice and Self Study)

Code & Output:

1. **Create a database "Student" with the following attributes Rollno, Name , Age, ContactNo, Email-Id, grade, hobby:**

use Students;

2. **Insert 5 appropriate values according to the below queries.**

db.students.insertMany([

{ "Rollno": 10, "Name": "John", "Age": 20, "ContactNo": "1234567890", "Email-Id": "john@example.com", "grade": "A", "hobby": "Reading" },

{ "Rollno": 11, "Name": "Alice", "Age": 21, "ContactNo": "9876543210", "Email-Id": "alice@example.com", "grade": "B", "hobby": "Painting" },

{ "Rollno": 12, "Name": "Bob", "Age": 22, "ContactNo": "2345678901", "Email-Id": "bob@example.com", "grade": "C", "hobby": "Cooking" },

{ "Rollno": 13, "Name": "Eve", "Age": 23, "ContactNo": "3456789012", "Email-Id": "eve@example.com", "grade": "A" },

{ "Rollno": 14, "Name": "Charlie", "Age": 24, "ContactNo": "4567890123", "Email-Id": "charlie@example.com", "hobby": "Gardening" }

```
Atlas atlas-wanmtx-shard-0 [primary] Student> use Students
switched to db Students
Atlas atlas-wanmtx-shard-0 [primary] Students> show collections

Atlas atlas-wanmtx-shard-0 [primary] Students> db.students.insertMany([
...     { "Rollno": 10, "Name": "John", "Age": 20, "ContactNo": "1234567890", "Email-Id":
"john@example.com", "grade": "A", "hobby": "Reading" },
...     { "Rollno": 11, "Name": "Alice", "Age": 21, "ContactNo": "9876543210", "Email-Id":
 "alice@example.com", "grade":
 "B", "hobby": "Painting" },
...     { "Rollno": 12, "Name": "Bob", "Age": 22, "ContactNo": "2345678901", "Email-Id": "
bob@example.com", "grade": "C", "hobby": "Cooking" },
...     { "Rollno": 13, "Name": "Eve", "Age": 23, "ContactNo": "3456789012", "Email-Id": "
eve@example.com", "grade": "A"
 },
...     { "Rollno": 14, "Name": "Charlie", "Age": 24, "ContactNo": "4567890123", "Email-Id
": "charlie@example.com", "hobby": "Gardening" }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("661ce9dc76a00ff8cc51dae1"),
    '1': ObjectId("661ce9dc76a00ff8cc51dae2"),
    '2': ObjectId("661ce9dc76a00ff8cc51dae3"),
    '3': ObjectId("661ce9dc76a00ff8cc51dae4"),
    '4': ObjectId("661ce9dc76a00ff8cc51dae5")
  }
}
```

])

3. Write query to update Email-Id of a student with rollno 10.

db.students.updateOne(

{ "Rollno": 10 },

{ $set: { "Email-Id": "john.doe@example.com" } }

)

```
Atlas atlas-wanmtx-shard-0 [primary] Students> db.students.updateOne(
...     { "Rollno": 10 },
...     { $set: { "Email-Id": "john.doe@example.com" } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

4. Replace the student name from “Alice” to “Alicee” of rollno 11

db.students.updateOne(

{ "Rollno": 11 },

{ $set: { "Name": "Alicee" } }

)

```
Atlas atlas-wanmtx-shard-0 [primary] Students> db.students.updateOne(
...     { "Rollno": 11 },
...     { $set: { "Name": "Alicee" } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

5. Display Student Name and grade(Add if grade is not present)where the _id column is 1.

db.students.find({}, { "Name": 1, "grade": { $ifNull: ["$grade", "Not available"] }, "_id": 0 })

```
Atlas atlas-wanmtx-shard-0 [primary] Students> db.students.find({}, { "Name": 1, "grade":
{ $ifNull: ["$grade", "Not available"] }, "_id": 0 })
[
  { Name: 'John', grade: 'A' },
  { Name: 'Alicee', grade: 'B' },
  { Name: 'Bob', grade: 'C' },
  { Name: 'Eve', grade: 'A' },
  { Name: 'Charlie', grade: 'Not available' }
]
```

6. Update to add hobbies

db.students.updateMany(

{ "Name": "Eve" },

{ $set: { "hobby": "Dancing" } }

)

```
Atlas atlas-wanmtx-shard-0 [primary] Students> db.students.updateMany(
...     { "Name": "Eve" },
...     { $set: { "hobby": "Dancing" } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

7. Find documents where hobbies is set neither to Chess nor to Skating

db.students.find({ "hobby": { $nin: ["Chess", "Skating"] } })

```
Atlas atlas-wanmtx-shard-0 [primary] Students> db.students.find({ "hobby": { $nin: ["Chess
", "Skating"] } })
[
  {
    _id: ObjectId("661ce9dc76a00ff8cc51dae1"),
    Rollno: 10,
    Name: 'John',
    Age: 20,
    ContactNo: '1234567890',
    'Email-Id': 'john.doe@example.com',
    grade: 'A',
    hobby: 'Reading'
  },
  {
    _id: ObjectId("661ce9dc76a00ff8cc51dae2"),
    Rollno: 11,
    Name: 'Alicee',
    Age: 21,
    ContactNo: '9876543210',
    'Email-Id': 'alice@example.com',
    grade: 'B',
    hobby: 'Painting'
  },
  {
    _id: ObjectId("661ce9dc76a00ff8cc51dae3"),
    Rollno: 12,
    Name: 'Bob',
    Age: 22,
    ContactNo: '2345678901',
    'Email-Id': 'bob@example.com',
    grade: 'C',
    hobby: 'Cooking'
  },
```

8. Find documents whose name begins with A

db.students.find({ "Name": /^A/ })

```
Atlas atlas-wanmtx-shard-0 [primary] Students> db.students.find({ "Name": /^A/ })
[
  {
    _id: ObjectId("661ce9dc76a00ff8cc51dae2"),
    Rollno: 11,
    Name: 'Alicee',
    Age: 21,
    ContactNo: '9876543210',
    'Email-Id': 'alice@example.com',
    grade: 'B',
    hobby: 'Painting'
  }
]
```

Lab 2

Q) Perform the following DB operations using Cassandra

a) Create a keyspace by name **Employee**

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

c) Insert the values into the table in **batch**

d) Update Employee name and Department of **Emp-Id 121**

e) Sort the details of Employee records based on **salary**

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

g) Update the altered table to **add project names**

h) Create a **TTL of 15 seconds** to display the values of Employees

Code & Output:

```
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC: $ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> create keyspace Employee with replication = {'class':'SimpleStrategy;,;replicationfactor':1};
SyntaxException: line 1:89 mismatched input ':' expecting '}' (...with replication = {'class':'SimpleStrategy;,;replicationfactor'[:]1...)
cqlsh> create keyspace Employee WITH replication={'class':'SimpleStrategy','replicationfactor':1};
ConfigurationException: Unrecognized strategy option {replicationfactor} passed to SimpleStrategy for keyspace employee
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> CREATE TABLE IF NOT EXISTS Employee_Info(
   ... Emp_Id INT PRIMARY KEY,
   ... Emp_name TEXT,
   ... designation TEXT,
   ... date_of_joining DATE,
   ... Salary FLOAT,
   ... Dep_name TEXT,
   ... Projects SET<TEXT>);
InvalidRequest: Error from server: code=2200 [Invalid query] message="No keyspace has been specified. USE a keyspace, or explicitly specify keyspace.tablename"
cqlsh> USE eMPLOYEE
   ...
cqlsh> USE Employee
   ...
cqlsh> USE Employee;
cqlsh:employee> CREATE TABLE IF NOT EXISTS Employee_Info( Emp_Id INT PRIMARY KEY, Emp_name TEXT, designation TEXT, date_of_joining DATE, Salary FLOAT, Dep_name TEXT, Projects SET<TEXT>);
cqlsh:employee> describe keyspace Employee

