INDEX

EXP NO:01 **DATE:**

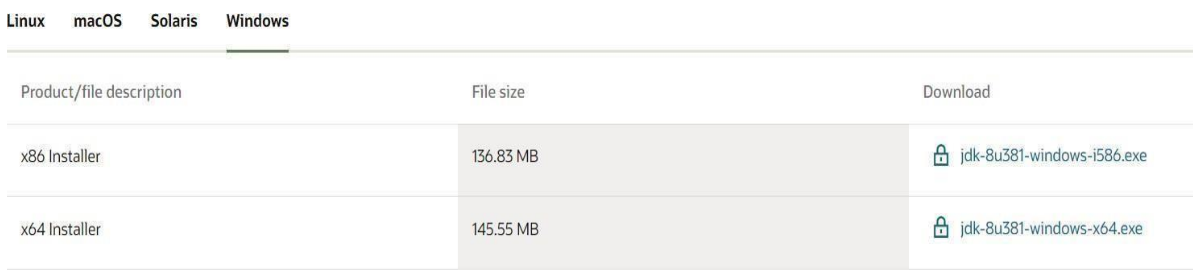
Downloading and installing Hadoop; Understanding different Hadoop modes. Start-up scripts, Configuration files.

ALGORITHM:

STEP 1: To install Hadoop the primary task is to setup and install java environment

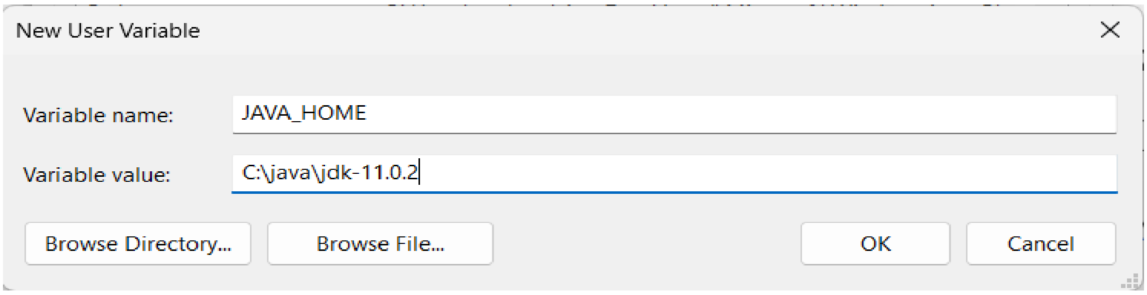
STEP 2: The java version that needed to be installed depends on the Hadoop‟s version. Here we are installing the latest version of Hadoop which is 3.3.0 which supports java version varyingfrom 8-11(runtime only).

STEP 3: Use the following link to install java https://www.oracle.com/java/technologies/downloads/#java8-windows



Product/file description	File size	Download
x86 Installer	136.83 MB	jdk-8u381-windows-i586.exe
x64 Installer	145.55 MB	jdk-8u381-windows-x64.exe

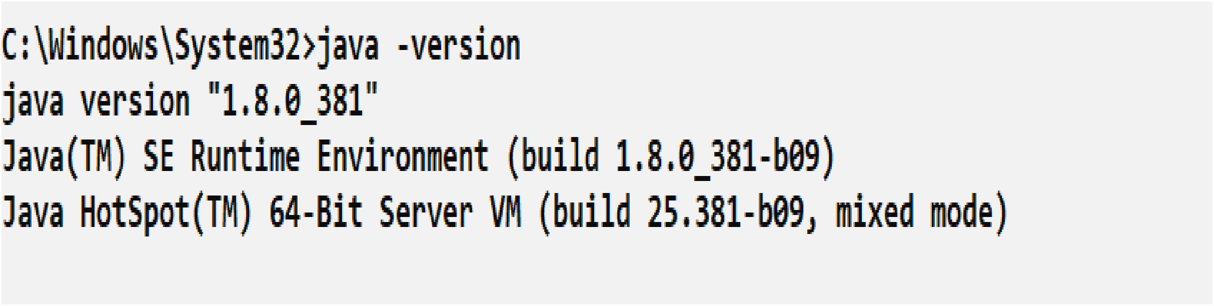
STEP 4: After installing java setup, the java environment in environmental variables directing the bin folder inside the java folder (**C:\java\jdk-11.0.2\bin**) copy the path till bin folder and paste it in the environmental variable define the new path and add the bin folder location as **JAVA_HOME=”C:\java\jdk11.0.2\bin”** and apply the changes



Variable name: JAVA_HOME

Variable value: C:\java\jdk-11.0.2

STEP 5: Now after setting up the java environment check the setup has been successfully set by using **java -version** command in your command prompt and it should display the version ofjava you have installed.

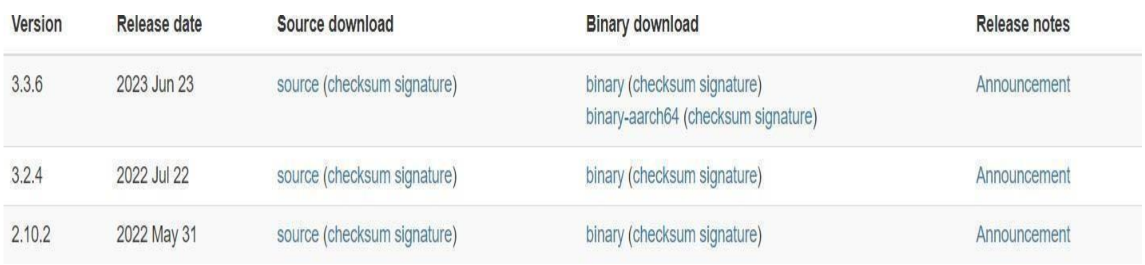


```
C:\Windows\System32>java -version
java version "1.8.0_381"
Java(TM) SE Runtime Environment (build 1.8.0_381-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.381-b09, mixed mode)
```

STEP 7: Hadoop is Unix distribution-based file with tar.gz extension we have to extract the file using the 7-zip manager which supports multiple formats follow this link to install 7-zip https://7-zip.org/

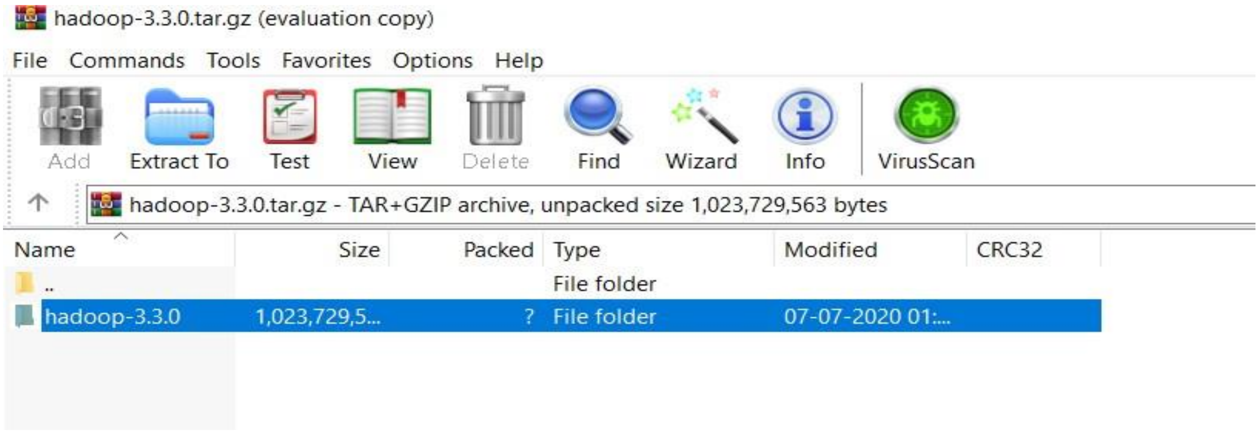
STEP 8: Now install the Notepad++ text editor which is further used to modify or edit the configuration file within Hadoop as per our requirement

STEP 9: After installing and setting up all the required application install Hadoop from the officialApache Hadoop website https://hadoop.apache.org/releases.html download the binary download which can run directly without any need for compilation.



Version	Release date	Source download	Binary download	Release notes
3.3.6	2023 Jun 23	source (checksum signature)	binary (checksum signature) binary-aarch64 (checksum signature)	Announcement
3.2.4	2022 Jul 22	source (checksum signature)	binary (checksum signature)	Announcement
2.10.2	2022 May 31	source (checksum signature)	binary (checksum signature)	Announcement

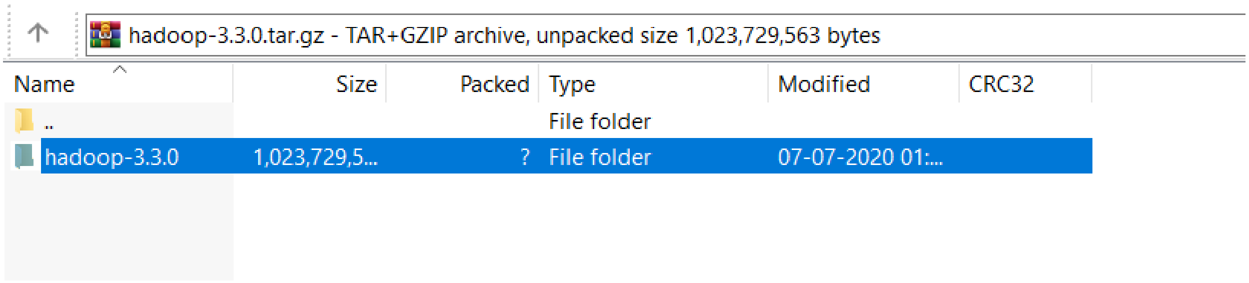
STEP 10: Run 7-zip manager as administrator and navigate to the path where Hadoop is located for extract the compiled binary download of Hadoop



hadoop-3.3.0.tar.gz - TAR+GZIP archive, unpacked size 1,023,729,563 bytes

Name	Size	Packed	Type	Modified	CRC32
..			File folder		
hadoop-3.3.0	1,023,729,5...	?	File folder	07-07-2020 01:...	

STEP 11: After doing the extraction process there is another compressed file with in the extracted file extract that as well.



