Task 1: Hadoop Service Startup and HDFS Interaction

Service Startup:

Start all essential Hadoop services: NameNode, DataNode, ResourceManager, NodeManager, and SecondaryNameNode.
Provide the command line instructions used to start each service.

Service Verification:

Use the `jps` command to list and verify that all core Hadoop services are running.
Capture a screenshot of the terminal output showing the running services and describe the function of each service:

- **NameNode**: Manages the HDFS filesystem namespace.
- **DataNode**: Stores the actual data in HDFS.
- **ResourceManager**: Manages resource allocation for MapReduce jobs.
- **NodeManager**: Manages the execution of containers on worker nodes.
- **SecondaryNameNode**: Periodically checkpoints the HDFS metadata.

Hadoop Service Testing via Browser:

Use the following URLs to verify the status of your Hadoop services through a web browser:

1. **NameNode Web UI**: `http://<namenode_host>:50070/`
2. **ResourceManager Web UI**: `http://<resourcemanager_host>:8088/`
3. **JobHistory Server Web UI**: `http://<historyserver_host>:10020/`
4. **YARN NodeManager Web UI**: `http://<nodemanager_host>:8042/`

Replace `<namenode_host>`, `<resourcemanager_host>`, and `<nodemanager_host>` with the IP address or hostname of the respective services. For a single-node cluster, these will typically be `localhost`.

HDFS Interaction:

- Create a directory in HDFS using the `hadoop fs -mkdir` command.
- Upload a small text file to HDFS with the `hadoop fs -put` command.
- List the files in the directory using the `hadoop fs -ls` command.
  Provide the commands used, and include screenshots of the terminal output showing the results.

Task 2: Implement Word Count Using MapReduce

Write a MapReduce program in Java to count the occurrences of each word in a given document. The program should map each word to a key-value pair and reduce the data by summing the counts. Run the job on your Hadoop cluster and submit the Java code, a sample input file, the output of the job, and a brief description of the steps taken.

Task 3: Implement Line Count Using MapReduce

Write a MapReduce program in Java to count the number of lines in a text document. The program should map each line to a key-value pair with the line count (1 for each line) and reduce the data by summing the counts. Run the job on your Hadoop cluster and submit the Java code, a sample input file, the output of the job, and a brief description of the steps taken.