CREATE KEYSPACE employee WITH replication = {'class': 'SimpleStrategy', 'replication_factor': '1'}  AND durable_writes = true;

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

```
cqlsh:employee> update employee_info using ttl 15 set salary = 0 where emp_id = 121;
cqlsh:employee> select * from employee_info;

 emp_id | bonus | date_of_joining | dep_name    | designation | emp_name    | projects                    | salary
--------+-------+-----------------+-------------+-------------+-------------+-----------------------------+---------
    120 | 12000 |      2024-05-06 | Engineering |   Developer | Priyanka GH |  {'Project B', 'ProjectA'} |   1e+06
    123 |  null |      2024-05-07 | Engineering |    Engineer |     Sadhana | {'Project M', 'Project P'} | 1.2e+06
    122 |  null |      2024-05-06 |  Management |          HR |     Rachana | {'Project C', 'Project M'} |   9e+05
    121 | 11000 |      2024-05-06 |  Management |   Developer |      Shreya |  {'Project C', 'ProjectA'} |         0

(4 rows)
cqlsh:employee> select * from employee_info;

 emp_id | bonus | date_of_joining | dep_name    | designation | emp_name    | projects                    | salary
--------+-------+-----------------+-------------+-------------+-------------+-----------------------------+---------
    120 | 12000 |      2024-05-06 | Engineering |   Developer | Priyanka GH |  {'Project B', 'ProjectA'} |   1e+06
    123 |  null |      2024-05-07 | Engineering |    Engineer |     Sadhana | {'Project M', 'Project P'} | 1.2e+06
    122 |  null |      2024-05-06 |  Management |          HR |     Rachana | {'Project C', 'Project M'} |   9e+05
    121 | 11000 |      2024-05-06 |  Management |   Developer |      Shreya |  {'Project C', 'ProjectA'} |    null

(4 rows)
cqlsh:employee>
```

```
    AND speculative_retry = '99p';
cqlsh:employee> select * from employee_info;

 emp_id | date_of_joining | dep_name    | designation | emp_name | projects                    | salary
--------+-----------------+-------------+-------------+----------+-----------------------------+---------
    120 |      2024-05-06 | Engineering |   Developer | Priyanka |  {'Project B', 'ProjectA'} |   1e+06
    123 |      2024-05-07 | Engineering |    Engineer |  Sadhana | {'Project M', 'Project P'} | 1.2e+06
    122 |      2024-05-06 |  Management |          HR |  Rachana | {'Project C', 'Project M'} |   9e+05
    121 |      2024-05-06 |  Management |   Developer |   Shreya |  {'Project C', 'ProjectA'} |   9e+05

(4 rows)
cqlsh:employee> update employee_info set emp_name = 'Priyanka GH' Where emp_id = '120';
InvalidRequest: Error from server: code=2200 [Invalid query] message="Invalid STRING constant (120) for "emp_id" of type int"
cqlsh:employee> update employee_info set emp_name = 'Priyanka GH' Where emp_id=120;
cqlsh:employee> select * from employee_info;

 emp_id | date_of_joining | dep_name    | designation | emp_name    | projects                    | salary
--------+-----------------+-------------+-------------+-------------+-----------------------------+---------
    120 |      2024-05-06 | Engineering |   Developer | Priyanka GH |  {'Project B', 'ProjectA'} |   1e+06
    123 |      2024-05-07 | Engineering |    Engineer |     Sadhana | {'Project M', 'Project P'} | 1.2e+06
    122 |      2024-05-06 |  Management |          HR |     Rachana | {'Project C', 'Project M'} |   9e+05
    121 |      2024-05-06 |  Management |   Developer |      Shreya |  {'Project C', 'ProjectA'} |   9e+05

(4 rows)
cqlsh:employee> select * from employee_info order by salary;
InvalidRequest: Error from server: code=2200 [Invalid query] message="ORDER BY is only supported when the partition key is restricted by an EQ or an IN."
cqlsh:employee> alter table employee_info add bonus INT;
cqlsh:employee> select * from employee_info;

 emp_id | bonus | date_of_joining | dep_name    | designation | emp_name    | projects                    | salary
--------+-------+-----------------+-------------+-------------+-------------+-----------------------------+---------
    120 |  null |      2024-05-06 | Engineering |   Developer | Priyanka GH |  {'Project B', 'ProjectA'} |   1e+06
    123 |  null |      2024-05-07 | Engineering |    Engineer |     Sadhana | {'Project M', 'Project P'} | 1.2e+06
    122 |  null |      2024-05-06 |  Management |          HR |     Rachana | {'Project C', 'Project M'} |   9e+05
    121 |  null |      2024-05-06 |  Management |   Developer |      Shreya |  {'Project C', 'ProjectA'} |   9e+05

(4 rows)
cqlsh:employee> update employee_info set bonus = 12000 where emp_id = 120;
cqlsh:employee> select * from employee_info;

 emp_id | bonus | date_of_joining | dep_name    | designation | emp_name    | projects                    | salary
--------+-------+-----------------+-------------+-------------+-------------+-----------------------------+---------
    120 | 12000 |      2024-05-06 | Engineering |   Developer | Priyanka GH |  {'Project B', 'ProjectA'} |   1e+06
    123 |  null |      2024-05-07 | Engineering |    Engineer |     Sadhana | {'Project M', 'Project P'} | 1.2e+06
    122 |  null |      2024-05-06 |  Management |          HR |     Rachana | {'Project C', 'Project M'} |   9e+05
    121 |  null |      2024-05-06 |  Management |   Developer |      Shreya |  {'Project C', 'ProjectA'} |   9e+05

(4 rows)
cqlsh:employee> update employee_info set bonus = 11000 where emp_id = 121;
cqlsh:employee> select * from employee_info using ttl 15 where emp_id = 123;
SyntaxException: line 1:28 mismatched input 'using' expecting EOF (select * from employee_info [using] ttl...)
cqlsh:employee> select * from employee_info where emp_id = 121 using ttl 15;
SyntaxException: line 1:47 no viable alternative at input 'using' (...employee_info where emp_id = 121 [using]...)
cqlsh:employee> update employee_info using ttl 15 set salary = 0 where emp_id = 121;
cqlsh:employee> select * from employee_info;
```

Lab 3

Q) Perform the following DB operations using Cassandra

a) Create a keyspace by name **Library**

b) 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

c) Insert the values into the table in **batch**

d) Display the details of the table created and **increase the value of the counter**

e) Write a query to show that a student with **id 112** has taken a book **"BDA" 2 times**

f) **Export** the created column to a **CSV file**

g) **Import** a given CSV dataset from **local file system** into Cassandra **column family**

Code & Output:

```
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> CREATE KEYSPACE Students WITH REPLICATION={
   ... 'class':'SimpleStrategy','replication_factor':1};
cqlsh> DESCRIBE KEYSPACES

students  system_auth         system_schema  system_views
system    system_distributed  system_traces  system_virtual_schema

cqlsh> SELECT * FROM system.schema_keyspaces;
InvalidRequest: Error from server: code=2200 [Invalid query] message="table schema_keyspaces does not exist"
cqlsh> use Students;
cqlsh:students> create table Students_info(Roll_No int Primary key,StudName text,DateOfJoining timestamp,last_exam_Percent double);
cqlsh:students> describe tables;

students_info

cqlsh:students> describe table students;
Table 'students' not found in keyspace 'students'
cqlsh:students> describe table students_info;

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