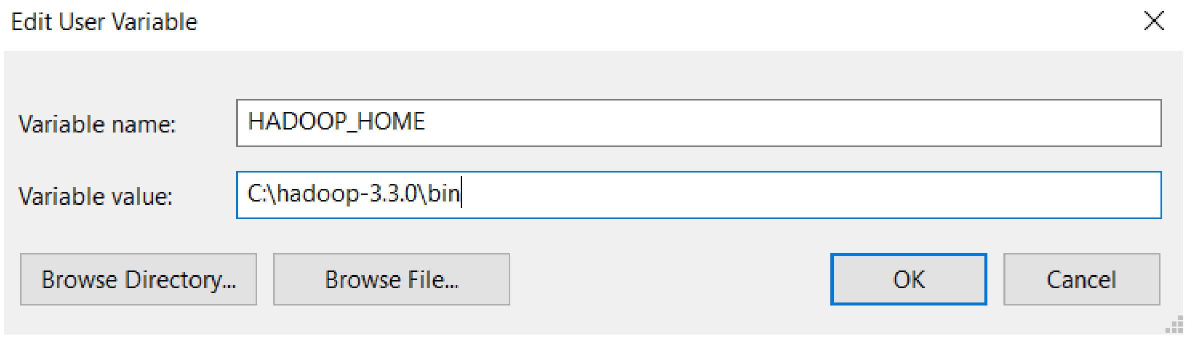
hadoop-3.3.0.tar.gz - TAR+GZIP archive, unpacked size 1,023,729,563 bytes

Name	Size	Packed	Type	Modified	CRC32
..			File folder		
hadoop-3.3.0	1,023,729,5...	?	File folder	07-07-2020 01:...	

STEP 12: From the extracted folder replace the bin file with the reliable windows supported configured file here is the drive link to download the bin file

https://drive.google.com/file/d/1kVhX9snOZ3oLUxDjh3AVI8fcRnEWAAE4/view

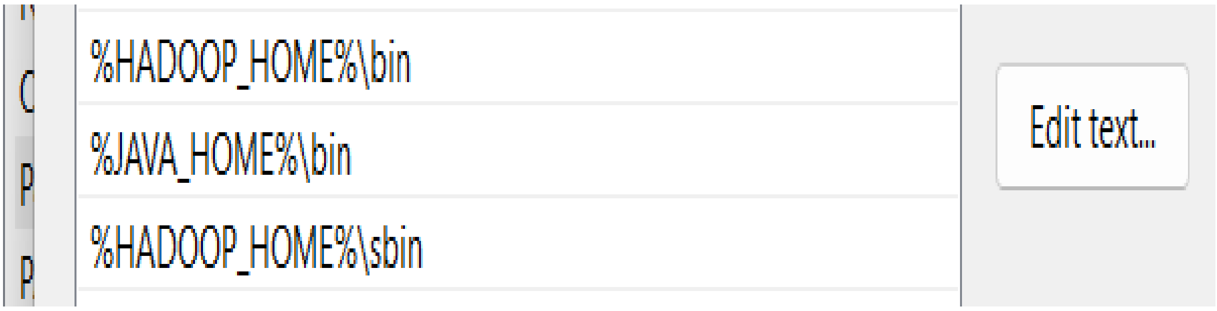
STEP 13: Setup the Hadoop environment in environment variable and set path location as **HADOOP_HOME= "C:\ hadoop-3.3.0\bin"**



Variable name: HADOOP_HOME

Variable value: C:\hadoop-3.3.0\bin

STEP 14: Add the Hadoop bin and sbin path location by editing the path. And add the bin, sbin location there

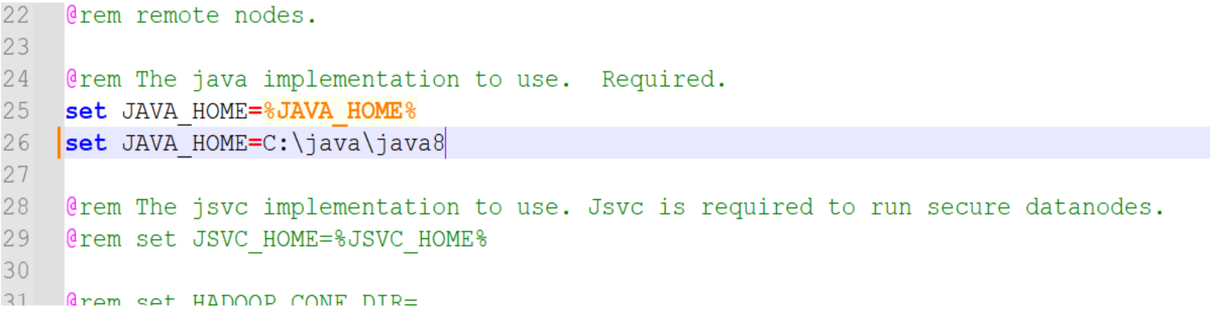


%HADOOP_HOME%\bin

%JAVA_HOME%\bin

%HADOOP_HOME%\sbin

STEP 15: Now open etc folder inside the Hadoop folder and locate the file Hadoop-env.cmd andset the java home location



```
@rem remote nodes.

@rem The java implementation to use.  Required.
set JAVA_HOME=%JAVA_HOME%
set JAVA_HOME=C:\java\java8

@rem The jsvc implementation to use. Jsvc is required to run secure datanodes.
@rem set JSVC_HOME=%JSVC_HOME%

@rem set HADOOP_CONF_DIR=
```

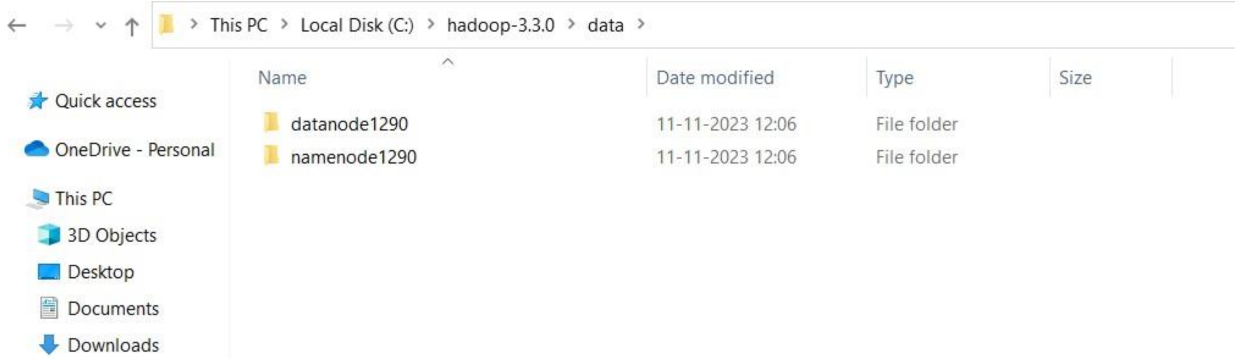
STEP 16: Edit the following configuration XML files core-site.xml, hdfs-site.xml, mapred-site.xml, yarn-site.xml are used to configure the behaviour of your Hadoop Cluster and save them.

STEP 17: Starting from core-site.xml edit it using notepad++ and following the program to configure.

PROGRAM (CORE-SITE.XML)

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

STEP 18: To edit the hdfs-site.xml create Date folder and then within the Data folder create Namenode, Datenode which are used to manage and cluster flow date and log files.



This PC > Local Disk (C:) > hadoop-3.3.0 > data >

Name	Date modified	Type	Size
datanode1290	11-11-2023 12:06	File folder	
namenode1290	11-11-2023 12:06	File folder	

STEP 19: Now open the hdfs-site.xml in notepad++ and add the following program

```
  <configuration
 <property>
  <name>dfs.replication</name>
 <value>1</value>
 </property>
 <property>
 <name>dfs.namenode.name.dir</name>
 <value> C:/hadoop-3.3.0/Data/datanode1290 </value>
 </property>
  <property>
 <name>dfs.datanode.data.dir</name>
 <value>> C:/hadoop-3.3.0/Data/namenode1290 </value>
</property>
</configuration>
```

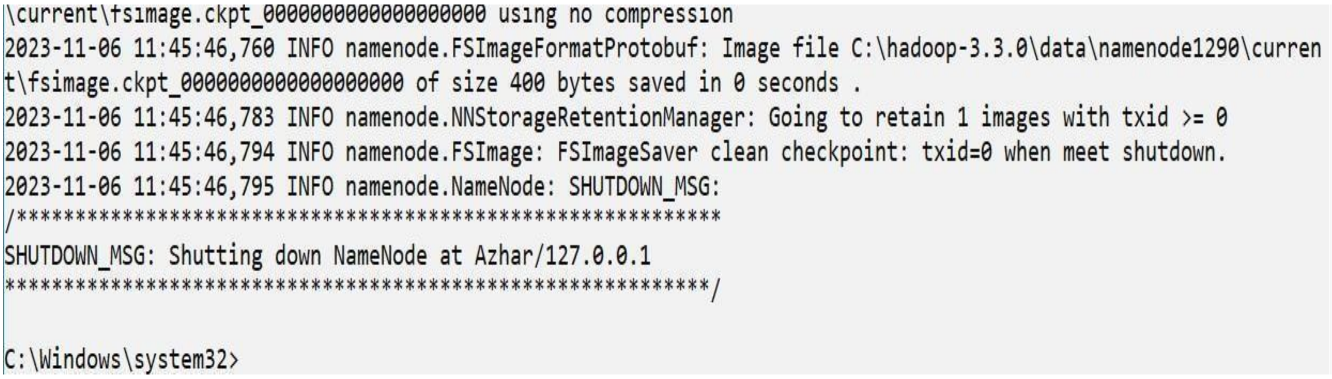
STEP 20: Edit mapred-site.xml file using notepad++ add this following program

```
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

STEP 21: Edit the yarn-site.xml with following program and save it.

