[Question 1 - 4] Suppose that you should build a linear regression model to predict child’s weight when his/her height is given. You are using a dataset collected from 10,000 children.

\[ h(x) = \theta_0 + \theta_1 x \]

<table>
<thead>
<tr>
<th>ID</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>...</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height (Inches)</td>
<td>40</td>
<td>41</td>
<td>38</td>
<td>51</td>
<td>48</td>
<td>40</td>
<td>...</td>
<td>42</td>
</tr>
<tr>
<td>Weight (lbs)</td>
<td>38</td>
<td>39</td>
<td>36</td>
<td>42</td>
<td>41</td>
<td>40</td>
<td>...</td>
<td>45</td>
</tr>
</tbody>
</table>

**Question 1.** In above regression model, \( \theta_0 \) and \( \theta_1 \) are the parameter vectors.  
(True/False)

**Question 2.** With above regression model, the weight of child can be predicted as the value of \( h(x) \).  
(True/False)

[Question 3 and 4] To fit the linear regression model, assume that you perform the Batch Gradient Descent with MapReduce using 10 mappers.

For the given sample size 10,000, you will need to use following formula:

\[ \theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \]

\[ \theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})x^{(i)} \]

**Question 3.** To calculate \( \theta_0 \), each mapper can take 1,000 data points, if they fit in a split with the default size.  
(True/False)

**Question 4.** To complete calculating \( \theta_0 \), this calculation will always require total 10 MapReduce jobs.  
(True/False)
(1) In Spark, RDDs are always stored in memory.  
(   True   /   False   )

(2) `collect` is an action in Spark.  
(   True   /   False   )

(3) Spark will not begin executing any operation until it sees an action.  
(   True   /   False   )

(4) Spark allows users to modify values stored in RDDs.  
(   True   /   False   )

(5) A Spark operation `groupByKey()` requires a narrow dependency.  
(   True   /   False   )
Suppose that there are two pair RDDs, RDD1 and RDD2.

\[ \text{RDD1} = \{(1,2), (1,3), (1,4), (2,5), (2,6), (3,7)\} \]
\[ \text{RDD2} = \{(2,2)\} \]

(6) If following function is applied to RDD1, what will be the result?
Function: RDD1.mapValues(x -> x*x)

a. \[ \{(1,4), (1,9), (1,16), (2,25), (2,36), (3,49)\} \]

b. \[ \{(1,2), (1,3), (1,4), (4,5), (4,6), (9,7)\} \]

c. \[ \{(1,4), (1,9), (1,16), (4,25), (4,36), (9,49)\} \]

d. \[ \{(1,2), (1,3), (1,4), (2,5), (2,6), (3,7)\} \]

(7) If following function is applied to RDD1, what will be the result?
Function: RDD1.reduceByKey((x,y) -> x+y)
Result:
\[ \{(1,9), (2,11), (3,7)\} \]

(8) If following function is applied to RDD1 and RDD2 what will be the result?
Function: RDD1.join(RDD2)

a. \[ \{(2,5,2), (2,6,2)\} \]

b. \[ \{(2,(Some(5),2)), (2,(Some(6),2))\} \]

c. \[ \{(1, (2, None)), (1, (3, None)), (1, (4, None)), (2, (Some(5), 2)), (2, (Some(6), 2)), (3, (7, None))\} \]

d. \[ \{(1, (None, 2)), (1, (None, 3)), (1, (None, 4)), (2, (2, Some(5))), (2, (2, Some(6))), (3, (None, 7))\} \]

(9) If following function is applied to RDD1 and RDD2 what will be the result?
Function: RDD1.rightOuterJoin(RDD2)

a. \[ \{(2,5,2), (2,6,2)\} \]

b. \[ \{(2,(Some(5),2)), (2,(Some(6),2))\} \]

c. \[ \{(1, (2, None)), (1, (3, None)), (1, (4, None)), (2, (Some(5), 2)), (2, (Some(6), 2)), (3, (7, None))\} \]

d. \[ \{(1, (None, 2)), (1, (None, 3)), (1, (None, 4)), (2, (2, Some(5))), (2, (2, Some(6))), (3, (None, 7))\} \]

(10) If following function is applied to RDD1, and RDD2 what will be the result?
Function: RDD1.leftOuterJoin(RDD2)

a. \[ \{(2,5,2), (2,6,2)\} \]

b. \[ \{(2,(Some(5),2)), (2,(Some(6),2))\} \]

c. \[ \{(1, (2, None)), (1, (3, None)), (1, (4, None)), (2, (5, Some(2))), (2, (6, Some(2))), (3, (7, None))\} \]

d. \[ \{(1, (None, 2)), (1, (None, 3)), (1, (None, 4)), (2, (2, Some(5))), (2, (2, Some(6))), (3, (None, 7))\} \]
Question 1.
Google File System (GFS) does support the complete standard APIs such as POSIX.  
(True/False)

Question 2.
The master in GFS maintains the locations of chunks only in the memory.  
(True/False)

Question 3.
A GFS (GFS 1) cluster consists of a single master and multiple chunk servers.  
(True/False)

Question 4.
GFS guarantees that all of the replicas of a chunk are stored in a single chunk server.  
(True/False)

Question 5.
The CAP theorem states that it is impossible for a distributed computer system to simultaneously provide all three of the following guarantees: Consistency, Atomicity, and Partition tolerance.  
(True/False)
1) In a key-value storage, users can query the contents using the stored values.

(True / False)

2) In Dynamo, if a node is temporarily unavailable, the metadata is propagated to the next node in the ring and provides a hint for subsequent read operations.

(True / False)

(Problem 3 and 4)
Suppose that there is Dynamo key-value storage with 3 storage nodes with the ids (A =0, B =8, C =11, D=14). The ID space spans: 0 ... (2^4-1). The replication factor is 3.

3) Consider the case where node B has received a query to retrieve a <key, value> pair. The hash value of the key was 0. To locate the storage node that stores this <key, value> pair, how many storage nodes should be visited including the first visited node (here, node B)?
   a. log16  b. 0  c. 2  d. 1
   Note: please make sure that this question does not asking the number of nodes to visit to “retrieve” values.

4) Consider that node E has been added to this storage cluster with the id of 3. What will be the hashed keys that will be stored in E? Consider all of the ranges for replication data.
   a. 1,2,3  b. 9,10,11,12,13,14,15,0,1,2,3  c. 15,0,1,2,3  d. 3,4,5,6,7
   No answer. Correct answer is the set of hash keys stored in the node D, A, and E, therefore, 12,13,14,15,0,1,2,3

5) Suppose that a distributed key-value storage cluster maintains Merkle trees to synchronize replications. Storage nodes, r_a, r_b, and r_c store replications of the same data blocks. Assume that the roots of the Merkle trees of r_a and r_b were different and the roots of the Merkle trees of r_b and r_c were the same. If the hash is calculated only for the write operation (write quorum is 2), which replication server might have one or more corrupted data blocks?
   a. r_a (answer)  b. r_b  c. r_c  d. none