```
cqlsh:students> Begin batch insert into Students_info(Roll_no, StudName,DateOfJoining, last_exam_Percent) values(1,'Sadhana','2023-10-09', 98) insert into Students_info(Roll_no, StudName,DateOfJoining, last_exam
_Percent) values(2,'Rutu','2023-10-10', 97) insert into Students_info(Roll_no, StudName,DateOfJoining, last_exam_Percent) values(3,'Rachana','2023-10-10', 97.5) insert into Students_info(Roll_no, StudName,DateOf
Joining, last_exam_Percent) values(4,'Charu','2023-10-06', 96.5) apply batch;
cqlsh:students> select * from students_info;

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       1 | 2023-10-08 18:30:00.000000+0000 |                98 |  Sadhana
       2 | 2023-10-09 18:30:00.000000+0000 |                97 |     Rutu
       4 | 2023-10-05 18:30:00.000000+0000 |              96.5 |    Charu
       3 | 2023-10-09 18:30:00.000000+0000 |              97.5 |  Rachana

(4 rows)
cqlsh:students> select * from students_info where roll_no in (1,2,3);

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       1 | 2023-10-08 18:30:00.000000+0000 |                98 |  Sadhana
       2 | 2023-10-09 18:30:00.000000+0000 |                97 |     Rutu
       3 | 2023-10-09 18:30:00.000000+0000 |              97.5 |  Rachana

(3 rows)
cqlsh:students> select * from students_info where Studname='Charu';
InvalidRequest: Error from server: code=2200 [Invalid query] message="Cannot execute this query as it might involve data filtering and thus may have unpredictable performance. If you want to execute this query d
espite the performance unpredictability, use ALLOW FILTERING"
cqlsh:students> create index on Students_info(StudName);
cqlsh:students> select * from students_info where Studname='Charu';

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       4 | 2023-10-05 18:30:00.000000+0000 |              96.5 |    Charu

(1 rows)
cqlsh:students> select Roll_no,StudName from students_info LIMIT 2;
```

```
(4 rows)
cqlsh:students> select * from students_info where roll_no in (1,2,3);

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       1 | 2023-10-08 18:30:00.000000+0000 |                98 |  Sadhana
       2 | 2023-10-09 18:30:00.000000+0000 |                97 |     Rutu
       3 | 2023-10-09 18:30:00.000000+0000 |              97.5 |  Rachana

(3 rows)
cqlsh:students> select * from students_info where Studname='Charu';
InvalidRequest: Error from server: code=2200 [Invalid query] message="Cannot execute this query as it might involve data filtering and thus may have unpredictable performance. If you want to execute this query d
espite the performance unpredictability, use ALLOW FILTERING"
cqlsh:students> create index on Students_info(StudName);
cqlsh:students> select * from students_info where Studname='Charu';

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       4 | 2023-10-05 18:30:00.000000+0000 |              96.5 |    Charu

(1 rows)
cqlsh:students> select Roll_no,StudName from students_info LIMIT 2;

 roll_no | studname
---------+----------
       1 |  Sadhana
       2 |     Rutu

(2 rows)
cqlsh:students> SELECT Roll_no as "USN" from Students_info;

 USN
-----
   1
   2
   4
   3
```

Lab 4

Q) Execution of HDFS Commands for interaction with Hadoop Environment. (Minimum 10 commands to be executed)

Code & Output:

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cd ./Desktop/
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [bmscecse-HP-Elite-Tower-800-G9-Desktop-PC]
Starting resourcemanager
Starting nodemanagers
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -mkdir /Lab05
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /Hadoop
ls: `/Hadoop': No such file or directory
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /Lab05
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ touch test.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ nano text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -put ./text.txt /Lab05/text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /Lab05
Found 1 items
-rw-r--r--   1 hadoop supergroup         19 2024-05-13 14:33 /Lab05/text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -cat /Lab05/text.txt
Hello
How are you?
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /Lab05
Found 2 items
-rw-r--r--   1 hadoop supergroup         15 2024-05-13 14:40 /Lab05/test.txt
-rw-r--r--   1 hadoop supergroup         19 2024-05-13 14:33 /Lab05/text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -getmerge /Lab05 /text.txt /Lab05 /test.txt ../
Downloads/Merged.txt
getmerge: `/text.txt': No such file or directory
getmerge: `/test.txt': No such file or directory
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -getmerge /Lab05/text.txt /Lab05/test.txt ../Do
wnloads/Merged.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -getfacl /Lab05
# file: /Lab05
# owner: hadoop
# group: supergroup
user::rwx
group::r-x
other::r-x
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -copyToLocal /Lab05/text.txt ../Documents
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -copyToLocal /Lab05/test.txt ../Documents
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -cat /Lab05/text.txt
Hello
How are you?
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -mv /Lab05 /test_Lab05
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -ls /test_Lab05
Found 2 items
-rw-r--r--   1 hadoop supergroup         15 2024-05-13 14:40 /test_Lab05/test.txt
-rw-r--r--   1 hadoop supergroup         19 2024-05-13 14:33 /test_Lab05/text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -cp /test_Lab05/ /Lab05
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -ls /Lab05
Found 2 items
-rw-r--r--   1 hadoop supergroup         15 2024-05-13 14:51 /Lab05/test.txt
-rw-r--r--   1 hadoop supergroup         19 2024-05-13 14:51 /Lab05/text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -ls /test_Lab05
Found 2 items
-rw-r--r--   1 hadoop supergroup         15 2024-05-13 14:40 /test_Lab05/test.txt
-rw-r--r--   1 hadoop supergroup         19 2024-05-13 14:33 /test_Lab05/text.txt
```

Lab 5

Q) Implement Wordcount program on Hadoop framework

Code :

```
//Driver Code
// Importing libraries
import java.io.IOException;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;

public class WCDriver extends Configured implements Tool {

    public int run(String[] args) throws IOException {
        if (args.length < 2) {
            System.out.println("Please give valid inputs");
            return -1;
        }

        JobConf conf = new JobConf(WCDriver.class);
        conf.setJobName("WordCount");

        FileInputFormat.setInputPaths(conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));
```

```
        conf.setMapperClass(WCMapper.class);
        conf.setReducerClass(WCReducer.class);

        conf.setMapOutputKeyClass(Text.class);
        conf.setMapOutputValueClass(IntWritable.class);

        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(IntWritable.class);

        JobClient.runJob(conf);
        return 0;
    }

    // Main Method
    public static void main(String[] args) throws Exception {
        int exitCode = ToolRunner.run(new WCDriver(), args);
        System.out.println("Job Exit Code: " + exitCode);

//Mapper Code
// Importing libraries
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.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;

import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;
```

```
public class WCMapper extends MapReduceBase implements Mapper<LongWritable, Text, Text,
IntWritable> {


    // Map function

    public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter
reporter)

            throws IOException {

        String line = value.toString();


        // Splitting the line on whitespace

        for (String word : line.split("\\s+")) {

            if (word.length() > 0) {

                output.collect(new Text(word), new IntWritable(1));

            }

        }

    }

}



//Reducer Code

// Importing libraries

import java.io.IOException;

import java.util.Iterator;


import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.MapReduceBase;

import org.apache.hadoop.mapred.OutputCollector;

import org.apache.hadoop.mapred.Reducer;

import org.apache.hadoop.mapred.Reporter;
```

```
public class WCReducer extends MapReduceBase implements Reducer<Text, IntWritable, Text,
IntWritable> {

    // Reduce function
    public void reduce(Text key, Iterator<IntWritable> values,
                    OutputCollector<Text, IntWritable> output,
                    Reporter reporter) throws IOException {
        int count = 0;

        // Counting the frequency of each word
        while (values.hasNext()) {
            count += values.next().get();
        }

        output.collect(key, new IntWritable(count));
    }
}
```