```
  <configuration>
  <property>
 <name>yarn.nodemanager.aux-services</name>
 <value>mapreduce_shuffle</value>
 </property>
 <property>
 <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
  <value>org.apache.hadoop.mapred.ShuffleHandler</value>
 </property>
 </configuration>
```

STEP 22: Now save them and open command prompt as administrator and run the followingcommand to **hdfs namenode -format** to format the contents of namenode

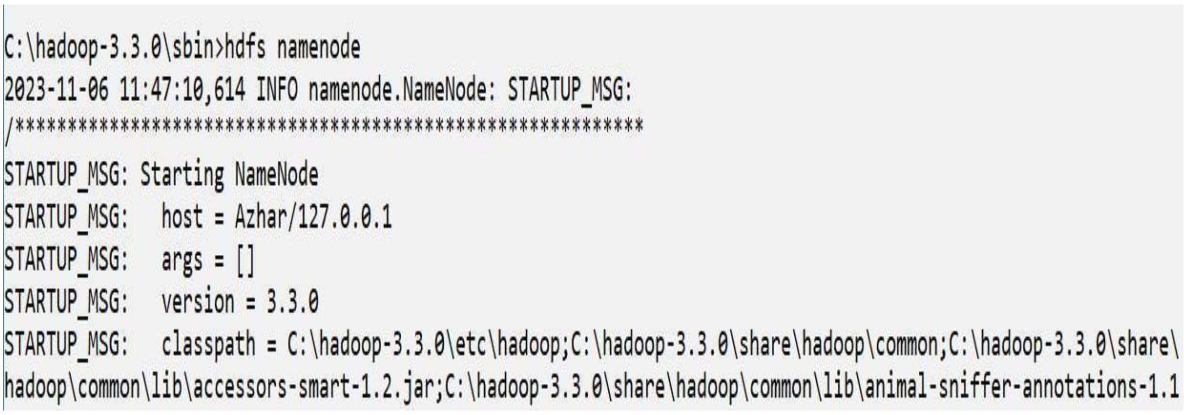


```
\current\fsimage.ckpt_0000000000000000000 using no compression
2023-11-06 11:45:46,760 INFO namenode.FSImageFormatProtobuf: Image file C:\hadoop-3.3.0\data\namenode1290\curren
t\fsimage.ckpt_0000000000000000000 of size 400 bytes saved in 0 seconds .
2023-11-06 11:45:46,783 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2023-11-06 11:45:46,794 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2023-11-06 11:45:46,795 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at Azhar/127.0.0.1
************************************************************/

C:\Windows\system32>
```

STEP 23: To check the daemons configured correctly open command prompt asadministrator and run the following command‟s

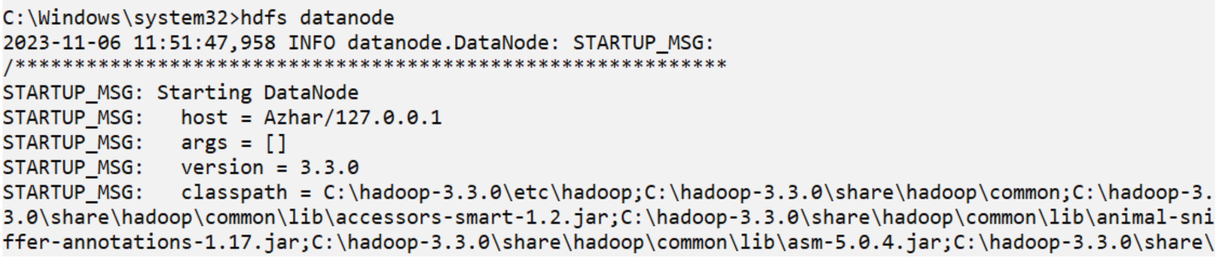
hdfs namenode, hdfs Datanode, yarn nodemanager, yarn resourcemanager

hdfs namenode



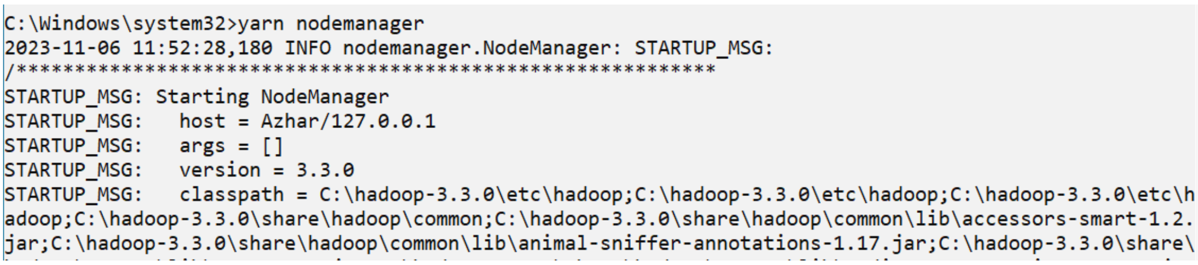
```
C:\hadoop-3.3.0\sbin>hdfs namenode
2023-11-06 11:47:10,614 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = Azhar/127.0.0.1
STARTUP_MSG:   args = []
STARTUP_MSG:   version = 3.3.0
STARTUP_MSG:   classpath = C:\hadoop-3.3.0\etc\hadoop;C:\hadoop-3.3.0\share\hadoop\common;C:\hadoop-3.3.0\share\
hadoop\common\lib\accessors-smart-1.2.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\animal-sniffer-annotations-1.1
```

hdfs datanode



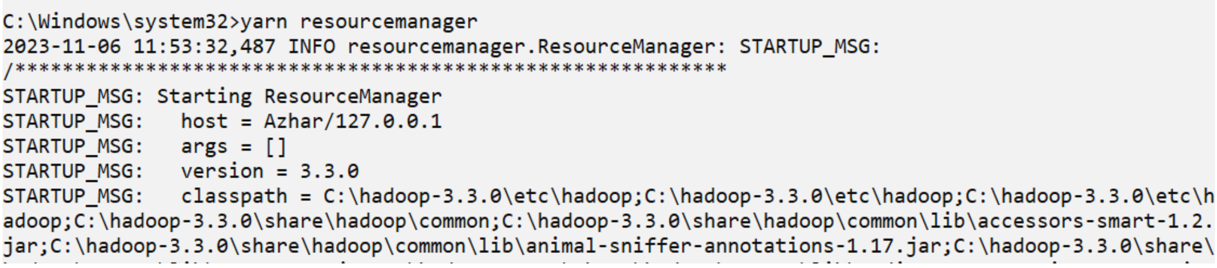
```
C:\Windows\system32>hdfs datanode
2023-11-06 11:51:47,958 INFO datanode.DataNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting DataNode
STARTUP_MSG:   host = Azhar/127.0.0.1
STARTUP_MSG:   args = []
STARTUP_MSG:   version = 3.3.0
STARTUP_MSG:   classpath = C:\hadoop-3.3.0\etc\hadoop;C:\hadoop-3.3.0\share\hadoop\common;C:\hadoop-3.
3.0\share\hadoop\common\lib\accessors-smart-1.2.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\animal-sni
ffer-annotations-1.17.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\asm-5.0.4.jar;C:\hadoop-3.3.0\share\
```

yarn nodemanager



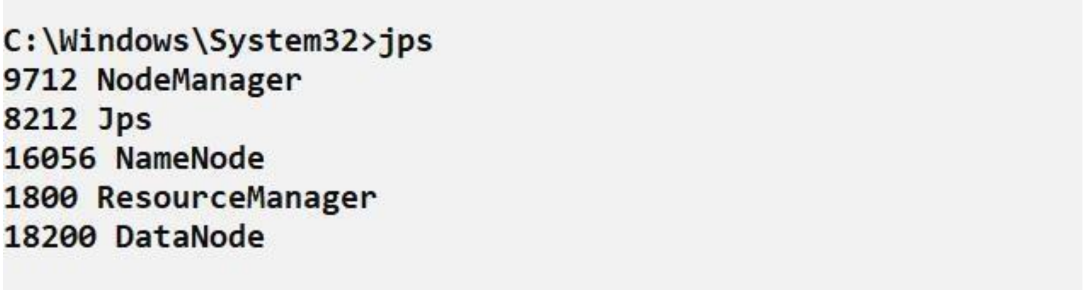
```
C:\Windows\system32>yarn nodemanager
2023-11-06 11:52:28,180 INFO nodemanager.NodeManager: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NodeManager
STARTUP_MSG:   host = Azhar/127.0.0.1
STARTUP_MSG:   args = []
STARTUP_MSG:   version = 3.3.0
STARTUP_MSG:   classpath = C:\hadoop-3.3.0\etc\hadoop;C:\hadoop-3.3.0\etc\hadoop;C:\hadoop-3.3.0\etc\h
adoop;C:\hadoop-3.3.0\share\hadoop\common;C:\hadoop-3.3.0\share\hadoop\common\lib\accessors-smart-1.2.
jar;C:\hadoop-3.3.0\share\hadoop\common\lib\animal-sniffer-annotations-1.17.jar;C:\hadoop-3.3.0\share\
```

yarn resourcemanager



```
C:\Windows\system32>yarn resourcemanager
2023-11-06 11:53:32,487 INFO resourcemanager.ResourceManager: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting ResourceManager
STARTUP_MSG:   host = Azhar/127.0.0.1
STARTUP_MSG:   args = []
STARTUP_MSG:   version = 3.3.0
STARTUP_MSG:   classpath = C:\hadoop-3.3.0\etc\hadoop;C:\hadoop-3.3.0\etc\hadoop;C:\hadoop-3.3.0\etc\h
adoop;C:\hadoop-3.3.0\share\hadoop\common;C:\hadoop-3.3.0\share\hadoop\common\lib\accessors-smart-1.2.
jar;C:\hadoop-3.3.0\share\hadoop\common\lib\animal-sniffer-annotations-1.17.jar;C:\hadoop-3.3.0\share\
```

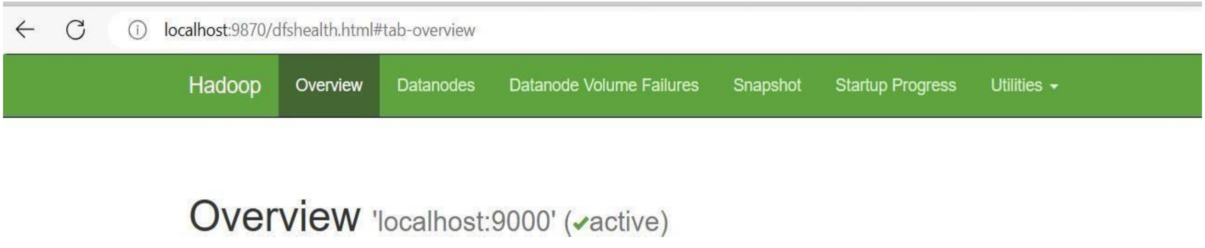
STEP 24: To check the check the daemons that are running in background we can use Java Virtual Machine Process Status which is used to list the java virtual machines that are currently running on a system it is used to display the process ID(PID) of each JVM



