Q1

Write a MapReduce program in Spark that implements a simple "Mutual/Common friend list of two friends". The key idea is that if two people are friend then they have a lot of mutual/common friends. This question will give any two Users as input, output the list of the user id of their mutual friends.

For example,

Alice's friends are Bob, Sam, Sara, Nancy

Bob's friends are Alice, Sam, Clara, Nancy

Sara's friends are Alice, Sam, Clara, Nancy

As Alice and Bob are friend and so, their mutual friend list is [Sam, Nancy]

As Sara and Bob are not friend and so, their mutual friend list is empty. (In this

case you may exclude them from your output).

Input:

Input files

soc-LiveJournal1Adj.txt

The input contains the adjacency list and has multiple lines in the following format:

<User><TAB><Friends>

Here, <User> is a unique integer ID corresponding to a unique user and <Friends> is a comma-separated list of unique IDs (<User> ID) corresponding to the friends of the user. Note that the friendships are mutual (i.e., edges are undirected): if A is friend with B then B is also friend with A. The data provided is consistent with that rule as there is an explicit entry for each side of each edge. So when you make the pair, always consider (A, B) or (B, A) for user A and B but not both.

Output:

The output should contain one line per user in the following format:

<User_A><TAB><User_B><TAB><Mutual/Common Friend List>

where <User_A> & <User_B> are unique IDs corresponding to a user A and B (A and B are friend). < Mutual/Common Friend List > is a comma-separated list of unique IDs corresponding to mutual friend list of User A and B. Please find the above output for the following pairs.

(0,4), (20, 22939), (1, 29826), (6222, 19272), (28041, 28056)

Q2.

Please answer this question by using dataset from Q1 and 'userdata' dataset as below.

Userdata.txt:

column1: userid

column2: firstname

column3: lastname

column4: address

column5: city

column6: state

column7: zipcode

column8: country

column9: username

column10: dateofbirth

Now Let's consider all the users (i.e. every line) in **soc-LiveJournal1Adj.txt.**

Find top-10 friend pairs by their total number of common friends. For each top-10 friend pair, print detailed information in decreasing order of total number of common friends. More specifically the output format can be:

<total number of common Friends><TAB><First Name of User A><TAB><Last Name of User A><TAB><Address of User A><First Name of User B><TAB><Last Name of User B><TAB><Address of User B>

Q3.

For this question, you will work with three datasets:

1. movies.csv

Contains movieID, title and genre

2. ratings.csv

Contains userID, movieID, ratings and timestamp

3. tags.csv

Contains userID, movieID, tags and timestamp

You need to compute the following results

1a. Calculate the average ratings of each movie.

1b. Give the names of bottom 10 movies with lowest average ratings.

2. Find average rating of each movie where the movie's tag is 'action'.

3. Find average rating of each movie where the movie's tag is 'action' and genre contains 'thrill'.

Hint: 'join by key' technique is needed.