Q1. HDFS (15 Marks)

Consider an HDFS using Erasure Coding.

1. Assume that (6,3)-Reed-Solomon coding is used, and each cell or parity is represented by an unsigned integer. Would the following 9 internal blocks (e.g., the first 6 internal blocks are raw data blocks and the last 3 internal blocks are parity blocks) come from the same striped block group? You need to justify your answer.

```
internal block1: [175, 234, 117, 250, 487, 185],
internal block2: [434,  78, 479, 407,  98, 409],
internal block3: [339, 261, 368, 148, 414, 105],
internal block4: [433, 344, 100, 155, 386, 222],
internal block5: [434, 180, 401, 327, 491, 421],
internal block6: [327,  47, 386,  79,  92, 453],
internal block7: [ 4650, 2265, 4079, 3137, 3946, 3782],
internal block8: [ 6614, 3496, 5095, 4192, 5455, 5607],
internal block9: [11264, 5761, 9174, 7329, 9401, 9389].
```

2. Assume that (x,y)-Reed-Solomon coding is used (i.e., each stripe contains x cells and y parities). Under what condition (x,y)-Reed-Solomon coding has the same maximum toleration as (6,3)-Reed-Solomon coding?

Q2. Spark and MapReduce (15 Marks)

1. Given an RDD `student_record` containing tuples that are formed as `(s_id, score)` (e.g., `("z3212321", 66)`). We want to get the top-2 highest scores (higher score comes first) for each student. If a student has only information for one course in `student_record`, then only one score should be output.

   Complete three functions `createCombiner`, `mergeValue`, and `mergeCombiners`, such that the output RDD (i.e., `score.combineByKey(createCombiner, mergeValue, mergeCombiners)`) serves our purpose. The key value pairs of the output RDD should be `(s_id, (score_1, score_2))` (e.g., `("z3212321", (93,78))`) or `(s_id, (score_1))` (e.g., `("z3212321", (93))`) if a student has only information for one course.

2. Consider the following code which is written by a (virtual) student for Project 1:

```
def collision_count(a, b, offset):
    counter = 0
    for i in range(len(a)):
        if abs(a[i]-b[i]) <= offset:
            counter += 1
    return counter

def c2lsh(data_hashes, query_hashes, alpha_m, beta_n):
    offset = 0
    cand_num = 0
    while cand_num < beta_n :
        candidates = data_hashes.flatMap(lambda x :
         [x[0]] if collision_count(x[1], query_hashes, offset)>=alpha_m else [])
        cand_num = candidates.count()
        offset += 1
    return candidates
```

   Is this implementation correct? You need to justify your answer.

Q3: LSH (15 marks)

LSH uses AND-OR composition to combine hash values and generate nearest neighbor candidates as discussed in this course. Now consider the following **OR-AND** composition scheme (namely *OALSH*) with min-hash functions $h_{i,j}(\cdot)$:

- o and q are considered as matching on a super hash function $H_i(\cdot)$ if $\exists j \in \{1, \ldots, R\}, h_{i,j}(o) = h_{i,j}(q)$.
- o is q's candidate if $H_i(o)$ matches $H_i(q)$ for all $i \in \{1, \ldots, S\}$.

1. Assume the Jaccard similarity between o and q is 0.8, compute the probability that o is a candidate of q if we use OALSH with $R = 4, S = 5$.
2. Given a threshold $t = 0.5$, we want to find the sets that have Jaccard similarity to q no less than t (considered as positives). Consider the OALSH scheme with $R = 2, S = 5$ and an LSH scheme with $k = 5$, what is the maximum l such that the expected recall ($recall = \frac{\text{number of returned positives}}{\text{total number of positives}}$) for LSH scheme is guaranteed to be higher than OALSH scheme?
3. Assume that we want to find the sets that have Jaccard similarity to q no less than a threshold t. Is it possible to have a parameter setting of k, l, R, S for LSH and OALSH scheme, such that OALSH would have a higher expected recall than LSH for any treshold t? You need to justify your answer.

Q4: Spark SQL(12 marks)

Write a PySpark SQL code to output the maximum and minimum scores (i.e., `max` and `min`) for each `Id` in a dataFrame named `record`, the output should be sorted by `Id`. Note that:

- You cannot inject SQL statements in your code.
- You cannot use RDD operations in your code.
- The output dataframe (i.e., `maxmin`) contains only three columns: `Id`, `max` and `min`.

For example,

```
record.show(5)
+--+------+-----+
|Id|Course|Score|
+--+------+-----+
| 1|  9313|   80|
| 1|  9318|   75|
| 1|  6714|   70|
| 2|  9021|   70|
| 3|  9313|   90|
+--+------+-----+

###########
## your code
###########

maxmin.show(3)
+--+---+---+
|Id|max|min|
+--+---+---+
| 1| 80| 70|
| 2| 70| 70|
| 3| 90| 50|
+--+---+---+
```

Q5: Stacking (15 marks)

Assume that we are applying stacking training mechanism to train a series of models. Suppose we have 3 base classifiers, and 1 meta classifier. We decide to use 5-fold separation (i.e., split the training data into 5 groups as in project 2) on the training set for base classifiers training.

1. Calculate the total number of classifiers trained in this stacking setting. You need to show your steps.
2. Calculate the number of times that an instance in the training set is processed as a training instance by all the classifiers. If an instance is processed by one classifier multiple times (e.g., multiple epochs while training the classifier), count it as one time. You need to show your steps.
3. Calculate the number of times an instance in the training set is processed to generate predictions by all the classifiers. You need to show your steps.

Q6: Mining Data Streams (13 marks)

Consider a Bloom filter of size m = 8 (i.e., 8 bits) and 2 hash functions that both take a string (lowercase) as input:

- $h1(str) = \sum_{c \in str}(c - \text{'a'}) \mod 8$
- $h2(str) = str.length \mod 8$

Here, $c - $ 'a' is used to compute the position of the letter c in the 26 alphabetical letters, e.g., $h1(\text{"bd"}) = (1 + 3) \mod 8 = 4$.

1. Given a set of string S = "hello", "map", "reduce", show the update of the Bloom filter.
2. Given a string "spark", use the Bloom filter to check whether it is contained in S.
3. Given S in the first subquestion and the Bloom filter with 8 bits, what is the percentage of the false positive probability? You need justify your answer.

Q7: Recommender System (15 marks)

Given the following ratings (? represents the unknown rating).

	users				
movies	3	5	?		2
		4		1	?
	4	?	5	2	

1. Predict the unknown ratings using baseline estimator.
2. Predict the unknown ratings using matrix factorization, where

$$Q = \begin{pmatrix} 2.3 & 1.2 & 1.5 & 0.4 \\ 1.5 & 3.2 & 0.6 & 1.7 \\ 2.1 & 1.3 & 2.8 & 0.4 \end{pmatrix} \text{ and } P = \begin{pmatrix} 0.7 & 0.7 & 0.7 & 0.5 \\ 0.7 & 0.9 & 0.8 & 0.3 \\ 0.8 & 0.6 & 0.7 & 0.8 \\ 0.1 & 0.1 & 0.6 & 0.4 \\ 0.4 & 0.6 & 0.5 & 0.7 \end{pmatrix}$$

3. Assume we have the true ratings as below:

	users				
movies	3	5	3		2
		4		1	4
	4	4	5	2	

Which of the above two estimations is better based on RMSE? You need to show your steps.