1) Pen down the limitations of MapReduce.
Ans--1) Hadoop MapReduce, read and write from the disk as a result it slows down the computation.
2)In MapReduce, developers need to hand code each and every operation which makes it very difficult to work.
3)MapReduce fails when it comes to real-time data processing as it was designed to perform batch processing on voluminous amounts of data.
4)As MapReduce only provides the batch engine. Hence, we are dependent on different engines. For example-
Storm, Giraph, Impala, etc. for other requirements. So, it is very difficult to manage many components.
5)MapReduce needs an external job scheduler for example, Oozie to schedule complex flows.

2) What is RDD? Explain few features of RDD?
Ans---RDD (Resilient Distributed Dataset) is the fundamental data structure of Apache Spark which are an immutable collection of objects which computes on the different node of the cluster. Each and every dataset in Spark RDD is logically partitioned across many servers so that they can be computed on different nodes of the cluster.
Features of RDD is as follows,
1.In-memory computation
The data inside RDD are stored in memory for as long as you want to store. Keeping the data in-memory improves the performance by an order of magnitudes.

2.Lazy Evaluation
The data inside RDDs are not evaluated on the go. The changes or the computation is performed only after an action is triggered. Thus, it limits how much work it has to do.
3.Fault Tolerance
Upon the failure of worker node, using lineage of operations we can re-compute the lost partition of RDD from the original one. Thus, we can easily recover the lost data.
4.Persistence
We can store the frequently used RDD in in-memory and we can also retrieve them directly from memory without going to disk, this speedup the execution. We can perform Multiple operations on the same data, this happens by storing the data explicitly in memory by calling persist() or cache() function.
5.Partitioning
RDD partition the records logically and distributes the data across various nodes in the cluster. The logical divisions are only for processing and internally it has no division. Thus, it provides parallelism.

3)List down few Spark RDD operations and explain each of them.
Ans----1.
val x = sc.parallelize(List("spark rdd example", "sample example"))--This creates RDD.
x.collect()--This is Action on RDD.

2.
val textFileLocalTest = sc.textFile("/Users/acadgild/test.txt") ---This creates RDD from File.

3.
val x = sc.parallelize(List("spark rdd example", "sample example"))
val y = x.flatMap(x => x.split(" ")) ---flatMap will remove the expression/separator in split function and return the combined result.

4.
val x = sc.parallelize(1 to 10, 2)--Creates RDD with 2 Partition.

5.
val y = x.filter(num => num%2==0)---Filters/returns numbers divisible by 2 only.

-------------------*******END***********---------------------