Input File -> hi how are you
how is your brother
how is your sister
how is your family

Output:

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cd ./Desktop/
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [bmscecse-HP-Elite-Tower-800-G9-Desktop-PC]
Starting resourcemanager
Starting nodemanagers
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -mkdir /Lab06
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /Lab06
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ jps
7360 DataNode
7928 ResourceManager
8681 Jps
7178 NameNode
8091 NodeManager
7644 SecondaryNameNode
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ cd ..
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cd ./Desktop/
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ nano file1.txt
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -copyFromLocal -f /home/hadoop/Desktop/file1.txt /rgs/test.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop jar /home/hadoop/Desktop/WordCount.jar wordcount.WordCount /rgs/test.txt /output
JAR does not exist or is not a normal file: /home/hadoop/Desktop/WordCount.jar
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop jar /home/hadoop/Desktop/Word_Count.jar wordcount.WordCount /rgs/test.txt /output
Exception in thread "main" java.lang.ClassNotFoundException: wordcount.WordCount
        at java.base/java.net.URLClassLoader.findClass(URLClassLoader.java:476)
        at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:594)
        at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:527)
        at java.base/java.lang.Class.forName0(Native Method)
        at java.base/java.lang.Class.forName(Class.java:398)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:321)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:241)
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -cat /output/part-00000
are     1
brother 1
family  1
hi      1
how     5
is      4
job     1
sister  1
you     1
your    4
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /output
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2024-05-21 15:21 /output/_SUCCESS
-rw-r--r--   1 hadoop supergroup         69 2024-05-21 15:21 /output/part-00000
```

Lab 6

Q) From the following link extract the weather data
https://github.com/tomwhite/hadoopbook/tree/master/input/ncdc/all

Create a Map Reduce program to
a) find average temperature for each year from NCDC data set.
b) find the mean max temperature for every month.

Find average temperature for each year from NCDC data set

Code:

```
//Driver Code

package temp;


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.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 both input and output parameters.");

            System.exit(-1);

        }
```

```
        // Creating a configuration and job instance
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "Average Calculation");

        job.setJarByClass(AverageDriver.class);

        // Input and output paths
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        // Setting mapper and reducer classes
        job.setMapperClass(AverageMapper.class);
        job.setReducerClass(AverageReducer.class);

        // Output key and value types
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

        // Submitting the job and waiting for it to complete
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

```
//Mapper Code
package temp;

import java.io.IOException;
```

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

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

    public static final int MISSING = 9999;

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

        String line = value.toString();

        // Extract year from fixed position
        String year = line.substring(15, 19);
        int temperature;

        // Determine if there's a '+' sign
        if (line.charAt(87) == '+') {
            temperature = Integer.parseInt(line.substring(88, 92));
        } else {
            temperature = Integer.parseInt(line.substring(87, 92));
        }

        // Quality check character
        String quality = line.substring(92, 93);

        // Only emit if data is valid
```

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

//Reducer Code

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> {

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

        int sumTemp = 0;
        int count = 0;

        for (IntWritable value : values) {
            sumTemp += value.get();
            count++;
        }

        if (count > 0) {
            int average = sumTemp / count;
```

```
        context.write(key, new IntWritable(average));
    }
    }
}
```

Output:

```
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
hadoop@localhost's password:
localhost: namenode is running as process 7959.  Stop it first and ensure /tmp/hadoop-hadoop-namenode.pid file is empty before retry.
Starting datanodes
hadoop@localhost's password:
localhost: datanode is running as process 8216.  Stop it first and ensure /tmp/hadoop-hadoop-datanode.pid file is empty before retry.
Starting secondary namenodes [bmscecse-HP-Elite-Tower-600-G9-Desktop-PC]
hadoop@bmscecse-hp-elite-tower-600-g9-desktop-pc's password:
bmscecse-HP-Elite-Tower-600-G9-Desktop-PC: secondarynamenode is running as process 8476.  Stop it first and ensure /tmp/hadoop-hadoop-secondarynamenode.pid file is empty before retry.
Starting resourcemanager
resourcemanager is running as process 8759.  Stop it first and ensure /tmp/hadoop-hadoop-resourcemanager.pid file is empty before retry.
Starting nodemanagers
hadoop@localhost's password:
localhost: nodemanager is running as process 9123.  Stop it first and ensure /tmp/hadoop-hadoop-nodemanager.pid file is empty before retry.
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ jps
13698 org.eclipse.equinox.launcher_1.6.1000.v20250227-1734.jar
9123 NodeManager
20278 Jps
7959 NameNode
8759 ResourceManager
8216 DataNode
8476 SecondaryNameNode
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -mkdir /vag
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -copyFromLocal /home/hadoop/Desktop/sample1.txt /vag/test.txt
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop jar /home/hadoop/Desktop/Temp2.jar avg.AverageDriver /vag/test.txt /vagre
2025-05-20 15:14:12,780 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2025-05-20 15:14:12,820 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2025-05-20 15:14:12,820 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2025-05-20 15:14:12,879 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2025-05-20 15:14:12,917 INFO input.FileInputFormat: Total input files to process : 1
2025-05-20 15:14:12,945 INFO mapreduce.JobSubmitter: number of splits:1
2025-05-20 15:14:13,007 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local510595367_0001
2025-05-20 15:14:13,007 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-05-20 15:14:13,066 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2025-05-20 15:14:13,066 INFO mapreduce.Job: Running job: job_local510595367_0001
2025-05-20 15:14:13,066 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2025-05-20 15:14:13,070 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-05-20 15:14:13,070 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2025-05-20 15:14:13,070 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2025-05-20 15:14:13,114 INFO mapred.LocalJobRunner: Waiting for map tasks
2025-05-20 15:14:13,114 INFO mapred.LocalJobRunner: Starting task: attempt_local510595367_0001_m_000000_0
2025-05-20 15:14:13,124 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-05-20 15:14:13,124 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2025-05-20 15:14:13,131 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2025-05-20 15:14:13,132 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/vag/test.txt:0+530
2025-05-20 15:14:13,165 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2025-05-20 15:14:13,165 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2025-05-20 15:14:13,165 INFO mapred.MapTask: soft limit at 83886080
2025-05-20 15:14:13,165 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2025-05-20 15:14:13,165 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2025-05-20 15:14:13,167 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
2025-05-20 15:14:13,208 INFO mapred.LocalJobRunner:
```

```
2025-05-20 15:14:13,341 INFO mapred.LocalJobRunner: Finishing task: attempt
2025-05-20 15:14:13,341 INFO mapred.LocalJobRunner: reduce task executor co
2025-05-20 15:14:14,068 INFO mapreduce.Job: Job job_local510595367_0001 running in uber mode : false
2025-05-20 15:14:14,070 INFO mapreduce.Job:  map 100% reduce 100%
2025-05-20 15:14:14,072 INFO mapreduce.Job: Job job_local510595367_0001 completed successfully
2025-05-20 15:14:14,086 INFO mapreduce.Job: Counters: 36
        File System Counters
                FILE: Number of bytes read=10920
                FILE: Number of bytes written=1284205
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=1060
                HDFS: Number of bytes written=15
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=5
                Map output records=5
                Map output bytes=45
                Map output materialized bytes=61
                Input split bytes=99
                Combine input records=0
                Combine output records=0
                Reduce input groups=2
                Reduce shuffle bytes=61
                Reduce input records=5
                Reduce output records=2
                Spilled Records=10
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=0
                Total committed heap usage (bytes)=1052770304
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=530
        File Output Format Counters
                Bytes Written=15
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -ls /vagre
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2025-05-20 15:14 /vagre/_SUCCESS
-rw-r--r--   1 hadoop supergroup         15 2025-05-20 15:14 /vagre/part-r-00000
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -cat /vagre/part-r-00000
1949    94
1950    3
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$
```

b) Find the mean max temperature for every month

Code:

```
//Driver Code
package meanmax;

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.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 both input and output parameters.");
            System.exit(-1);
        }

        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "Mean and Max Temperature");

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

