1l What is the difference between TextInputFormat and KeyValueInputFormat class?

The TextInputFormat class converts every row of the source file into key/value types where the BytesWritable key represents the offset of the record and the Text value represents the entire record itself.

The KeyValueTextInputFormat is an extended version of TextInputFormat , which is useful when we have to fetch every source record as Text/Text pair where the key/value were populated from the record by splitting the record with a fixed delimiter.

**TextInputFormat:**  
1) It reads lines of text files and provides the offset of the line as key to the Mapper and actual line as Value to the mapper.  
2) TextInputFormat is the default file format in Hadoop .  
  
**KeyValueInputFormat:**  
1) Reads text file and parses lines into key, Val pairs. Everything up to the first tab character is sent as key to the Mapper and the remainder of the line is sent as value to the mapper.

2. How is the splitting of file invoked in Hadoop framework?

 It is invoked by the Hadoop framework by running getInputSplit()method of the Input format class (like FileInputFormat) defined by the user.

3. Consider case scenario: In M/R system, - HDFS block size is 64 MB - Input format is FileInputFormat – We have 3 files of size 64K, 65Mb and 127Mb How many input splits will be made by Hadoop framework for each file?  
Hadoop will make 5 splits as follows   
- 1 split for 64K files   
- 2  splits for 65Mb files   
- 2 splits for 127Mb file

4. After the Map phase finishes, the Hadoop framework performs “Partitioning, Shuffle and sort”. Explain each event in brief.

**- Partitioning**  
Partitioning is the process of determining which reducer instance will receive which intermediate keys and values. Each mapper must determine for all of its output (key, value) pairs which reducer will receive them. It is necessary that for any key, regardless of which mapper instance generated it, the destination partition is the same  
  
**- Shuffle**  
After the first map tasks have completed, the nodes may still be performing several more map tasks each. But they also begin exchanging the intermediate outputs from the map tasks to where they are required by the reducers. This process of moving map outputs to the reducers is known as shuffling.  
  
**- Sort**  
Each reduce task is responsible for reducing the values associated with several intermediate keys. The set of intermediate keys on a single node is automatically sorted by Hadoop before they are presented to the Reducer

5 What is a Combiner?

The Combiner is a "mini-reduce" process which operates only on data generated by a mapper. The Combiner will receive as input all data emitted by the Mapper instances on a given node. The output from the Combiner is then sent to the Reducers, instead of the output from the Mappers.

6. What is Hadoop streaming?

Streaming is a generic API that allows programs written in virtually any language to be used as Hadoop Mapper and Reducer implementations

 7. What are the most commonly defined input formats in Hadoop and explain each in brief.

These are the most common input formats defined in Hadoop:

1. TextInputFormat
2. KeyValueInputFormat
3. SequenceFileInputFormat

Default input format in Hadoop is TextInputFormat.

8. Explain what is distributed Cache in MapReduce Framework ?

Distributed Cache is a facility provided by the Map/Reduce framework to cache files (text, archives, jars and so on) needed by applications during execution of the job. The framework will copy the necessary files to the slave node before any tasks for the job are executed on that node.

9. Explain what happens in textinputformat ?

In textinputformat, each line in the text file is a record. Key is the byte offset of the line and value is the content of the line. For instance, **Key: longWritable, value: text**.

10.Explain what is Sequencefileinputformat?

Sequencefileinputformatis an input format for reading in sequence files. Key and value are user defined. It is a specific compressed binary file format which is optimized for passing the data between the output of one MapReduce job to the input of some other MapReduce job.