```
C:\Windows\System32>jps
9712 NodeManager
8212 Jps
16056 NameNode
1800 ResourceManager
18200 DataNode
```

STEP 25: Now we can access the Namenode and Datanode as web user interface (web-UI)by using the following localhost address

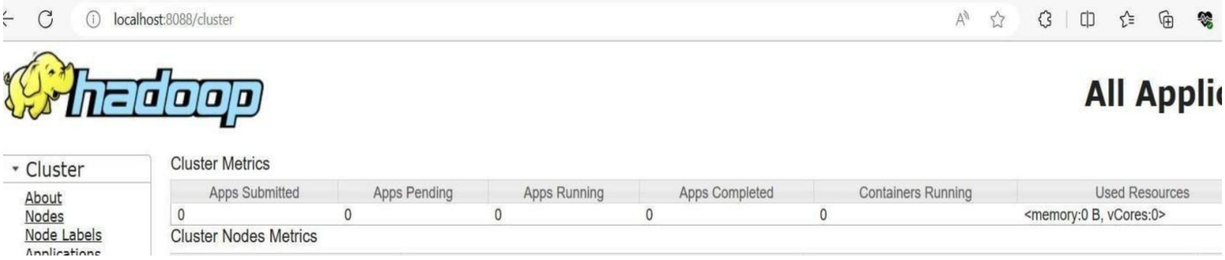
localhost:9870



Overview 'localhost:9000' (✓active)

STEP 26: To access the Datanode use the following localhost address

localhost:8088



Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Used Resources
0	0	0	0	0	<memory:0 B, vCores:0>

Cluster Nodes Metrics

EXP NO:02 **DATE:**

IMPLEMENTATION OF HADOOP FILE MANAGEMENT TASKS

ALGORITHM:

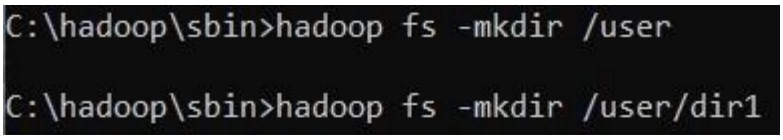
1. Creating a directory in HDFS

SYNTAX:

hadoop fs-mkdir <paths>

EXAMPLE:

hadoop fs-mkdir /user
hadoop fs-mkdir /user/dirl



```
C:\hadoop\sbin>hadoop fs -mkdir /user

C:\hadoop\sbin>hadoop fs -mkdir /user/dir1
```

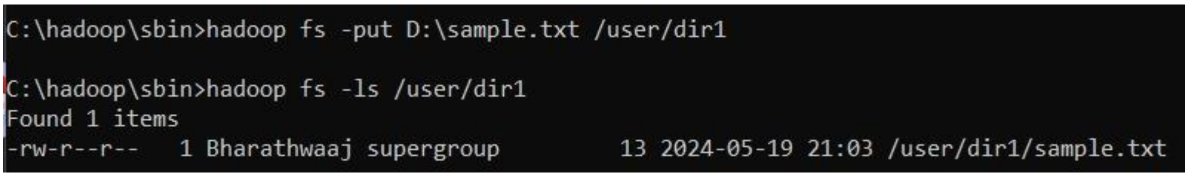
2. Listing the contents of a directory

SYNTAX:

hadoop fs-Is <directory name>

EXAMPLE:

hadoop fs-Is/user hadoop fs-Is/user/dirl



```
C:\hadoop\sbin>hadoop fs -put D:\sample.txt /user/dir1

C:\hadoop\sbin>hadoop fs -ls /user/dir1
Found 1 items
-rw-r--r--   1 Bharathwaaj supergroup         13 2024-05-19 21:03 /user/dir1/sample.txt
```

3. Uploading and downloading a file in HDFS

SYNTAX: (UPLOAD)

hadoop fs-put <local file system path> <hdfs destination path>

EXAMPLE:

hadoop fs -put C:\Home\samplefile.txt.txt/user/dir1/ hadoop fs-Is/user/dirl

SYNTAX: (DOWNLOAD)

hadoop fs-get<hdfs sre> <local dat>

EXAMPLE:

hadoop fs-get/user/dir1/samplefile.txt C:\Home Hadoopfiles

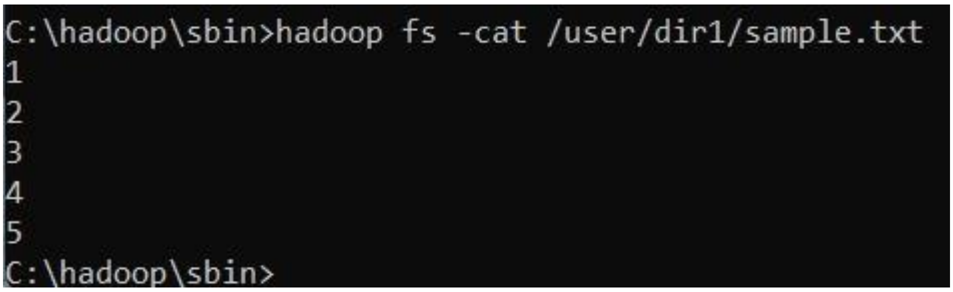
4. See the contents of a file

SYNTAX:

hadoop fs-cat <path[filename]>

EXAMPLE:

hadoop fs-cat/user/dir1/samplefile.txt



```
C:\hadoop\sbin>hadoop fs -cat /user/dir1/sample.txt
1
2
3
4
5
C:\hadoop\sbin>
```

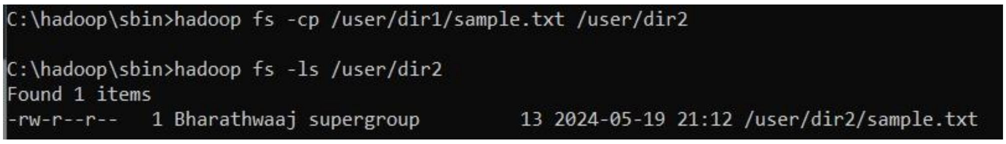
5. Copy a file from source to destination

SYNTAX:

hadoop fs-cp<src> <dst>

EXAMPLE:

hadoop fs/user/dir1/samplefile.txt/user/dir2



```
C:\hadoop\sbin>hadoop fs -cp /user/dir1/sample.txt /user/dir2

C:\hadoop\sbin>hadoop fs -ls /user/dir2
Found 1 items
-rw-r--r--   1 Bharathwaaj supergroup         13 2024-05-19 21:12 /user/dir2/sample.txt
```

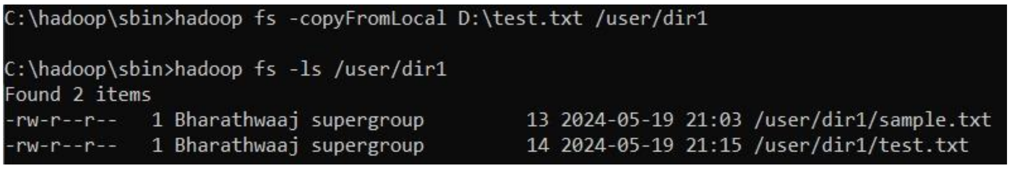
6. Copy a file from and to local file system to hdfs

SYNTAX: (FROM)

hadoop fs-copyFromLocal <local file system file path> <hdfs dst>

EXAMPLE:

hadoop fs-copyFromLocal C:\Home\test.txt/user/dirl



```
C:\hadoop\sbin>hadoop fs -copyFromLocal D:\test.txt /user/dir1

C:\hadoop\sbin>hadoop fs -ls /user/dir1
Found 2 items
-rw-r--r--   1 Bharathwaaj supergroup         13 2024-05-19 21:03 /user/dir1/sample.txt
-rw-r--r--   1 Bharathwaaj supergroup         14 2024-05-19 21:15 /user/dir1/test.txt
```

SYNTAX: (TO)

hadoop fs-copy ToLocal <hdfs sre> <local dst>

EXAMPLE:

hadoop fs-copy/ToLocal/user/dir1/samplefile.txt C:\Home\copy

7. Move file from source to destination

SYNTAX:

hadoop fs -mv <sre> <dt>

EXAMPLE:

hadoop fis-mv/user/dir1/test.txt/user/dir2

8. Remove a file or directory in hdfs

SYNTAX:

hadoop fs-rm <arg>

EXAMPLE:

hadoop fs -m/user/dirl/samplefile.txt

SYNTAX: (Recursive method for deleting directories)

hadoop fs-rm-r <arg>

EXAMPLE:

hadoop fs-rm-r/user/dirl

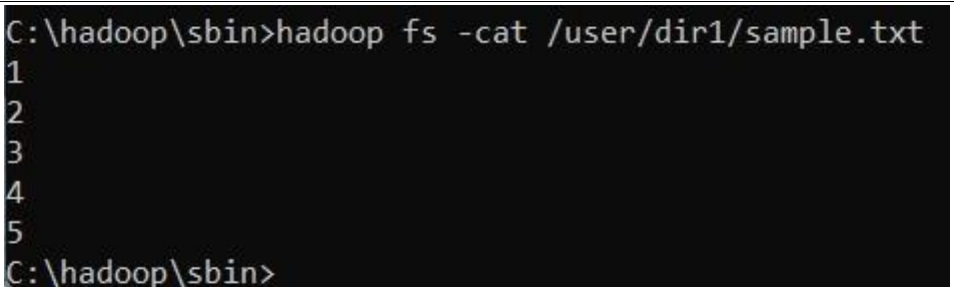
9. Display few lines of a file

SYNTAX:

hadoop fs-tail <path[filename]>

EXAMPLE:

hadoop fs-tail/user/dir2/samplefile.txt



```
C:\hadoop\sbin>hadoop fs -cat /user/dir1/sample.txt
1
2
3
4
5
C:\hadoop\sbin>
```

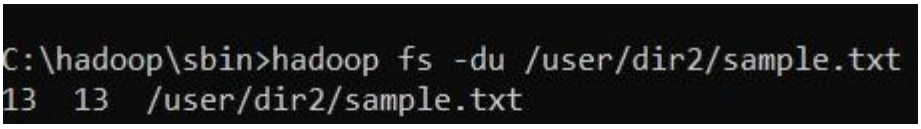
10. Display the aggregate length of a file

SYNTAX:

hadoop fs-du <path>

EXAMPLE:

hadoop fs-du/user/dir2/samplefile.txt



```
C:\hadoop\sbin>hadoop fs -du /user/dir2/sample.txt
13  13  /user/dir2/sample.txt
```

EXP NO:03 **DATE:**

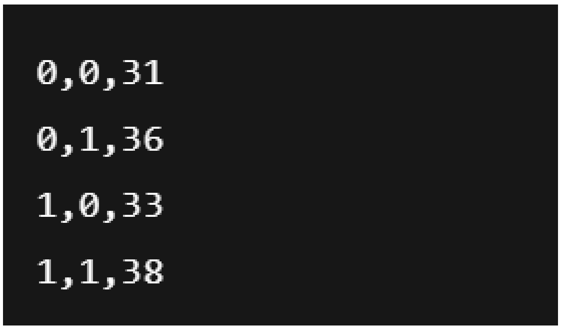
IMPLEMENT OF MATRIX MULTIPLICATION WITH HADOOP MAP REDUCE

PROGRAM :

