Question Bank(Module 3 & module 4)

1. Explain the two phases of the MapReduce framework and their roles in processing the weather data with diagram.
2. How does data flow from the map phase to the reduce phase with diagram?
3. Discuss the reasons for storing intermediate map output on local disk rather than HDFS.
4. How does Hadoop handle the failure of a node during map task execution?
5. Why is it beneficial for Hadoop to run map tasks on nodes where the input data resides?
6. What are the potential data locality scenarios for a map task?
7. Explain how the reduce function in the MaxTemperatureReducer class determines the maximum temperature for each year.
8. What types are used for the input and output of the reduce function?
9. Define scaling out. Explain map reduce work flow with multiple reduce tasks.
10. Define scaling out. Explain map reduce work flow with single reduce tasks.
11. Develop a Mapper java code for the maximum temperature example
12. Develop a Reducer java code for the maximum temperature example
13. What are the key design principles of the Hadoop Distributed File System (HDFS)?
14. What is considered a "very large" file in the context of HDFS, and how does HDFS manage such files?
15. How does the namenode's memory management limit the number of files that can be stored in HDFS, and what are the consequences of this limitation?
16. What challenges does HDFS face when handling lots of small files, and how can these challenges be mitigated?
17. Explain the following :
    1. Blocks
    2. Namenodes and Datanodes
    3. HDFS Federation
    4. HDFS High-Availability
    5. Failover Controller
    6. Fencing
18. What is the role of the failover controller in the HDFS high availability (HA) architecture?
19. How does the failover controller use ZooKeeper to manage the transition between active and standby namenodes in HDFS?
20. What is the heartbeating mechanism in the failover controller, and how does it help in monitoring the namenode’s health?
21. Explain Anatomy of a File Read with a neat diagram.
22. Explain Anatomy of a File write with a neat diagram.
23. What is the purpose of a combiner function in a MapReduce job, and how does it optimize data processing?
24. In the context of the maximum temperature example, how does using a combiner function affect the data flow between the map and reduce tasks?

Module – 4

1. Explain the architecture of a spark application with a diagram.
2. What are the differences between SparkR and sparklyr, and in what scenarios would each be used for integrating R with Spark?
3. What are Spark’s language APIs, and how do they enable the use of various programming languages with Spark?
4. What is a DataFrame in Spark, and how does it differ from a traditional table or spreadsheet?
5. Explain dataframes in spark with a neat diagram.
6. What is a partition in Spark, and how does it relate to the parallelism of data processing in a cluster?
7. How does the number of partitions affect the parallelism of a Spark job, and what happens if there are more partitions than executors?
8. What are transformations in Spark, and how do they allow you to modify DataFrames despite their immutability?
9. What is the difference between narrow and wide transformations in Spark, and how do these differences impact data processing?
10. How does the concept of lazy evaluation in Spark relate to the execution of transformations, and what are the implications for performance tuning?
11. What is lazy evaluation in Spark, and how does it differ from immediate execution of operations?
12. How does lazy evaluation help Spark optimize the execution of transformations across a cluster?
13. How does lazy evaluation enable Spark to perform optimizations like predicate pushdown?
14. Explain structured API Execution with diagram.