```
        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);
    }
}

//Mapper Code

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;

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

```
        String line = value.toString();
  // Extract month from positions 19-20
        String month = line.substring(19, 21);
        int temperature;

        // Extract temperature considering optional '+'
        if (line.charAt(87) == '+') {
            temperature = Integer.parseInt(line.substring(88, 92));
        } else {
            temperature = Integer.parseInt(line.substring(87, 92));
        }

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

        if (temperature != MISSING && quality.matches("[01459]")) {
            context.write(new Text(month), new IntWritable(temperature));
        }
    }
}

//Reducer Code
package meanmax;

import java.io.IOException;

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

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

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

        int sumTemp = 0;
        int count = 0;
        int maxTemp = Integer.MIN_VALUE;

        for (IntWritable value : values) {
            int temp = value.get();
            sumTemp += temp;
            count++;

            if (temp > maxTemp) {
                maxTemp = temp;
            }
        }

        if (count > 0) {
            int avgTemp = sumTemp / count;
            String result = "mean=" + avgTemp + " max=" + maxTemp;
            context.write(key, new Text(result));
        }
    }
}
```

Output:

```
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
hadoop@localhost's password:
localhost: namenode is running as process 7959.  Stop it first and ensure /tmp/hadoop-hadoop-namenode.pid file is empty before retry.
Starting datanodes
hadoop@localhost's password:
localhost: datanode is running as process 8216.  Stop it first and ensure /tmp/hadoop-hadoop-datanode.pid file is empty before retry.
Starting secondary namenodes [bmscecse-HP-Elite-Tower-600-G9-Desktop-PC]
hadoop@bmscecse-hp-elite-tower-600-g9-desktop-pc's password:
bmscecse-HP-Elite-Tower-600-G9-Desktop-PC: secondarynamenode is running as process 8476.  Stop it first and ensure /tmp/hadoop-hadoop-secondarynamenode.pid file is empty before retry.
Starting resourcemanager
resourcemanager is running as process 8759.  Stop it first and ensure /tmp/hadoop-hadoop-resourcemanager.pid file is empty before retry.
Starting nodemanagers
hadoop@localhost's password:
localhost: nodemanager is running as process 9123.  Stop it first and ensure /tmp/hadoop-hadoop-nodemanager.pid file is empty before retry.
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ jps
13698 org.eclipse.equinox.launcher_1.6.1000.v20250227-1734.jar
23346 Jps
9123 NodeManager
7959 NameNode
8759 ResourceManager
8216 DataNode
8476 SecondaryNameNode
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -mkdir /vag1
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -copyFromLocal /home/hadoop/Desktop/1901.txt /vag1/test.txt
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop jar /home/hadoop/Desktop/meanmax1.jar meanmax.mmDriver /vag1/test.txt /vag1re
2025-05-20 15:33:29,196 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2025-05-20 15:33:29,239 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2025-05-20 15:33:29,240 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2025-05-20 15:33:29,300 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2025-05-20 15:33:29,339 INFO input.FileInputFormat: Total input files to process : 1
2025-05-20 15:33:29,367 INFO mapreduce.JobSubmitter: number of splits:1
2025-05-20 15:33:29,430 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1952234974_0001
2025-05-20 15:33:29,430 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-05-20 15:33:29,491 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2025-05-20 15:33:29,491 INFO mapreduce.Job: Running job: job_local1952234974_0001
2025-05-20 15:33:29,492 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2025-05-20 15:33:29,496 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-05-20 15:33:29,496 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2025-05-20 15:33:29,496 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2025-05-20 15:33:29,546 INFO mapred.LocalJobRunner: Waiting for map tasks
2025-05-20 15:33:29,546 INFO mapred.LocalJobRunner: Starting task: attempt_local1952234974_0001_m_000000_0
2025-05-20 15:33:29,557 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-05-20 15:33:29,557 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2025-05-20 15:33:29,564 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2025-05-20 15:33:29,566 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/vag1/test.txt:0+888190
2025-05-20 15:33:29,599 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2025-05-20 15:33:29,599 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2025-05-20 15:33:29,599 INFO mapred.MapTask: soft limit at 83886080
2025-05-20 15:33:29,599 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2025-05-20 15:33:29,599 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2025-05-20 15:33:29,601 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
```

```
                HDFS: Number of bytes read=1776380
                HDFS: Number of bytes written=74
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=6565
                Map output records=6564
                Map output bytes=45948
                Map output materialized bytes=59082
                Input split bytes=100
                Combine input records=0
                Combine output records=0
                Reduce input groups=12
                Reduce shuffle bytes=59082
                Reduce input records=6564
                Reduce output records=12
                Spilled Records=13128
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=3
                Total committed heap usage (bytes)=1052770304
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=888190
        File Output Format Counters
                Bytes Written=74
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -ls /vag1re
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2025-05-20 15:33 /vag1re/_SUCCESS
-rw-r--r--   1 hadoop supergroup         74 2025-05-20 15:33 /vag1re/part-r-00000
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -cat /vag1re/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
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$
```

Lab 7

Q) For a given Text file, Create a Map Reduce program to sort the content in an alphabetic order listing only top 10 maximum occurrences of words.

Code:

```
//TopN

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 = 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);
            }
        }
    }
}


//TopNMapper

package samples.topn;
```

```
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

package samples.topn;

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));
        }
    }
}
```

```
//TopNCombiner

package samples.topn;

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));
    }
}


//utils.java
package utils;

import java.util.*;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
```

```
public class MiscUtils {
    public static Map<Text, IntWritable> sortByValues(Map<Text, IntWritable> map) {
        List<Map.Entry<Text, IntWritable>> list = new LinkedList<>(map.entrySet());

        // Sort the list in descending order of values
        Collections.sort(list, new Comparator<Map.Entry<Text, IntWritable>>() {
            public int compare(Map.Entry<Text, IntWritable> o1, Map.Entry<Text, IntWritable> o2) {
                return o2.getValue().compareTo(o1.getValue());
            }
        });

        // Maintain insertion order with LinkedHashMap
        Map<Text, IntWritable> sortedMap = new LinkedHashMap<>();
        for (Map.Entry<Text, IntWritable> entry : list) {
            sortedMap.put(entry.getKey(), entry.getValue());
        }
        return sortedMap;
    }
}
```