```
import java.io.IOException; import java.util.*;
import java.util.AbstractMap.SimpleEntry;
import java.util.Map.Entry;
import org.apache.hadoop.fs.Path;
importorg.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
importorg.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
public class TwoStepMatrixMultiplication {
public static class Map extends Mapper<LongWritable, Text, Text, Text> {
public void map(LongWritable key, Text value, Context
context) throws IOException, InterruptedException {
String line = value.toString();
String[] indicesAndValue = line.split(",");
Text outputKey = new Text();
Text outputValue = new Text();
if(indicesAndValue[0].equals("A"))
{
outputKey.set(indicesAndValue[2]);
outputValue.set("A," + indicesAndValue[1] + ","
+indicesAndValue[3]);
context.write(outputKey, outputValue);
}
else
{
outputKey.set(indicesAndValue[1]);
outputValue.set("B," + indicesAndValue[2] + "," +
indicesAndValue[3]);
context.write(outputKey, outputValue);
}
}
}

public static class Reduce extends Reducer<Text, Text, Text, Text> {
public void reduce(Text key, Iterable<Text> values, Context context) throws IOException,
InterruptedException {
String[] value;
ArrayList<Entry<Integer, Float>> listA = newArrayList<Entry<Integer,
Float>>();
ArrayList<Entry<Integer, Float>> listB = new
ArrayList<Entry<Integer, Float>>();
for (Text val : values)
{
value = val.toString().split(",");
if (value[0].equals("A"))
{
listA.add(new SimpleEntry<Integer, Float>(Integer.parseInt(value[1]),
Float.parseFloat(value[2])));
}

else
{

listB.add(new SimpleEntry<Integer, Float>(Integer.parseInt(value[1]),
Float.parseFloat(value[2])));
}
}
String i;
float a_ij;
String k;
float b_jk;
Text outputValue = new Text();
for (Entry<Integer, Float> a: listA)
{
i = Integer.toString(a.getKey());
a_ij = a.getValue();
for (Entry<Integer, Float> b : listB) {
k =Integer.toString(b.getKey());
b_jk = b.getValue();
outputValue.set(i + "," + k + "," + Float.toString(a_ij*b_jk));
context.write(null, outputValue);

}
}
}
}

public static void main(String[] args) throws Exception {
Configuration conf = new Configuration();
Job job = new Job(conf,
"MatrixMatrixMultiplicationTwoSteps");

job.setJarByClass(TwoStepMatrixMultiplication.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(Text.class);
job.setMapperClass(Map.class);
job.setReducerClass(Reduce.class);
job.setInputFormatClass(TextInputFormat.class);
job.setOutputFormatClass(TextOutputFormat.class);
```

```
FileInputFormat.addInputPath(job, new Path("hdfs://
127.0.0.1:9000/matrixin"));
FileOutputFormat.setOutputPath(job, new Path("hdfs://
127.0.0.1:9000/matrixout"));
job.waitForCompletion(true);
}
```

OUTPUT :



```
0,0,31
0,1,36
1,0,33
1,1,38
```

EXP NO:04 **DATE:**

BASIC WORD COUNT MAP REDUCE PROGRAM

PROGRAM:

```
package com.mapreduce, java,
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable,
import org.apache.hadoop.io. Long Writable;
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 WC Mapper extends MapReduceBase implements Mapper Long Writable, Text,
Text, Int Writable>
private final static Int Writable one new IntWritable(1);
private Text word new Text();
public void map(LongWritable key, Text value OutputCollector Text.IntWritable output
Reporter reporter) throws IOException
String line value.toString();
StringTokenizer tokenizer = new StringTokenizer(line);
while (tokenizer.hasMoreTokens())
word.set(tokenizer.nextToken());
output.collect(word, one);
```

STEP 7: Now Create another class with name "WC Reducer.java" and paste the below program in it.

PROGRAM:

```
package com.mapreduce.java;
import java.io.IOException;
import java.util.Iterator,
import org.apache.hadoop.io.Int Writable;
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 WC Reducer extends MapReduceBase implements
Reducer Text,IntWritable, Text, Int Writable>
public void reduce(Text key, Iterator Int Writable values, OutputCollector Text, IntWritable>
output, Reporter reporter) throws IOException
int sum-0;
```

```
while (values.hasNext())
mum values.next().get();
output.collect(key.new IntWritable(sum));
```

STEP 8: Now, Create another class with name "WC_runner.java" and paste the below program in it.

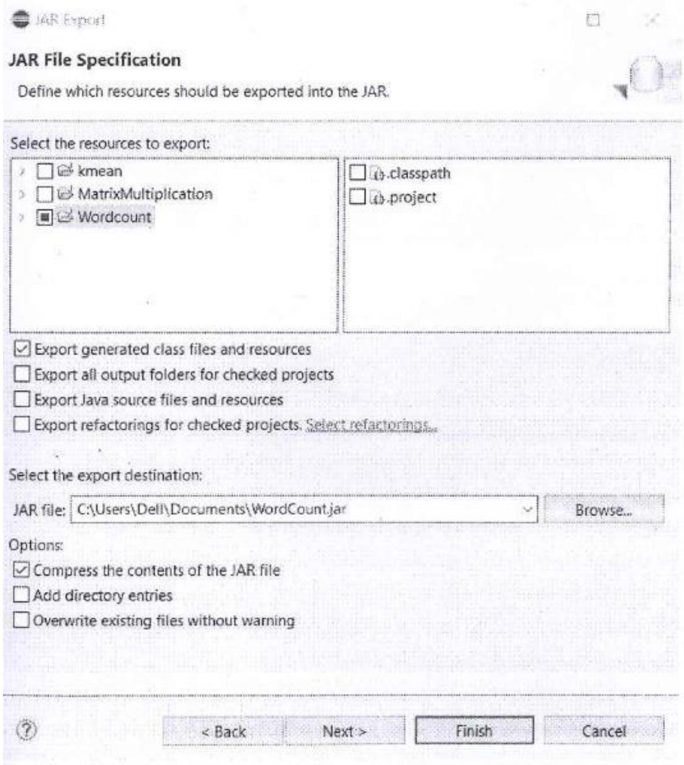
PROGRAM:

```
package com.mapreduce.java;
import java.io.IOException;
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.mapred.TextInputFormat;
import org.apache.hadoop.mapred. TextOutputFormat;
public class WC_Runner
public static void main(String[] args) throws IOExceptior JobConf conf new
JobConf(WC_Runner.class);
conf.setJobName("WordCount"); conf.setOutputKeyClass(Text.class);
conf.setOutputValueClass(IntWritable.class);
conf.setMapperClass (WC_Mapper.class);
conf.setCombinerClass(WC_Reducer.class);
conf.setReducerClass (WC Reducer.class);
conf.setInputFormat(TextInputFormat.class);
conf.setOutputFormat(TextOutputFormat.class);
FilelnputFormat.setInputPaths(conf.new Path(args[0]));
FileOutputFormat.setOutputPath(conf.new Path(args[1])): JobClient.runJob(conf);
```

STEP 9: To resolve the errors in the programs we should add two External jar files to it.

- Hadoop common :2.7.3 jar
- Hadoop_mapreduce:client:core 2.7.1.jar

STEP 10: Now export the project into a Jar file and name it as "WordCount.jar"



JAR Export

JAR File Specification

Define which resources should be exported into the JAR.

Select the resources to export:

- □ kmean
- □ MatrixMultiplication
- ■ Wordcount

□ .classpath
□ .project

☑ Export generated class files and resources
□ Export all output folders for checked projects
□ Export Java source files and resources
□ Export refactorings for checked projects. Select refactorings...

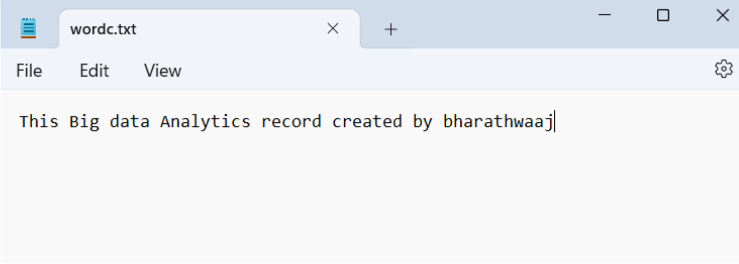
Select the export destination:

JAR file: C:\Users\Dell\Documents\WordCount.jar

Options:

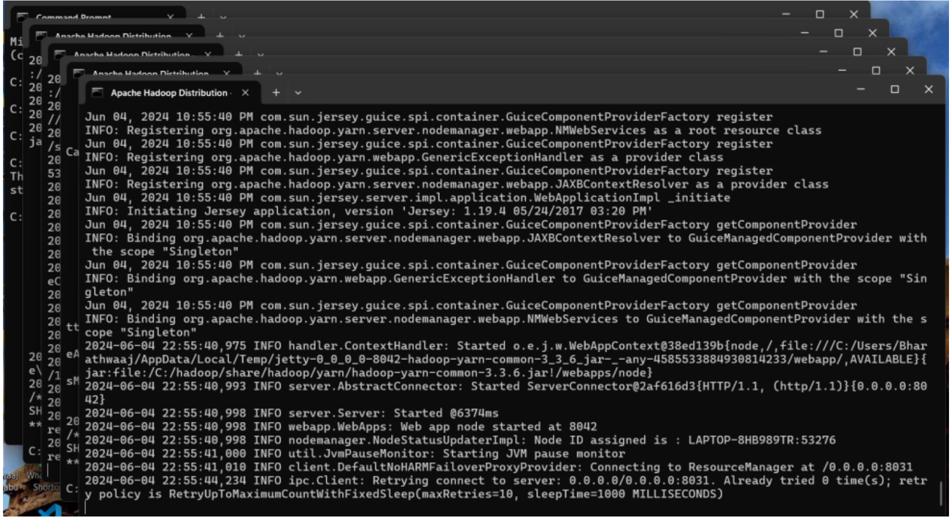
☑ Compress the contents of the JAR file
□ Add directory entries
□ Overwrite existing files without warning

STEP 11: Now create a Text file in Notepad and name it as "worde.txt" and write some content inside the text file and save it.



