Question 1 (marks = 50)

Premise:

If A is an n x m matrix and B is an m x p matrix,

$$\mathbf{A} = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ A_{21} & A_{22} & \cdots & A_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nm} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1p} \\ B_{21} & B_{22} & \cdots & B_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ B_{m1} & B_{m2} & \cdots & B_{mp} \end{pmatrix}$$

the matrix product AB (denoted without multiplication signs or dots) is defined to be the n x p matrix

$$\mathbf{AB} = \begin{pmatrix} (\mathbf{AB})_{11} & (\mathbf{AB})_{12} & \cdots & (\mathbf{AB})_{1p} \\ (\mathbf{AB})_{21} & (\mathbf{AB})_{22} & \cdots & (\mathbf{AB})_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ (\mathbf{AB})_{n1} & (\mathbf{AB})_{n2} & \cdots & (\mathbf{AB})_{np} \end{pmatrix}$$

where each i, j entry is given by multiplying the entries A_{ik} (across row i of A) by the entries B_{kj} (down column j of B), for k = 1, 2, ..., m, and summing the results over k:

$$(\mathbf{AB})_{ij} = \sum_{k=1}^{m} A_{ik} B_{kj} \,.$$

Thus the product AB is defined only if the number of columns in A is equal to the number of rows in B, in this case m.

Problem Statement:

a. Perform the matrix multiplication of two input matrices, containing integer values only. The matrices will be multiplication compatible (i.e., given two matrices $A_{m \times n}$ and $B_{p \times q}$, the equality n=p will hold). **(marks = 30)**
b. Consider multiplying two matrices of size n x m and m x p, and there are k number of nodes/servers in the cluster. Calculate the total computation and communication cost for the multiplication process. The total computation and communication cost can be computed by adding the computation and communication cost separately for each map and reduce step. You have to provide two equations for two different costs, in terms of m, n, p and k. **(marks = 20)**

Input:

Input will be a text file ***input_matrices.txt*** containing two integer matrices as follows, with each row containing 1 integer:

A, 0, 0, 7
A, 0, 1, 4
...
B, 0, 0, 3
B, 0, 1, 6
...

The left hand side matrix is denoted as A and right hand side matrix as B. Next two fields are matrix indices (row and column respectively) and the last field is the value at that index.

Output:

The output file produced by your code should be a text file containing only the indices and the values of the product matrix, as follows:
0, 0, 22
0, 1, 18
…

Question 2 (marks = 50)

Premise:

To calculate the k^{th} percentile of a list of integers (where k is any number between zero and one hundred), we need to do the following steps:

1. Sort the values in ascending order.
2. Multiply k percent by the total number of values, n. This number is called the index.
3. If the index obtained in step 2 is not a whole number, round it up to the nearest whole number and go to step 4. If the index obtained in step 2 is a whole number, go to step 5.
4. Count the values in your data set from left to right (from the smallest to the largest value) until you reach the number indicated by step 3. The corresponding value in your data set is the k^{th} percentile.
5. Count the values in your data set from left to right until you reach the number indicated by step 2. The k^{th} percentile is the average of that corresponding value in your data set and the value that directly follows it.

Problem Statement:

Given a list of integers, your task is to find the 25th, 50th and 75th percentiles of the list. For example, given a list of 10 integers [2, 8, 7, 4, 1, 2, 2, 4, 2, 8], their 25th percentile is 2 (at position 3), 50th percentile is (2+4)/2 = 3 (2 is at 5th position and 4 is at 6th) and 75th percentile is 7(at position 8).
Also write the algorithm/approach you will use for solving the above problem using RDDs. Extra credit will be provided for not sorting the entire list in any of your steps.
(Marks: Code + Write-up = 30
Extra credit = 20)

Input:

Input will be a text file ***input_list.txt*** containing a list of integers, with each row containing 1 integer.

Output:

Your code should output the result as 3 tab-separated values as the following:
<25th percentile><\tab><50th percentile><\tab><75th percentile>

Deliverables:

Submit a zipped file ***<Roll_No>_a2.tar.gz*** (e.g., ***16CS60R75_a2.tar.gz*** if your roll no. is 16CS60R75). The zipped file should contain two code files, ***a2q1a.scala*** and ***a2q2.scala*** (the extensions will change accordingly if you write your codes in Python or Java), each containing the solution code for the corresponding question. It will also contain the write-ups, ***a2q1b.pdf*** and ***a2q2_algo.pdf***.