Output:

```
-rw-r--r--   1 hadoop supergroup         15 2025-05-20 15:14 /vagre/part-r-00000
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -cat /vagre/part-r-00000
1949    94
1950    3
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -mkdir /vag2
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -copyFromLocal /home/hadoop/Desktop/sample.txt /vag2/test.txt
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop jar /home/hadoop/Desktop/topn1.jar topn.TopN /vag2/test.txt /vag2re
2025-05-20 15:57:09,527 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2025-05-20 15:57:09,569 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2025-05-20 15:57:09,569 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2025-05-20 15:57:09,670 INFO input.FileInputFormat: Total input files to process : 1
2025-05-20 15:57:09,699 INFO mapreduce.JobSubmitter: number of splits:1
2025-05-20 15:57:09,763 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local867925332_0001
2025-05-20 15:57:09,763 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-05-20 15:57:09,823 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2025-05-20 15:57:09,823 INFO mapreduce.Job: Running job: job_local867925332_0001
2025-05-20 15:57:09,824 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2025-05-20 15:57:09,828 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-05-20 15:57:09,828 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2025-05-20 15:57:09,829 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2025-05-20 15:57:09,871 INFO mapred.LocalJobRunner: Waiting for map tasks
2025-05-20 15:57:09,872 INFO mapred.LocalJobRunner: Starting task: attempt_local867925332_0001_m_000000_0
2025-05-20 15:57:09,882 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-05-20 15:57:09,882 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2025-05-20 15:57:09,889 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2025-05-20 15:57:09,890 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/vag2/test.txt:0+73
2025-05-20 15:57:09,923 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2025-05-20 15:57:09,923 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2025-05-20 15:57:09,923 INFO mapred.MapTask: soft limit at 83886080
2025-05-20 15:57:09,923 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2025-05-20 15:57:09,923 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2025-05-20 15:57:09,925 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
2025-05-20 15:57:09,964 INFO mapred.LocalJobRunner:
2025-05-20 15:57:09,965 INFO mapred.MapTask: Starting flush of map output
2025-05-20 15:57:09,965 INFO mapred.MapTask: Spilling map output
2025-05-20 15:57:09,965 INFO mapred.MapTask: bufstart = 0; bufend = 137; bufvoid = 104857600
2025-05-20 15:57:09,965 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 26214336(104857344); length = 61/6553600
2025-05-20 15:57:09,968 INFO mapred.MapTask: Finished spill 0
2025-05-20 15:57:09,973 INFO mapred.Task: Task:attempt_local867925332_0001_m_000000_0 is done. And is in the process of committing
2025-05-20 15:57:09,975 INFO mapred.LocalJobRunner: map
2025-05-20 15:57:09,975 INFO mapred.Task: Task 'attempt_local867925332_0001_m_000000_0' done.
2025-05-20 15:57:09,978 INFO mapred.Task: Final Counters for attempt_local867925332_0001_m_000000_0: Counters: 23
        File System Counters
                FILE: Number of bytes read=9900
                FILE: Number of bytes written=647132
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=73
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=5
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=1
```

```
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=146
                HDFS: Number of bytes written=63
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=4
                Map output records=16
                Map output bytes=137
                Map output materialized bytes=175
                Input split bytes=100
                Combine input records=0
                Combine output records=0
                Reduce input groups=9
                Reduce shuffle bytes=175
                Reduce input records=16
                Reduce output records=9
                Spilled Records=32
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=0
                Total committed heap usage (bytes)=1052770304
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=73
        File Output Format Counters
                Bytes Written=63
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -ls /vag2re
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2025-05-20 15:57 /vag2re/_SUCCESS
-rw-r--r--   1 hadoop supergroup         63 2025-05-20 15:57 /vag2re/part-r-00000
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -cat /vag2re/part-r-00000
how     4
your    3
is      3
brother 1
are     1
hi      1
sister  1
family  1
you     1
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$
```

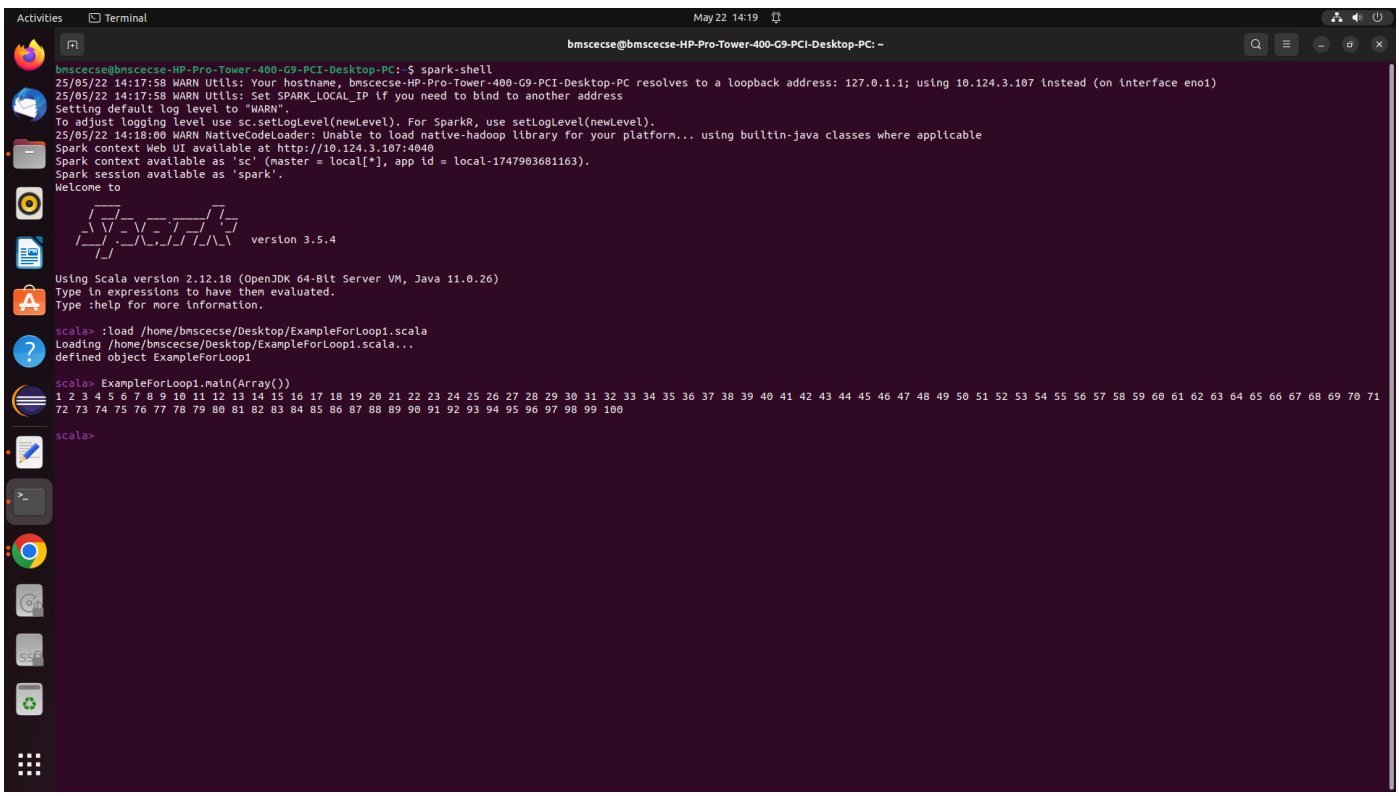
Lab 8

Q) Write a Scala program to print numbers from 1 to 100 using for loop.

