PART – A (5 x 6 = 30)

Following some codes are given – you have to write what does they do? Please use **one or two** line(s) to write what the code segments does.
(For example, you can say – "this code reverses a string". Or you can say – "this code sorts a set of integers" For some questions you might be asked to give sample input and output).
PLEASE DO NOT write that "this code will give error" as we are more concern about the logical flow here, NOT the syntax.

1. The map and reduce functions of a Hadoop job are given. What does the job performs?

```
public static class Map extends Mapper<LongWritable, Text, LongWritable, Text> {
    int s;
    public void map(LongWritable key, Text value, Context context)
    throws IOException, InterruptedException {
        String line = value.toString();
        StringTokenizer tokenizer = new StringTokenizer(line);
        String op = "";
        while (tokenizer.hasMoreTokens()){
            s = Integer.parseInt(tokenizer.nextToken());
            s = s*s;
            op+=s+" ";
        }
        context.write(key,new Text(op)) ;
    }
}
public static class Reduce extends Reducer<LongWritable, Text, LongWritable, IntWritable>{
    public void reduce(LongWritable key, Iterable<Text> values, Context context)
    throws IOException, InterruptedException {
        int x = 0;
        for (Text val : values) {
            String line = val.toString();
            StringTokenizer tokenizer = new StringTokenizer(line);
            while( tokenizer.hasMoreTokens()){
                int num = Integer.parseInt( tokenizer.nextToken() ) ;
                x += num;
            }
        }
        context.write(key, new IntWritable(x));
    }
}
```

Answer: The code is calculating the sum of the squares of integers in the file.

2. If the following map and reduce function works on the movies.dat (the data file from your HWs) file then what is expected to be produced?

```
public static class Map extends Mapper<LongWritable, Text, Text, IntWritable> {
        private static final IntWritable one = new IntWritable(1);
        private Text word = new Text();
        private String[] entry, genres;

        public void map(LongWritable key, Text value, Context context)
        throws IOException, InterruptedException {
                String line = value.toString();
                entry = line.split("::");
                genres = entry[2].split("\\|");
                for(String genre : genres){
                        word.set(genre);
                        context.write(word,one);
                }
        }
}

public static class Reduce extends Reducer<Text, IntWritable, Text, IntWritable> {
    public void reduce(Text key, Iterable<IntWritable> values, Context context)
    throws IOException, InterruptedException {
            int sum=0;
            for(IntWritable v : values){
                    sum = sum + v.get();
            }
            context.write(key, new IntWritable(sum));
    }
}
```

entry = 1, Toy, Animation | Action
genres = Action, Animation

word.set(Action)
context.write(Action, 1)
context.write(Adventure, 1)
context.write(Adventure, 1)

Answer ::

It is calculating the count of movies falling into each genre

Example:

Animation, 5 [5 represents, that there are 5 movies with genre Animation]

Action, 7 [7 represents, that there are 7 movies with genre Action]

3. If the following map and reduce function works on the movies.dat (the data file from your HWs) file then what is expected to be produced? The program takes user input. Please write a sample user input and sample output of the program.

```
public static class Map extends Mapper<LongWritable, Text, NullWritable, Text> {
      private String cmdParam = "";
      private String[] inputParams;
      private String[] entry, genres;

      public void setup(Context context){
            Configuration conf = context.getConfiguration();
            cmdParam = conf.get("params");
      }
      public void map(LongWritable key, Text value, Context context)
      throws IOException, InterruptedException {
            inputParams = cmdParam.split(",");
            entry = value.toString().split("::");
            if(entry.length>2) {
                  int countMatch=0;
                  for(String eachInput : inputParams){
                        if(entry[2].matches(".*"+eachInput+".*"))
                              countMatch++;
                        if(countMatch==inputParams.length)
                              context.write(NullWritable.get(),new Text(entry[1]));
                  }
            }
      }
}

public static class Reduce extends Reducer<NullWritable, Text, NullWritable, Text> {
      public void reduce(NullWritable key, Iterable<Text> values, Context context)
      throws IOException, InterruptedException {
            for(Text m : values) {
                  context.write(key, m);
            }
      }
}
```

some Genre entered. ex: Action, Adventure

Action | Adventure | Comedy

Answer: It outputs all those movies whose genre list at least has all the genres entered by user at command line

ex:

movies.dat

1::Toy Story (1995)::Animation|Children's|Comedy
2::Jumanji (1995)::Adventure|Children's|Fantasy
3::Grumpier Old Men (1995)::Comedy|Romance

