Terna Engineering College
Computer Engineering Department

Program: Sem VII

Course: Big Data Analytics & Computational Lab -I (BDA&CL-I)

Experiment No. 07

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

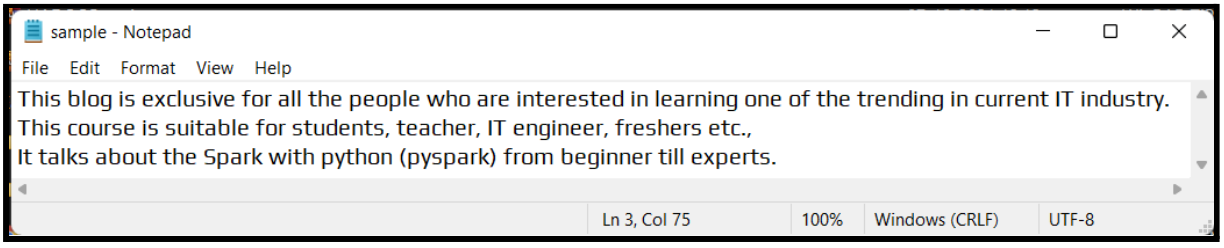
(Students must submit the soft copy as per the following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Blackboard access available)

Roll No. 50	Name: AMEY THAKUR
Class: BE-COMPS-50	Batch: B3
Date of Experiment: 05-10-2021	Date of Submission: 05-10-2021
Grade :	

Aim: To implement a word count program using Map Reduce.

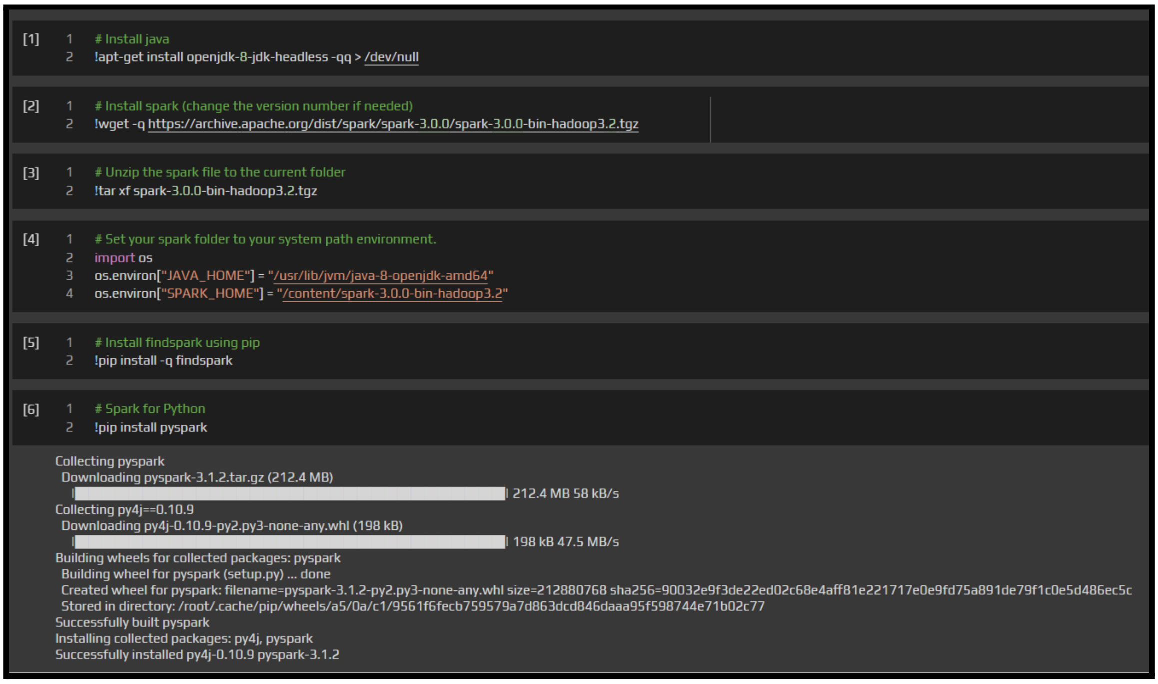
B.1 Software Code written by a student:

➔ **Input file:** sample.txt



B.2 Input and Output:

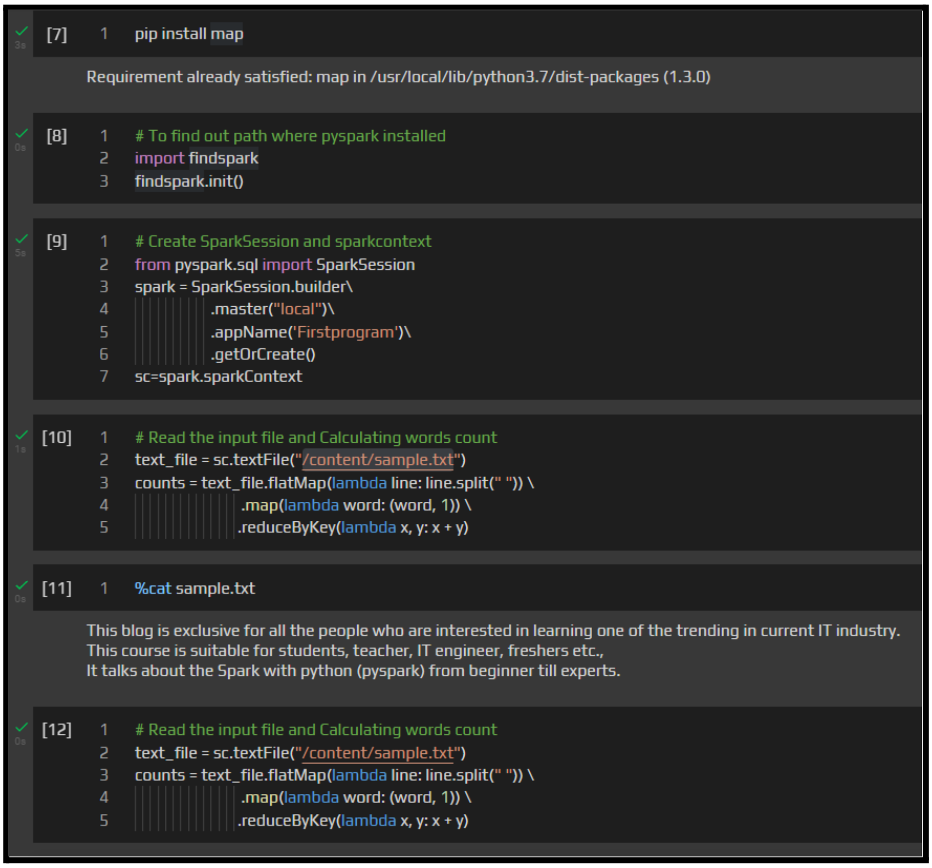
→ **Step 1:** Installing Hadoop and Pyspark.



```
[1]
# Install java
!apt-get install openjdk-8-jdk-headless -qq > /dev/null

[2]
# Install spark (change the version number if needed)
!wget -q https://archive.apache.org/dist/spark/spark-3.0.0/spark-3.0.0-bin-hadoop3.2.tgz

[3]
# Unzip the spark file to the current folder
!tar xf spark-3.0.0-bin-hadoop3.2.tgz

[4]
# Set your spark folder to your system path environment.
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.0.0-bin-hadoop3.2"

[5]
# Install findspark using pip
!pip install -q findspark

[6]
# Spark for Python
!pip install pyspark
```

```
Collecting pyspark
  Downloading pyspark-3.1.2.tar.gz (212.4 MB)
     |████████████████████████████████| 212.4 MB 58 kB/s
Collecting py4j==0.10.9
  Downloading py4j-0.10.9-py2.py3-none-any.whl (198 kB)
     |████████████████████████████████| 198 kB 47.5 MB/s
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.1.2-py2.py3-none-any.whl size=212880768 sha256=90032e9f3de22ed02c68e4aff81e221717e0e9fd75a891de79f1c0e5d486ec5c
  Stored in directory: /root/.cache/pip/wheels/a5/0a/c1/9561f6fecb759579a7d863dcd846daaa95f598744e71b02c77
Successfully built pyspark
Installing collected packages: py4j, pyspark
Successfully installed py4j-0.10.9 pyspark-3.1.2
```