```
This Big data Analytics record created by bharathwaaj
```

STEP 12: Now run all the demons in Hadoop.

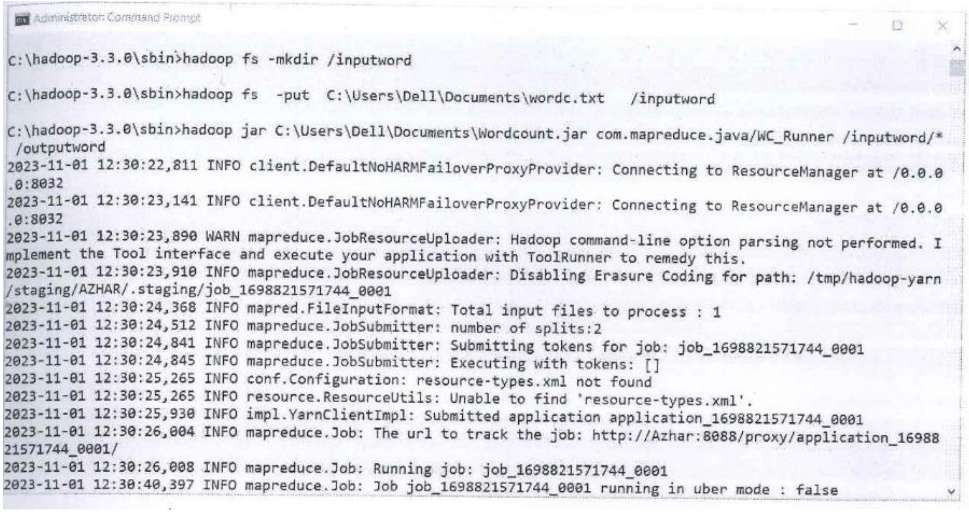


Command Prompt
Apache Hadoop Distribution
Apache Hadoop Distribution
Apache Hadoop Distribution
Apache Hadoop Distribution

```
Jun 04, 2024 10:55:40 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory register
INFO: Registering org.apache.hadoop.yarn.server.nodemanager.webapp.NMWebServices as a root resource class
Jun 04, 2024 10:55:40 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory register
INFO: Registering org.apache.hadoop.yarn.webapp.GenericExceptionHandler as a provider class
Jun 04, 2024 10:55:40 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory register
INFO: Registering org.apache.hadoop.yarn.server.nodemanager.webapp.JAXBContextResolver as a provider class
Jun 04, 2024 10:55:40 PM com.sun.jersey.server.impl.application.WebApplicationImpl _initiate
INFO: Initiating Jersey application, version 'Jersey: 1.19.4 05/24/2017 03:20 PM'
Jun 04, 2024 10:55:40 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.server.nodemanager.webapp.JAXBContextResolver to GuiceManagedComponentProvider with
 the scope "Singleton"
Jun 04, 2024 10:55:40 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.webapp.GenericExceptionHandler to GuiceManagedComponentProvider with the scope "Sin
gleton"
Jun 04, 2024 10:55:40 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.server.nodemanager.webapp.NMWebServices to GuiceManagedComponentProvider with the s
cope "Singleton"
2024-06-04 22:55:40,975 INFO handler.ContextHandler: Started o.e.j.w.WebAppContext@38ed139b{node,/,file:///C:/Users/Bhar
athwaaj/AppData/Local/Temp/jetty-0_0_0_0-8042-hadoop-yarn-common-3_3_6_jar-_-any-4585533884930814233/webapp/,AVAILABLE}{
jar:file:/C:/hadoop/share/hadoop/yarn/hadoop-yarn-common-3.3.6.jar!/webapps/node}
2024-06-04 22:55:40,993 INFO server.AbstractConnector: Started ServerConnector@2af616d3{HTTP/1.1, (http/1.1)}{0.0.0.0:80
42}
2024-06-04 22:55:40,998 INFO server.Server: Started @6374ms
2024-06-04 22:55:40,998 INFO webapp.WebApps: Web app node started at 8042
2024-06-04 22:55:40,998 INFO nodemanager.NodeStatusUpdaterImpl: Node ID assigned is : LAPTOP-8HB989TR:53276
2024-06-04 22:55:41,000 INFO util.JvmPauseMonitor: Starting JVM pause monitor
2024-06-04 22:55:41,010 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8031
2024-06-04 22:55:44,234 INFO ipc.Client: Retrying connect to server: 0.0.0.0/0.0.0.0:8031. Already tried 0 time(s); retr
y policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
```

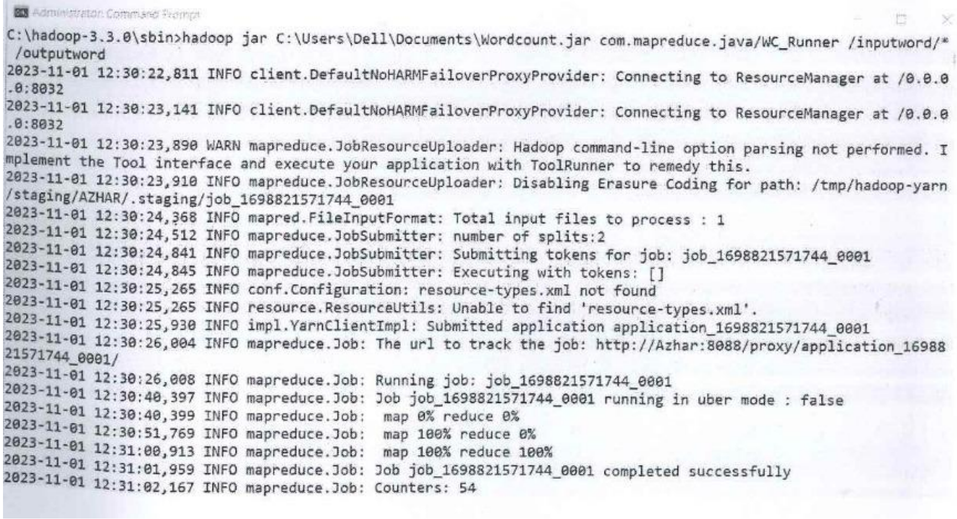
STEP 13: Create a new input directory named as "inputword.
By using the command: **hadoop fs -mkdir inputword**

STEP 14: Now put the "worde.txt" file to the inputword directory.
By using the command: **hadoop fs -put C:\Users\Dell Documents\worde.txt /inputword**



```
C:\hadoop-3.3.0\sbin>hadoop fs -mkdir /inputword

C:\hadoop-3.3.0\sbin>hadoop fs  -put  C:\Users\Dell\Documents\wordc.txt   /inputword

C:\hadoop-3.3.0\sbin>hadoop jar C:\Users\Dell\Documents\Wordcount.jar com.mapreduce.java/WC_Runner /inputword/*
 /outputword
2023-11-01 12:30:22,811 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0
.0:8032
2023-11-01 12:30:23,141 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0
.0:8032
2023-11-01 12:30:23,890 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. I
mplement the Tool interface and execute your application with ToolRunner to remedy this.
2023-11-01 12:30:23,910 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn
/staging/AZHAR/.staging/job_1698821571744_0001
2023-11-01 12:30:24,368 INFO mapred.FileInputFormat: Total input files to process : 1
2023-11-01 12:30:24,512 INFO mapreduce.JobSubmitter: number of splits:2
2023-11-01 12:30:24,841 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1698821571744_0001
2023-11-01 12:30:24,845 INFO mapreduce.JobSubmitter: Executing with tokens: []
2023-11-01 12:30:25,265 INFO conf.Configuration: resource-types.xml not found
2023-11-01 12:30:25,265 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2023-11-01 12:30:25,930 INFO impl.YarnClientImpl: Submitted application application_1698821571744_0001
2023-11-01 12:30:26,004 INFO mapreduce.Job: The url to track the job: http://Azhar:8088/proxy/application_16988
21571744_0001/
2023-11-01 12:30:26,008 INFO mapreduce.Job: Running job: job_1698821571744_0001
2023-11-01 12:30:40,397 INFO mapreduce.Job: Job job_1698821571744_0001 running in uber mode : false
```

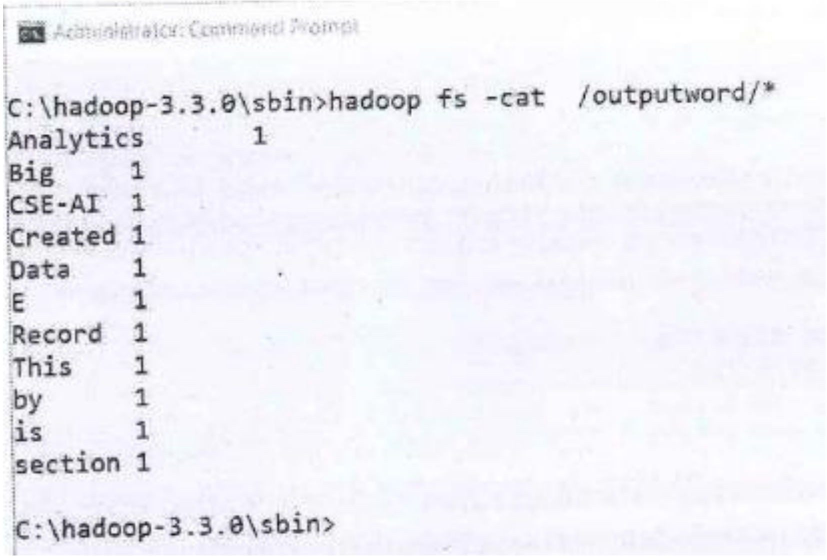
STEP 15: Run the Jar file created from the project
Using the command: **hadoop jar CoUsers/Dell/Documents Wordcount.jar com.mapreduce.java/WC_Runner /inputword//outputword**



```
C:\hadoop-3.3.0\sbin>hadoop jar C:\Users\Dell\Documents\Wordcount.jar com.mapreduce.java/WC_Runner /inputword/*
 /outputword
2023-11-01 12:30:22,811 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0
.0:8032
2023-11-01 12:30:23,141 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0
.0:8032
2023-11-01 12:30:23,890 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. I
mplement the Tool interface and execute your application with ToolRunner to remedy this.
2023-11-01 12:30:23,910 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn
/staging/AZHAR/.staging/job_1698821571744_0001
2023-11-01 12:30:24,368 INFO mapred.FileInputFormat: Total input files to process : 1
2023-11-01 12:30:24,512 INFO mapreduce.JobSubmitter: number of splits:2
2023-11-01 12:30:24,841 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1698821571744_0001
2023-11-01 12:30:24,845 INFO mapreduce.JobSubmitter: Executing with tokens: []
2023-11-01 12:30:25,265 INFO conf.Configuration: resource-types.xml not found
2023-11-01 12:30:25,265 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2023-11-01 12:30:25,930 INFO impl.YarnClientImpl: Submitted application application_1698821571744_0001
2023-11-01 12:30:26,004 INFO mapreduce.Job: The url to track the job: http://Azhar:8088/proxy/application_16988
21571744_0001/
2023-11-01 12:30:26,008 INFO mapreduce.Job: Running job: job_1698821571744_0001
2023-11-01 12:30:40,397 INFO mapreduce.Job: Job job_1698821571744_0001 running in uber mode : false
2023-11-01 12:30:40,399 INFO mapreduce.Job:  map 0% reduce 0%
2023-11-01 12:30:51,769 INFO mapreduce.Job:  map 100% reduce 0%
2023-11-01 12:31:00,913 INFO mapreduce.Job:  map 100% reduce 100%
2023-11-01 12:31:01,959 INFO mapreduce.Job: Job job_1698821571744_0001 completed successfully
2023-11-01 12:31:02,167 INFO mapreduce.Job: Counters: 54
```

STEP 16: At last Print your output for the WordCount text file.
Using the Command: **hadoop fs-cat /outputword/**

OUTPUT:



```
C:\hadoop-3.3.0\sbin>hadoop fs -cat  /outputword/*
Analytics       1
Big     1
CSE-AI  1
Created 1
Data    1
E       1
Record  1
This    1
by      1
is      1
section 1

C:\hadoop-3.3.0\sbin>
```

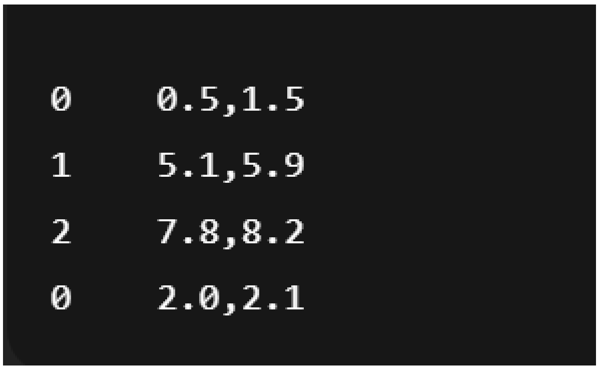
EXP NO:05 **DATE:**

IMPLEMENTATION OF K-MEANS CLUSTERING USING MAP REDUCE

PROGRAM :

```
Package it.unipi.hadoop.mapreduce;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import it.unipi.hadoop.model.Point;
public class KMeansMapper extends Mapper<LongWritable, Text, IntWritable, Point> {
private Point[] centroids;
private int p;
private final Point point = new Point();
private final IntWritable centroid = new IntWritable();
public void setup(Context context) {
int k = Integer.parseInt(context.getConfiguration().get("k"));
this.p = Integer.parseInt(context.getConfiguration().get("distance"));
this.centroids = new Point[k];
for(int i = 0; i < k; i++) {
String[] centroid = context.getConfiguration().getStrings("centroid." + i);
this.centroids[i] = new Point(centroid);
}
}
public void map(LongWritable key, Text value, Context context)
throws IOException, InterruptedException {
// Contruct the point
String[] pointString = value.toString().split(",");
point.set(pointString);
// Initialize variables
float minDist = Float.POSITIVE_INFINITY;
float distance = 0.0f;
int nearest = -1;

// Find the closest centroid

for (int i = 0; i < centroids.length; i++) {

distance = point.distance(centroids[i], p);
if(distance < minDist) {

nearest = i; minDist = distance;

}
}

centroid.set(nearest);
context.write(centroid, point);
}
}
```

OUTPUT :



```
0    0.5,1.5
1    5.1,5.9
2    7.8,8.2
0    2.0,2.1
```

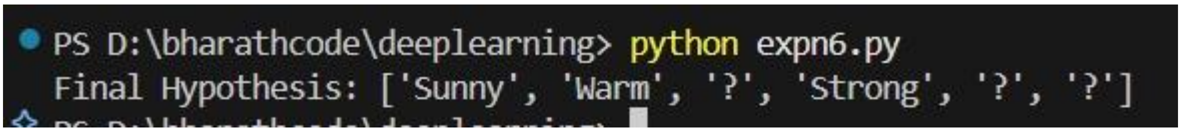
EXP NO:06 **DATE:**

Implement and demonstrate the FIND-S Algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a CSV file.

PROGRAM :

```
import pandas as pd

def find_s_algorithm(filename):
    data = pd.read_csv("D:\\bharathcode\\deeplearning\\data.csv")

    attributes = data.iloc[:, :-1].values     # all columns except last
    target = data.iloc[:, -1].values           # last column

       hypothesis = ['0'] * (attributes.shape[1])

    for i, val in enumerate(attributes):
        if target[i].lower() == "yes":   # positive example
            for j in range(len(hypothesis)):
                if hypothesis[j] == '0':
                    hypothesis[j] = val[j]
                elif hypothesis[j] != val[j]:
                    hypothesis[j] = '?'

    return hypothesis


if __name__ == "__main__":
    final_hypothesis = find_s_algorithm("training_data.csv")
    print("Final Hypothesis:", final_hypothesis)
```

OUTPUT:



```
PS D:\bharathcode\deeplearning> python expn6.py
Final Hypothesis: ['Sunny', 'Warm', '?', 'Strong', '?', '?']
PS D:\bharathcode\deeplearning>
```

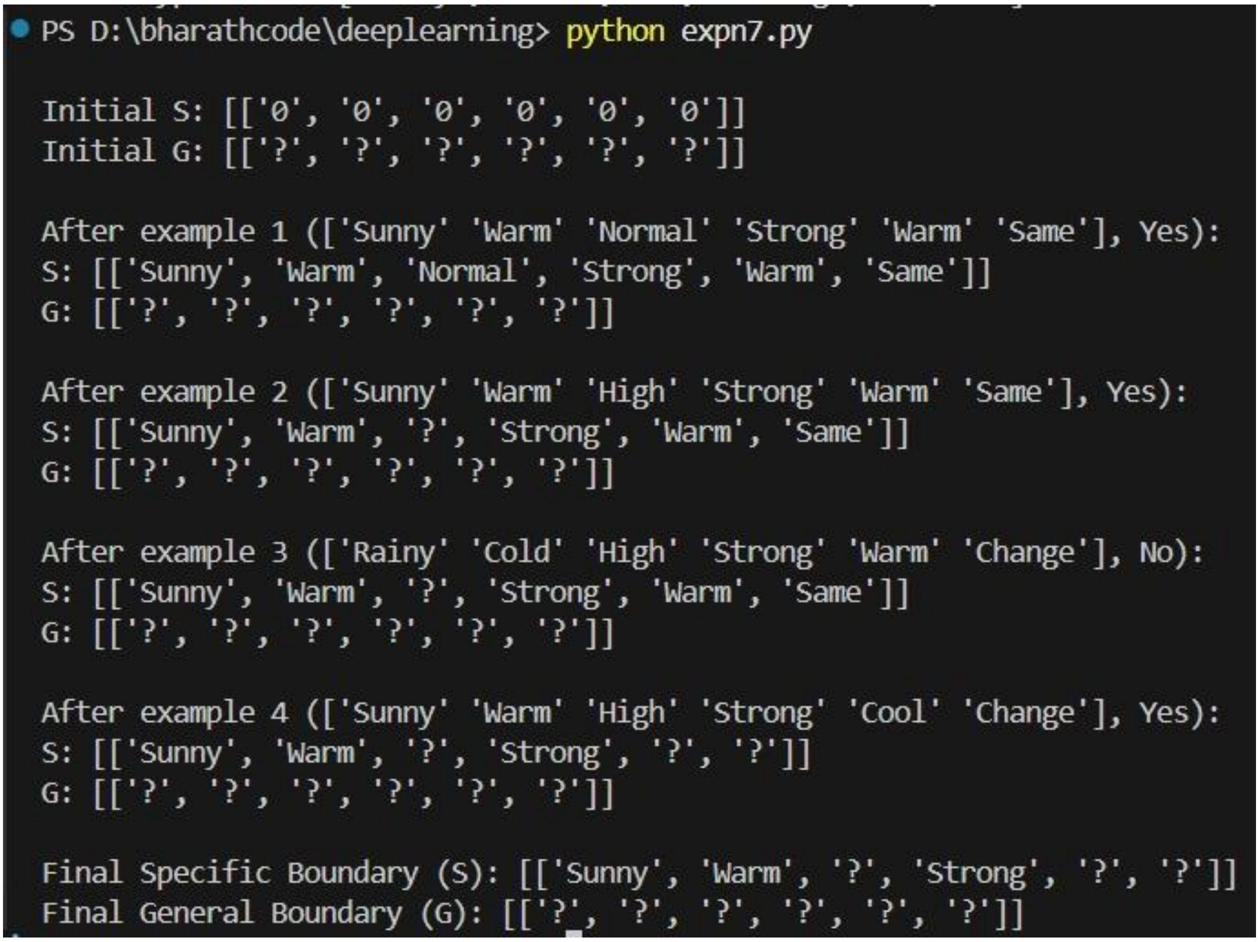
EXP NO:07 **DATE:**

For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples.

PROGRAM :

```
import pandas as pd

def candidate_elimination(filename):
    data = pd.read_csv("D:\\bharathcode\\deeplearning\\data.csv")
    concepts = data.iloc[:, :-1].values
    target = data.iloc[:, -1].values

    S = [['0'] * len(concepts[0])]
    G = [['?'] * len(concepts[0])]

    print("\nInitial S:", S)
    print("Initial G:", G)

    for i, h in enumerate(concepts):
        if target[i].lower() == "yes":
            G = [g for g in G if all(g[j] in ['?', h[j]] for j in range(len(h)))]

            for j in range(len(S[0])):
                if S[0][j] == '0':
                    S[0][j] = h[j]
                elif S[0][j] != h[j]:
                    S[0][j] = '?'

        else:
            if all(S[0][j] in ['?', h[j]] for j in range(len(h))):
                # Specialize G
                new_G = []
                for j in range(len(S[0])):
                    if S[0][j] == '?':
                        new_hypothesis = S[0].copy()
                        new_hypothesis[j] = h[j] + "_not"   # mark specialization
                        new_G.append(new_hypothesis)
                G.extend(new_G)

        print(f"\nAfter example {i+1} ({h}, {target[i]}):")
        print("S:", S)
        print("G:", G)
     return S, G

if __name__ == "__main__":
    S_final, G_final = candidate_elimination("training_data.csv")
    print("\nFinal Specific Boundary (S):", S_final)
    print("Final General Boundary (G):", G_final)
```

OUTPUT:



```
PS D:\bharathcode\deeplearning> python expn7.py

Initial S: [['0', '0', '0', '0', '0', '0']]
Initial G: [['?', '?', '?', '?', '?', '?']]

After example 1 (['Sunny' 'Warm' 'Normal' 'Strong' 'Warm' 'Same'], Yes):
S: [['Sunny', 'Warm', 'Normal', 'Strong', 'Warm', 'Same']]
G: [['?', '?', '?', '?', '?', '?']]