User Input: Adventure, Children's

output: Toy Story (1995)
Jumanji (1995)

4. Following are a Mapper class and a Reducer class. What do you think – what are they doing? Please write in very brief and give sample input it can take and give the sample output for your input.

```
public class TheMapper extends Mapper<LongWritable, Text, IntWritable, Text>{
   String[] tokens,components;
   String outputValue="";
   private Text keyMap = new Text();
   Text valueMap = new Text();

   protected void map(LongWritable key, Text value, Context context)
   throws IOException, InterruptedException {
         tokens = value.toString().split("#");
         components = tokens[1].split(",");
         for(int i=0;i<components.length;i++) {
               keyMap.set(String.valueOf(i));
               outputValue=tokens[0]+","+components[i];
               valueMap.set(outputValue);
               context.write(new IntWritable(i),valueMap);
         }
   }
}

public class TheReducer extends Reducer<IntWritable, Text, NullWritable, Text>{
   TreeMap<Integer, String> tm;
   String reduceValue="";
   String[] tokens;
   Text valueReduce = new Text();

   public void reduce(IntWritable key, Iterable<Text> values, Context context)
   throws IOException, InterruptedException {
      tm = new TreeMap<Integer,String>();
      for(Text value : values) {
            tokens = value.toString().split(",");
            tm.put(Integer.parseInt(tokens[0]), tokens[1]);
      }
      reduceValue="";
      for(String value : tm.values()) {
            if(reduceValue.equals(""))
                  reduceValue=value;
            else
                  reduceValue=reduceValue+","+value;
      }
      valueReduce.set(reduceValue);
      context.write(NullWritable.get(), valueReduce);
      tm.clear();
   }
}
```

Answer: The code is concatenating strings column wise starting after the "#".

Input:
A # First, Perfect
B # Second, Very Good
C # Third, Good.

Output:
First, Second, Third
Perfect, Very Good, Good

5. What kind of join is implement here between movies and ratings in the following pig code ? (The dataset is same as your homework)

```
M = LOAD '/HW_3_Data/movies' USING PigStorage(';') AS
                    (MovieID:int,Title:chararray,Genres:chararray);
R = LOAD '/HW_3_Data/ratings' USING PigStorage(';') AS
                    (UserID:int,MovieID:int,Rating:int,Timestamp:int);
C = COGROUP M by MovieID, R by MovieID;
J = foreach C generate flatten(M), flatten(R);
```

Answer:

The join implemented is inner join, which is one to many.

PART – B (10 x 4 = 40)

For each of the questions in this part – you would be given 5 choices. Please mark **ONLY ONE** choice.
If you think there is no correct choice – please mark the closet possible choice.

1. Hadoop creates one map task for _____
   a) each input file
   b) each node
   c) each split
   d) each logical unit
   e) each computational unit

2. The relation of Reduce task and data locality
   a) Reduce task has the advantage of data locality
   b) Reduce task has not the advantage of data locality
   c) Reduce task has the advantage of data locality if the map has that advantage
   d) Depends on data location
   e) Depends on number of reducers

3. During shuffle and sorting phase
   a) Sorting takes place only in map side
   b) Sorting takes place only in reduce side
   c) Sorting takes place only in both map and reduce side
   d) Sorting takes place before partitioning
   e) Sorting takes place after shuffle

4. Which scenario is not true :
   a) Tuples with same keys go to same reducer
   b) Tuples with same keys go to same mapper
   c) Tuples with same keys go to different reducer
   d) Tuples with same keys go to different mapper
   e) Tuples with different keys go to different mapper

5. Generally speaking, which one is true in terms of time consumption?
   a) In-memory join > map-side join > reduce-side join
   b) In-memory join < map-side join < reduce-side join
   c) In-memory join > map-side join < reduce-side join
   d) In-memory join = map-side join > reduce-side join
   e) In-memory join = map-side join < reduce-side join