→ **Step 2:** Installing Map and reading the sample text file and calculating words counts.



```
[7]
pip install map
```

```
Requirement already satisfied: map in /usr/local/lib/python3.7/dist-packages (1.3.0)
```

```
[8]
# To find out path where pyspark installed
import findspark
findspark.init()

[9]
# Create SparkSession and sparkcontext
from pyspark.sql import SparkSession
spark = SparkSession.builder\
            .master("local")\
            .appName('Firstprogram')\
            .getOrCreate()
sc=spark.sparkContext

[10]
# Read the input file and Calculating words count
text_file = sc.textFile("/content/sample.txt")
counts = text_file.flatMap(lambda line: line.split(" ")) \
                  .map(lambda word: (word, 1)) \
                  .reduceByKey(lambda x, y: x + y)

[11]
%cat sample.txt
```

```
This blog is exclusive for all the people who are interested in learning one of the trending in current IT industry.
This course is suitable for students, teacher, IT engineer, freshers etc.,
It talks about the Spark with python (pyspark) from beginner till experts.
```

```
[12]
# Read the input file and Calculating words count
text_file = sc.textFile("/content/sample.txt")
counts = text_file.flatMap(lambda line: line.split(" ")) \
                  .map(lambda word: (word, 1)) \
                  .reduceByKey(lambda x, y: x + y)
```

➔ **Step 3:** Printing each word with its respective count.

```
[13]
# Printing each word with its respective count
output = counts.collect()
for (word, count) in output:
    print("%s: %i" % (word, count))
```

```
This: 2
blog: 1
is: 2
exclusive: 1
for: 2
all: 1
the: 3
people: 1
who: 1
are: 1
interested: 1
in: 2
learning: 1
one: 1
of: 1
trending: 1
current: 1
IT: 2
industry.: 1
course: 1
suitable: 1
students,: 1
teacher,: 1
engineer,: 1
freshers: 1
etc.,: 1
It: 1
talks: 1
about: 1
Spark: 1
with: 1
python: 1
(pyspark): 1
from: 1
beginner: 1
till: 1
experts.: 1
```

➔ **Step 4:** Stopping the Spark-session and Spark context.

```
[14]
# Stopping Spark-Session and Spark context
sc.stop()
spark.stop()
```

B.3 Observations and learning:

We are able to acquire fundamental enabling techniques and scalable algorithms like Hadoop, Map Reduce and NO SQL in big data analytics.

B.4 Conclusion:

We have learned To implement a word count program using Map Reduce.

B.5 Question of Curiosity:

1. What is Flatten?

Ans:

The FLATTEN operator looks like a UDF syntactically, but it is actually an operator that changes the structure of tuples and bags in a way that a UDF cannot. Flatten un-nests tuples as well as bags. The idea is the same, but the operation and result are different for each type of structure.

2. How does Pig differ from MapReduce?

Ans:

Sr. No.	Pig	MapReduce
1	It is a Data Flow Language.	It is a Data Processing Language.
2	It converts the query into map-reduce functions.	It converts the job into map-reduce functions.
3	It is a High-level Language.	It is a Low-level Language.
4	Makes it easy for the user to perform Join operations.	It is difficult for the user to perform join operations.
5	There is less compilation time as the Pig operator converts it into MapReduce jobs.	It has several jobs therefore execution time is more.