**Answer1:**

**TextInputFormat:** 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.

**KeyValueInputFormat:** 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.

**Answer2:**

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

**Answer3:**

Hadoop will make 5 splits as follows:

- 1 split for 64K files

- 2 splits for 65MB files

- 2 splits for 127MB files

**Answer4:**

**Partitioning:** It 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.

**Answer5:**

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 o4n a given node. The output from the Combiner is then sent to the Reducers, instead of the output from the Mappers.

**Answer6:**

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

**Answer7:**

 The most common Input Formats defined in Hadoop are:

* **TextInputFormat:** 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.
* **KeyValueInputFormat:** 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
* **SequenceFileInputFormat: Reads sequence file and stores sequences of binary key-value pairs.**

**Answer8:**

Distributed Cache is a facility provided by the MapReduce 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.

**Answer9:**

In text input format, each line will create a line object, that is an hexa-decimal number. Key is considered as a line object and value is considered as a whole line text. This is how the data gets processed by a mapper. The mapper will receive the ‘key’ as a ‘LongWritable‘ parameter and value as a ‘text‘ parameter.

**Answer10:**

Sequencefileinputformat is used for reading files in sequence. It is a specific compressed binary file format which is optimized for passing data between the output of one MapReduce job to the input of some other MapReduce job.