6. If a TT (Task Tracker) fails to communicate with JT (Job Tracker) for a period of time (by default, 1 minute in Hadoop), JT will assume that TT in question has crashed. Then If the job is in the reduce phase, JT asks another TT to re-execute _____.
   a) all Reducers that already finished on the failed TT
   b) all Reducers that were in progress on the failed TT
   c) all Reducers that previously on the failed TT
   d) all Reducers that were in progress and previously ran on the failed TT
   e) all Reducers that are scheduled on the failed TT

7. Who maps a block-id to a physical location on disk?
   a) Mapper
   b) Reducer
   c) Combiner
   d) Datanode
   e) Namenode

8. Which one is the correct order of execution in a map task ?
   a) setup – map – cleanup
   b) setup – map – cleanup - reduce
   c) setup – map – combine - cleanup
   d) map - map - cleanup
   e) setup – cleanup - map

9. Which kind of join(s) is/are necessary and sufficient for HW-2, Q-2 ?
   a) One to one
   b) One to many
   c) Many to many
   d) Both (a) & (b)
   e) Both (a) & (c)

1 – M

10. For the following Hadoop job :

```
public class AvgNumAll {

  public static class Map extends Mapper<LongWritable, Text, NullWritable, IntWritable> {
        int num;
        public void map(LongWritable key, Text value, Context context)
        throws IOException, InterruptedException {
                String line = value.toString();
                StringTokenizer tokenizer = new StringTokenizer(line);
                while (tokenizer.hasMoreTokens()){
                     num = Integer.parseInt(tokenizer.nextToken());
                     context.write(NullWritable.get(), new IntWritable(num));
                }
        }
  }

  public static class Reduce extends Reducer<NullWritable, IntWritable, NullWritable,
  IntWritable> {
        private int count,sum=0;
        private int avg;

        public void reduce(NullWritable key, Iterable<IntWritable> values, Context context)
        throws IOException, InterruptedException {
              count=0;
              sum=0;
              for (IntWritable val : values) {
                 count++;
                 sum+=val.get();

              }
              avg = sum/count;
              context.write(NullWritable.get(), new IntWritable(avg));
        }
}

public static void main(String[] args) throws Exception {
          Configuration conf = new Configuration();
          Job job = new Job(conf, "AvgNumAll");
          job.setJarByClass(AvgNumAll.class);

          job.setOutputKeyClass(NullWritable.class);
          job.setOutputValueClass(IntWritable.class);

          job.setMapperClass(Map.class);
          job.setCombinerClass(Reduce.class);
          job.setReducerClass(Reduce.class);

          job.setInputFormatClass(TextInputFormat.class);
          job.setOutputFormatClass(TextOutputFormat.class);
          FileInputFormat.addInputPath(job, new Path(args[0]));
          FileOutputFormat.setOutputPath(job, new Path(args[1]));
          job.waitForCompletion(true);
```

10+4+6+7
4

```
}

}
```

The above code will:

a) Calculate the average of some integers correctly

b) Calculate the average of some integers incorrectly

c) Need cleanup( ) method to make it correct

d) Calculate the average of some integers correctly if number of map task = 1

e) Calculate the average of some integers correctly if number of reduce task = 1

PART – C (15 x 2 = 30)

For each of the question in this part, you have to write 'True' or 'False' on the left side.

False 1. Map task never writes output to local disk.

False 2. Number of mappers depend on the size of the input data.

True 3. Number of reducers can be set by users.

False 4. Map and Reduce phases use HTTP protocol to share the data.

False 5. Map reduce framework exploits push schedule rather than pull schedule.

True 6. The RecordReader class actually loads data from its source and converts it into (K, V) pairs suitable for reading by Mappers.

True 7. As HW-1, question-3: To emit the top 10 entry from mapper based on some criteria, we **must** use the cleanup( ) method.

True 8. MapReduce attempts to locate slow tasks (*stragglers*) and run redundant (*speculative*) tasks.

False 9. In Pig logical operator for co-group/group is converted to 4 Physical operators

True 10. In Pig GLOBAL REARRANGE will determine MapReduce boundaries

True 11. Pig does Lazy evaluation (data not processed prior to STORE command)

True 12. Pig Latin is a procedural language.

False 13. In Hive, multi-table insert is possible not dynamic partition.

True 14. In Hive, data in each partition are divided into buckets based on hash of a column in the table. Each bucket is stored as a file in the partition directory.

False 15. Hive is used for data flow language and Pig is for data warehousing.