Code:

```
object ExampleForLoop1 {
  def main(args: Array[String]): Unit = {
    for (counter <- 1 to 100)
      print(counter + " ")
    // to print new line
    println()
  }
}
```

Output:



Lab 9

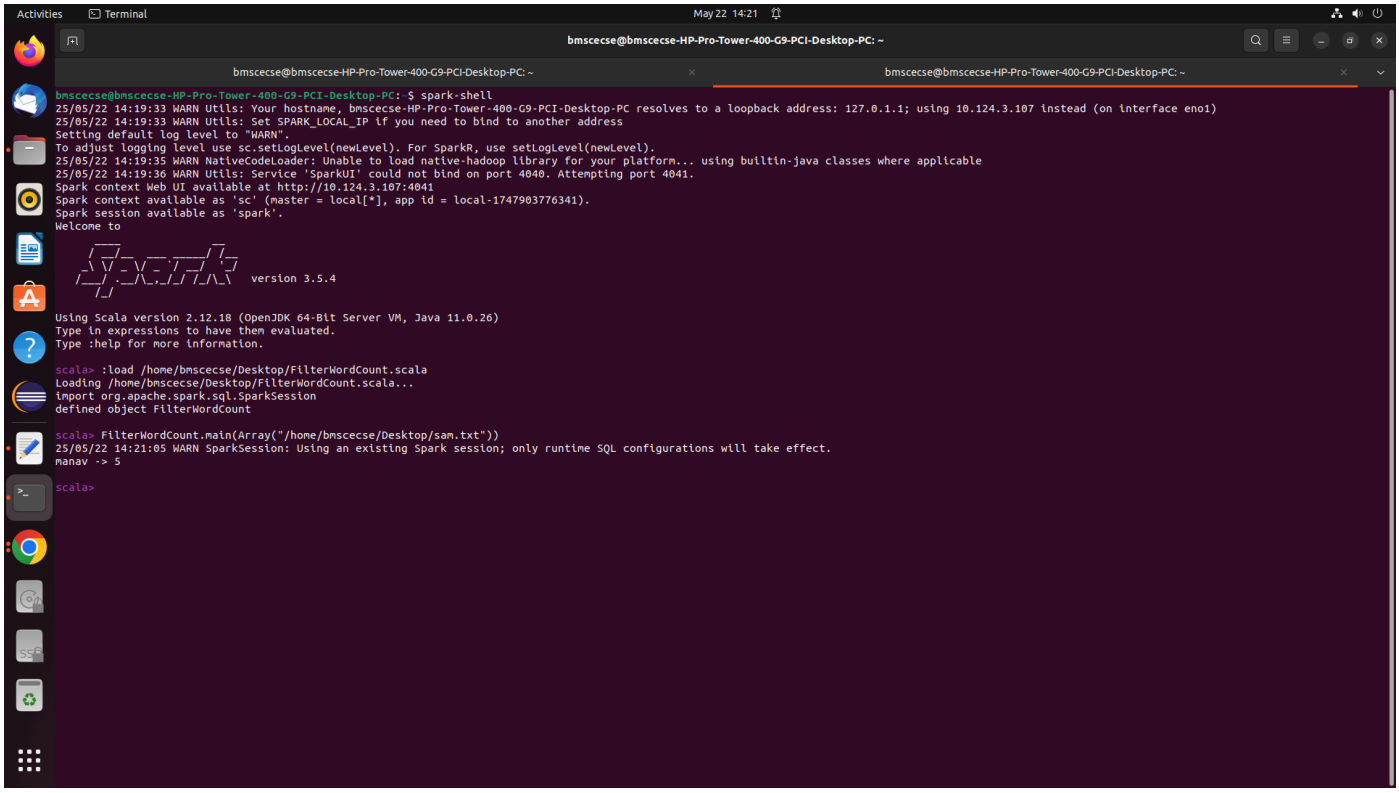
Q) Using RDD and FlatMap count how many times each word appears in a file and write out a list of words whose count is strictly greater than 4 using Spark.

Code:

```
import org.apache.spark.sql.SparkSession


object FilterWordCount {
  def main(args: Array[String]): Unit = {
    if (args.length < 1) {
      System.err.println("Usage: FilterWordCount <file>")
      System.exit(1)
    }

    // Create a SparkSession, which internally manages the SparkContext
    val spark = SparkSession.builder()
      .appName("FilterWordCount")
      .master("local[*]")  // Use local[*] for running on local machine with multiple cores
      .getOrCreate()

    // Use the SparkSession's SparkContext to read the file
    val rdd = spark.sparkContext.textFile(args(0))

    val counts = rdd
      .flatMap(_.split("\\s+"))
      .map(_.replaceAll("""[\p{Punct}]""", ""))
      .filter(_.nonEmpty)
      .map(w => (w.toLowerCase, 1))
      .reduceByKey(_ + _)
      .filter(_._2 > 4)

    counts.collect().foreach{ case (w, c) => println(s"$w -> $c") }
```

```
    // Stop the SparkSession when done
    spark.stop()
  }
}
```

Output:



Lab 10

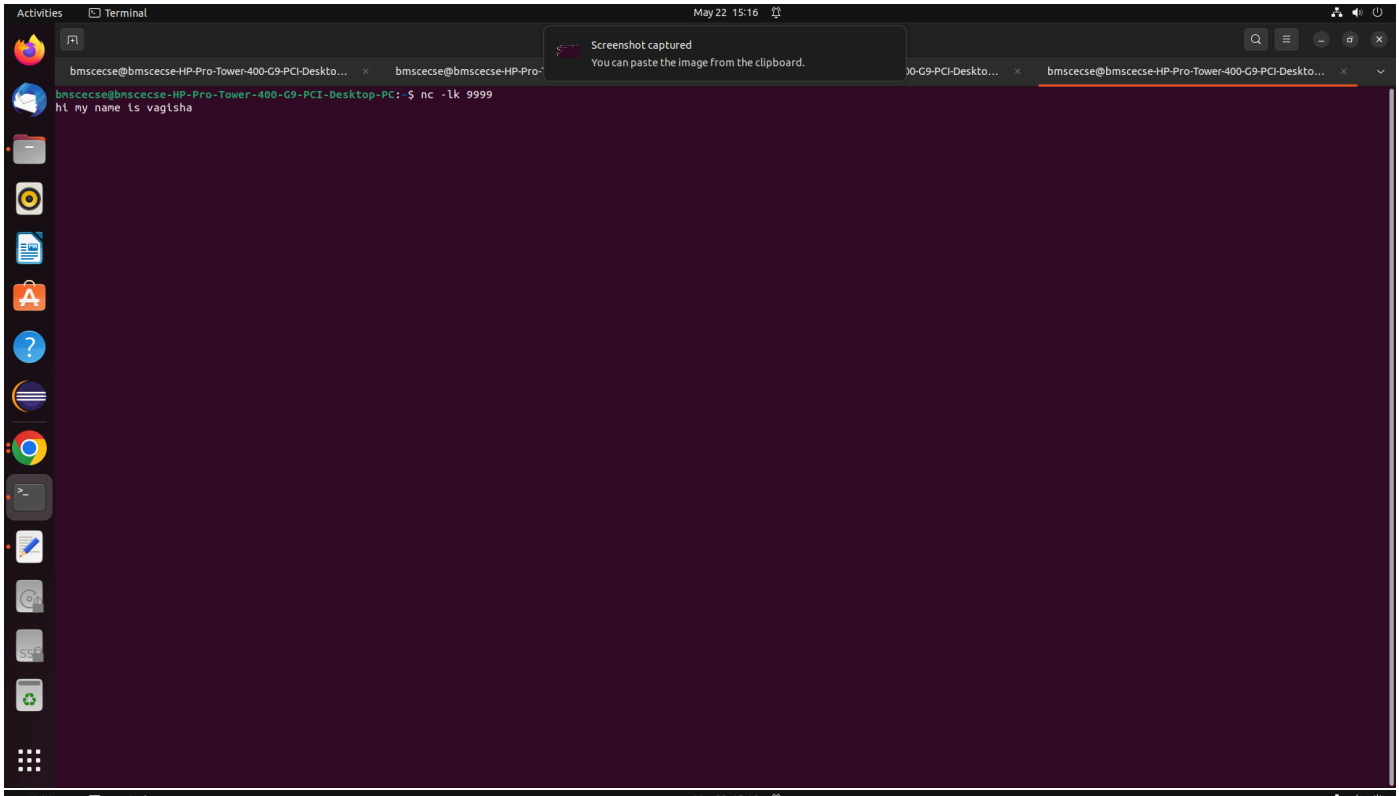
Q) Write a simple streaming program in Spark to receive text data streams on a particular port, perform basic text cleaning (like white space removal, stop words removal, lemmatization, etc.), and print the cleaned text on the screen. (Open Ended Question).