After example 2 (['Sunny' 'Warm' 'High' 'Strong' 'Warm' 'Same'], Yes):
S: [['Sunny', 'Warm', '?', 'Strong', 'Warm', 'Same']]
G: [['?', '?', '?', '?', '?', '?']]

After example 3 (['Rainy' 'Cold' 'High' 'Strong' 'Warm' 'Change'], No):
S: [['Sunny', 'Warm', '?', 'Strong', 'Warm', 'Same']]
G: [['?', '?', '?', '?', '?', '?']]

After example 4 (['Sunny' 'Warm' 'High' 'Strong' 'Cool' 'Change'], Yes):
S: [['Sunny', 'Warm', '?', 'Strong', '?', '?']]
G: [['?', '?', '?', '?', '?', '?']]

Final Specific Boundary (S): [['Sunny', 'Warm', '?', 'Strong', '?', '?']]
Final General Boundary (G): [['?', '?', '?', '?', '?', '?']]
```

EXP NO:08 **DATE:**

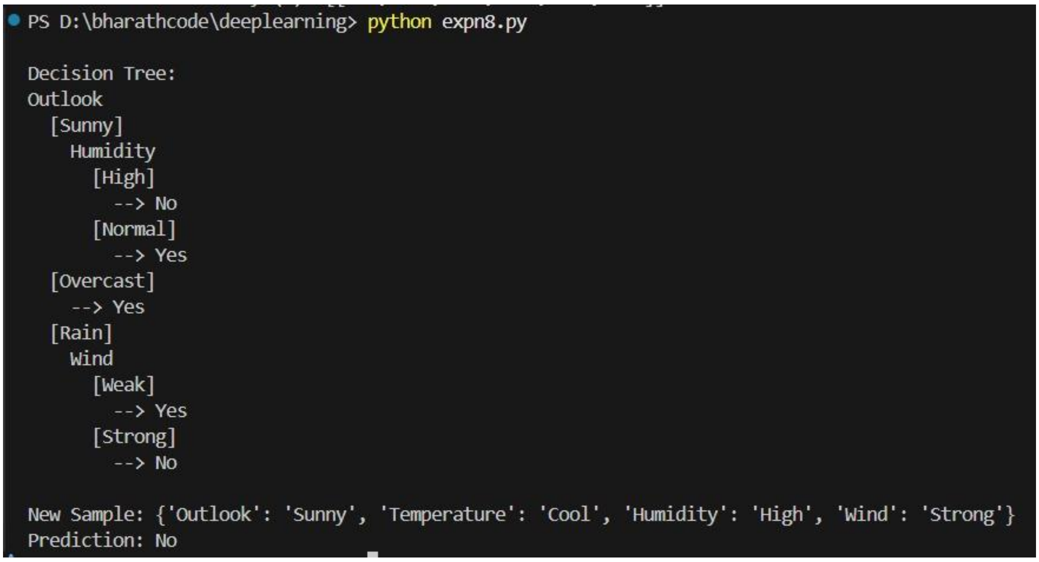
Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.

PROGRAM :

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from collections import Counter
import math

def entropy(y):
    counter = Counter(y)
    total = len(y)
    return -sum((count/total) * math.log2(count/total) for count in counter.values())

def info_gain(data, feature, target):
    total_entropy = entropy(data[target])
    values = data[feature].unique()
    weighted_entropy = 0
    for v in values:
        subset = data[data[feature] == v]
        weighted_entropy += (len(subset)/len(data)) * entropy(subset[target])
    return total_entropy - weighted_entropy

def id3(data, features, target):
    if len(set(data[target])) == 1:
        return list(data[target])[0]
    if len(features) == 0:
        return Counter(data[target]).most_common(1)[0][0]
    gains = [info_gain(data, f, target) for f in features]
    best_feature = features[np.argmax(gains)]
    tree = {best_feature: {}}
    remaining_features = [f for f in features if f != best_feature]
    for value in data[best_feature].unique():
        subset = data[data[best_feature] == value]
        subtree = id3(subset, remaining_features, target)
        tree[best_feature][value] = subtree
    return tree

def predict(tree, sample):
    if not isinstance(tree, dict):   # Leaf node
        return tree
    root = next(iter(tree))
    value = sample.get(root)
    if value in tree[root]:
        return predict(tree[root][value], sample)
    else:
        return None
```

```
def plot_tree(tree, depth=0, indent="   "):
    if not isinstance(tree, dict):
        print(indent * depth + f"--> {tree}")
        return
    for key, value in tree.items():
        print(indent * depth + str(key))
        for k in value:
            print(indent * (depth+1) + f"[{k}]")
            plot_tree(value[k], depth+2, indent)
if __name__ == "__main__":
    data = pd.read_csv("D:\\bharathcode\\deeplearning\\tennis.csv")
    target = 'PlayTennis'
    features = list(data.columns[:-1])
    decision_tree = id3(data, features, target)
    print("\nDecision Tree:")
    plot_tree(decision_tree)
    new_sample = {'Outlook': 'Sunny', 'Temperature': 'Cool', 'Humidity': 'High', 'Wind':
'Strong'}
    prediction = predict(decision_tree, new_sample)
    print("\nNew Sample:", new_sample)
     print("Prediction:", prediction)
```

OUTPUT :



```
PS D:\bharathcode\deeplearning> python expn8.py

Decision Tree:
Outlook
  [Sunny]
    Humidity
      [High]
        --> No
      [Normal]
        --> Yes
  [Overcast]
    --> Yes
  [Rain]
    Wind
      [Weak]
        --> Yes
      [Strong]
        --> No

New Sample: {'Outlook': 'Sunny', 'Temperature': 'Cool', 'Humidity': 'High', 'Wind': 'Strong'}
Prediction: No
```

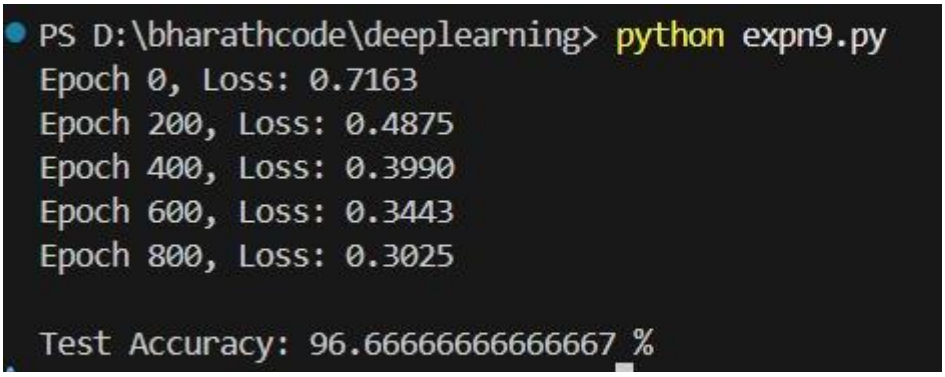
EXP NO:09 **DATE:**

Build an Artificial Neural Network by implementing the Backpropagation algorithm and test the same using appropriate data sets

PROGRAM :

```
import numpy as np
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder, StandardScaler

def sigmoid(x): return 1 / (1 + np.exp(-x))
def dsigmoid(x): return x * (1 - x)
def softmax(x):
    exp = np.exp(x - np.max(x, axis=1, keepdims=True))
    return exp / np.sum(exp, axis=1, keepdims=True)

iris = load_iris()
X = iris.data
y = iris.target.reshape(-1, 1)
enc = OneHotEncoder(sparse_output=False)
y = enc.fit_transform(y)

scaler = StandardScaler()
X = scaler.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

np.random.seed(1)
W1 = np.random.randn(4, 5)    # 4 inputs → 5 hidden neurons
b1 = np.zeros((1, 5))
W2 = np.random.randn(5, 3)    # 5 hidden → 3 output classes
b2 = np.zeros((1, 3))

lr = 0.05
for epoch in range(1000):
    z1 = X_train @ W1 + b1
    a1 = sigmoid(z1)
    z2 = a1 @ W2 + b2
    a2 = softmax(z2)

    loss = -np.mean(np.sum(y_train * np.log(a2 + 1e-8), axis=1))

    d2 = a2 - y_train
    dW2 = a1.T @ d2 / len(X_train)
    db2 = np.mean(d2, axis=0, keepdims=True)
    d1 = (d2 @ W2.T) * dsigmoid(a1)
    dW1 = X_train.T @ d1 / len(X_train)
    db1 = np.mean(d1, axis=0, keepdims=True)

    W1 -= lr * dW1
    b1 -= lr * db1
    W2 -= lr * dW2
    b2 -= lr * db2
    if epoch % 200 == 0:
        print(f"Epoch {epoch}, Loss: {loss:.4f}")
z1 = X_test @ W1 + b1
a1 = sigmoid(z1)
z2 = a1 @ W2 + b2
a2 = softmax(z2)

preds = np.argmax(a2, axis=1)
true = np.argmax(y_test, axis=1)
acc = np.mean(preds == true)
print("\nTest Accuracy:", acc * 100, "%")
```

OUTPUT :



```
PS D:\bharathcode\deeplearning> python expn9.py
Epoch 0, Loss: 0.7163
Epoch 200, Loss: 0.4875
Epoch 400, Loss: 0.3990
Epoch 600, Loss: 0.3443
Epoch 800, Loss: 0.3025

Test Accuracy: 96.66666666666667 %
```

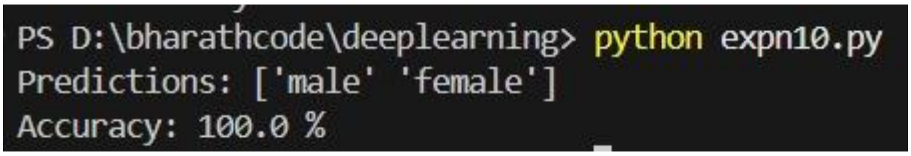
EXP NO:10 **DATE:**

Write a program to implement the naive Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.

PROGRAM :

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score
data = pd.read_csv("data1.csv")

X = data.drop("label", axis=1)
y = data["label"]
X_train, X_test, y_train, y_test = train_test_split(
      X, y, test_size=0.2, random_state=42
)

nb = GaussianNB()
nb.fit(X_train, y_train)

y_pred = nb.predict(X_test)
acc = accuracy_score(y_test, y_pred)
print("Predictions:", y_pred)
print("Accuracy:", acc * 100, "%")
```

OUTPUT :



```
PS D:\bharathcode\deeplearning> python expn10.py
Predictions: ['male' 'female']
Accuracy: 100.0 %
```