1. Explain the importance of below 4 demons in job execution with minimum of 5 points

**• Name node**

1. NameNode is the Master Node in Hadoop File System.It only stores the meta data of HDFS---contains the directory tree of all the files in file system and track the files across the cluster.
2. NameNode knows the list of the blocks and its location for any given file in HDFS. With this information NameNode knows how to construct the file from blocks.
3. NameNode is so critical to HDFS and when the NameNode is down, HDFS/Hadoop cluster is inaccessible and considered down
4. NameNode is a single point of failure in Hadoop cluster.
5. NameNode is usually configured with a lot of memory (RAM). Because the block locations are help in main memory.

**• Data node**

1. DataNode is responsible for storing the actual data in HDFS.
2. Datanode is the Slave node in HDFS.
3. NameNode and DataNode are in constant communication.
4. When a DataNode starts up it announce itself to the NameNode along with the list of blocks it is responsible for.
5. When a DataNode is down, it does not affect the availability of data or the cluster. NameNode will arrange for replication for the blocks managed by the DataNode that is not available.

• Resource Manager

1. **ResourceManager (RM)** is the master per cluster that arbitrates all the available cluster resources and thus helps manage the distributed applications running on the YARN system.
2. Resource Manager works together with the per-node **NodeManagers (NMs)** and the per-application **ApplicationMasters (AMs)**.
3. Resource Manager knows where the slaves are located(Rack Awareness) and how many resources they have.
4. Resource Manager runs several services, the most important is the Resource Scheduler which decides how to assign the resources.
5. Resource Manager is the single point of failure in YARN Architecture.

**• Node manager**

1. **Node Manager** (many per cluster) is the slave of the Yarn infrastructure.
2. **When it starts, it announces himself to Resource Manager.**
3. Each Node Manager offers some resources to the cluster ie. The amount of memory and number of vcores.
4. Each Node Manager tracks the available data processing resources on its slave node and sends regular reports to the Resource Manager.
5. Node Manager can scale much more easily and its code is much more maintainable.