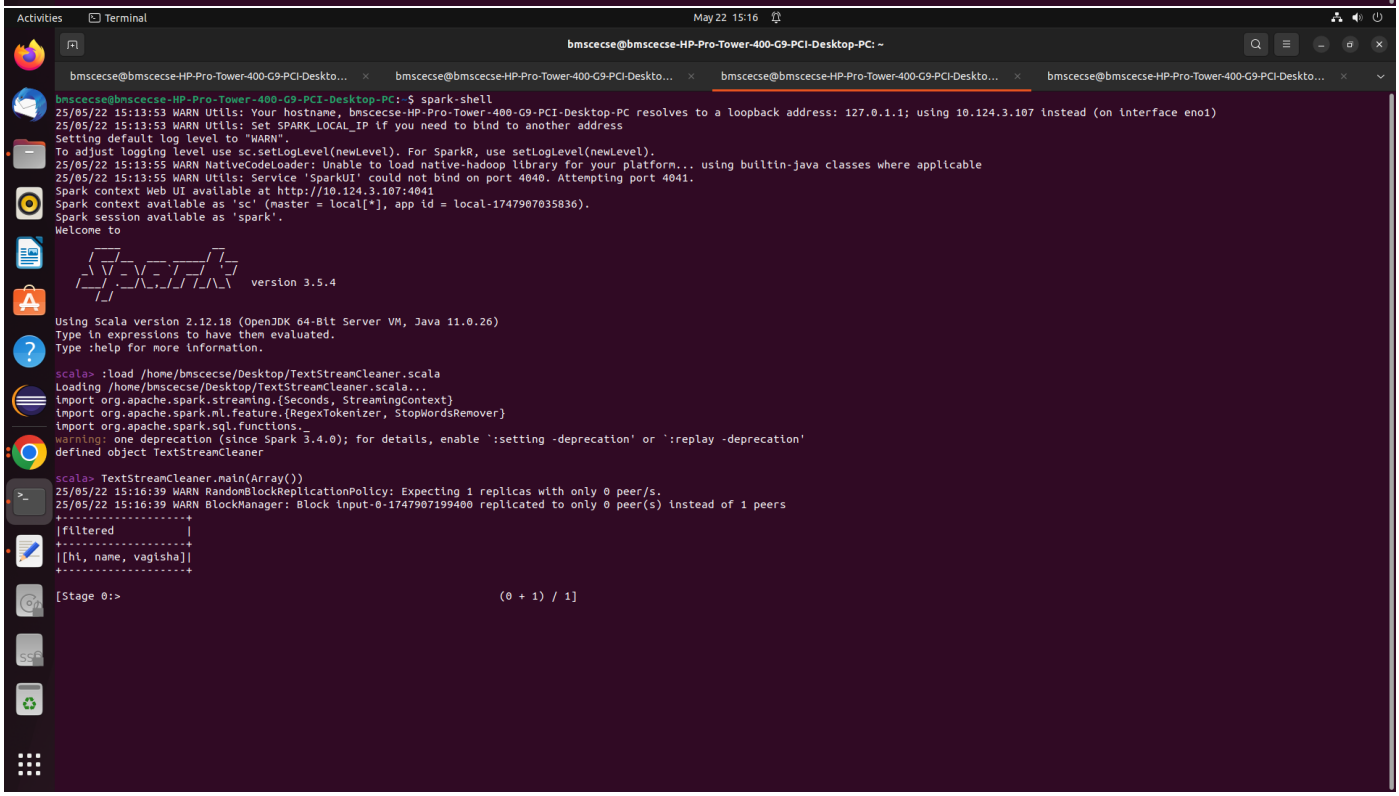
Code:

```
import org.apache.spark.streaming.{Seconds, StreamingContext}

import org.apache.spark.ml.feature.{RegexTokenizer, StopWordsRemover}

import org.apache.spark.sql.functions._


object TextStreamCleaner {

  def main(args: Array[String]): Unit = {

    // Use existing SparkContext from REPL

    val ssc = new StreamingContext(sc, Seconds(5))

    val sparkSession = spark  // Use existing SparkSession from REPL


    import sparkSession.implicits._


    val lines = ssc.socketTextStream("localhost", 9999)


    lines.foreachRDD { rdd =>

      if (!rdd.isEmpty()) {

        val df = rdd.toDF("text")


        // Tokenization

        val tokenizer = new RegexTokenizer()

          .setInputCol("text")

          .setOutputCol("words")

          .setPattern("\\W")


        val tokenizedDF = tokenizer.transform(df)
```

```
        // Stop words removal
        val remover = new StopWordsRemover()
          .setInputCol("words")
          .setOutputCol("filtered")

        val cleanedDF = remover.transform(tokenizedDF)

        cleanedDF.select("filtered").show(false)
      }
    }

    ssc.start()
    ssc.awaitTermination()
  }
}
```

Output:


Activities
Terminal
May 22 15:16
Screenshot captured
You can paste the image from the clipboard.
bmscecse@bmscecse-HP-Pro-Tower-400-G9-PCI-Deskto...
bmscecse@bmscecse-HP-Pro-Tower-400-G9-PCI-Deskto...
bmscecse@bmscecse-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~$ nc -lk 9999
hi my name is vagisha


Activities
Terminal
May 22 15:16
bmscecse@bmscecse-HP-Pro-Tower-400-G9-PCI-Desktop-PC: ~
bmscecse@bmscecse-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~$ spark-shell
25/05/22 15:13:53 WARN Utils: Your hostname, bmscecse-HP-Pro-Tower-400-G9-PCI-Desktop-PC resolves to a loopback address: 127.0.1.1; using 10.124.3.107 instead (on interface eno1)
25/05/22 15:13:53 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
25/05/22 15:13:55 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
25/05/22 15:13:55 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
Spark context Web UI available at http://10.124.3.107:4041
Spark context available as 'sc' (master = local[*], app id = local-1747907035836).
Spark session available as 'spark'.
Welcome to
version 3.5.4
Using Scala version 2.12.18 (OpenJDK 64-Bit Server VM, Java 11.0.26)
Type in expressions to have them evaluated.
Type :help for more information.
scala> :load /home/bmscecse/Desktop/TextStreamCleaner.scala
Loading /home/bmscecse/Desktop/TextStreamCleaner.scala...
import org.apache.spark.streaming.{Seconds, StreamingContext}
import org.apache.spark.ml.feature.{RegexTokenizer, StopWordsRemover}
import org.apache.spark.sql.functions._
warning: one deprecation (since Spark 3.4.0); for details, enable `:setting -deprecation' or `:replay -deprecation'
defined object TextStreamCleaner
scala> TextStreamCleaner.main(Array())
25/05/22 15:16:39 WARN RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
25/05/22 15:16:39 WARN BlockManager: Block input-0-1747907199400 replicated to only 0 peer(s) instead of 1 peers
+------------------+
|filtered |
+------------------+
|[hi, name, vagisha]|
+------------------+
[Stage 0:> (0 + 1) / 1]