**BIG DATA PROGRAMMING**

**PROJECT-1**

**Case1-FacebookMutualFriends**

**Team Members and collaboration:**

Roshini varada --Facebook Mutual Friends

Zakari, Abdulmuhaymin --Youtube analysis

Sarika Reddy Kota -- Hive Use case

Pallavi Arikatla -- Solr use case

**Idea: (Question4- Part-1)**

To identify the Mutual Friends of two people in a network with the help of map-reduce. The Map reduce algorithm has two functions map() and reduce() functions where the mapper takes key as the person and value as his list friends. The mapper emits outputs with sets of key value pairs where pair of users becomes the key and the combined list of sets of friends becomes the value. This will be given as an input to the reducer which will intersect the both sets to identify the common friends.

**Usage or the real time scenario: (Question4- Part-2)**

Facebook or the other social networking sites identifies the mutual friends between people. Whenever we visit somebody's profile the mutual contacts for the both profiles will be displayed.

**Example:**

If A and B has C,D as their common friends, If A visits B's profile A can identify C,D as their mutual friends and vice versa.

**Approach and solution: (Question4- Part-3)**

**Flow Chart or Pictorial Representation:**

1.The identification of mutual friends for a unit takes place in 5 phases. They are

a.Input

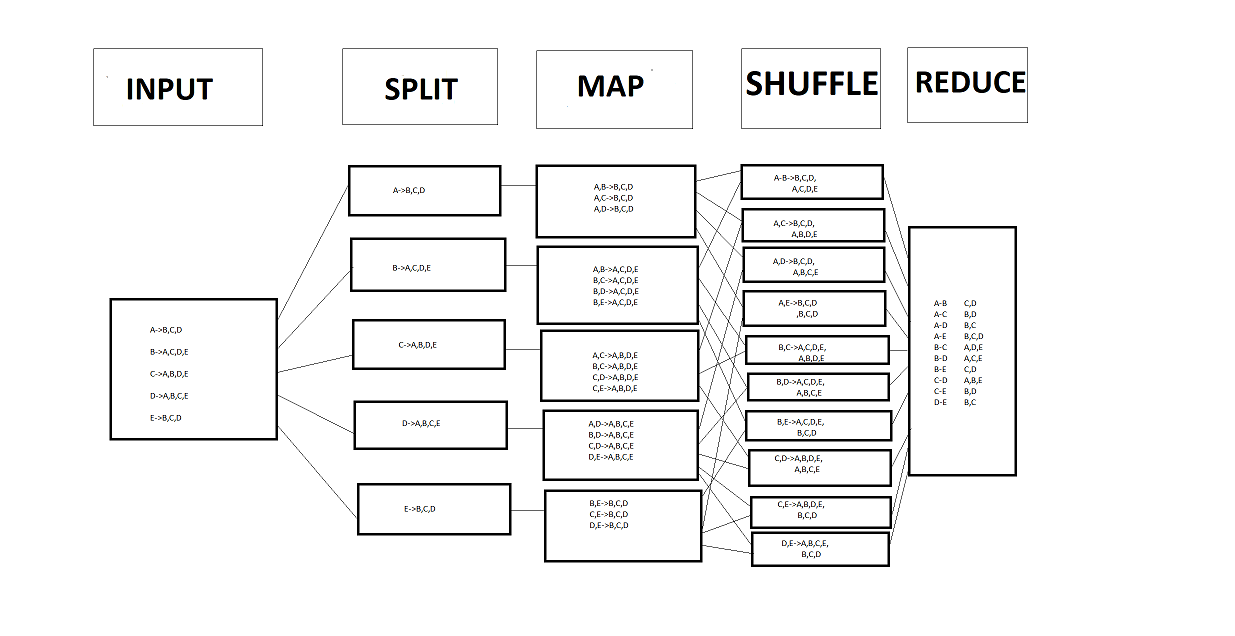
b.Split

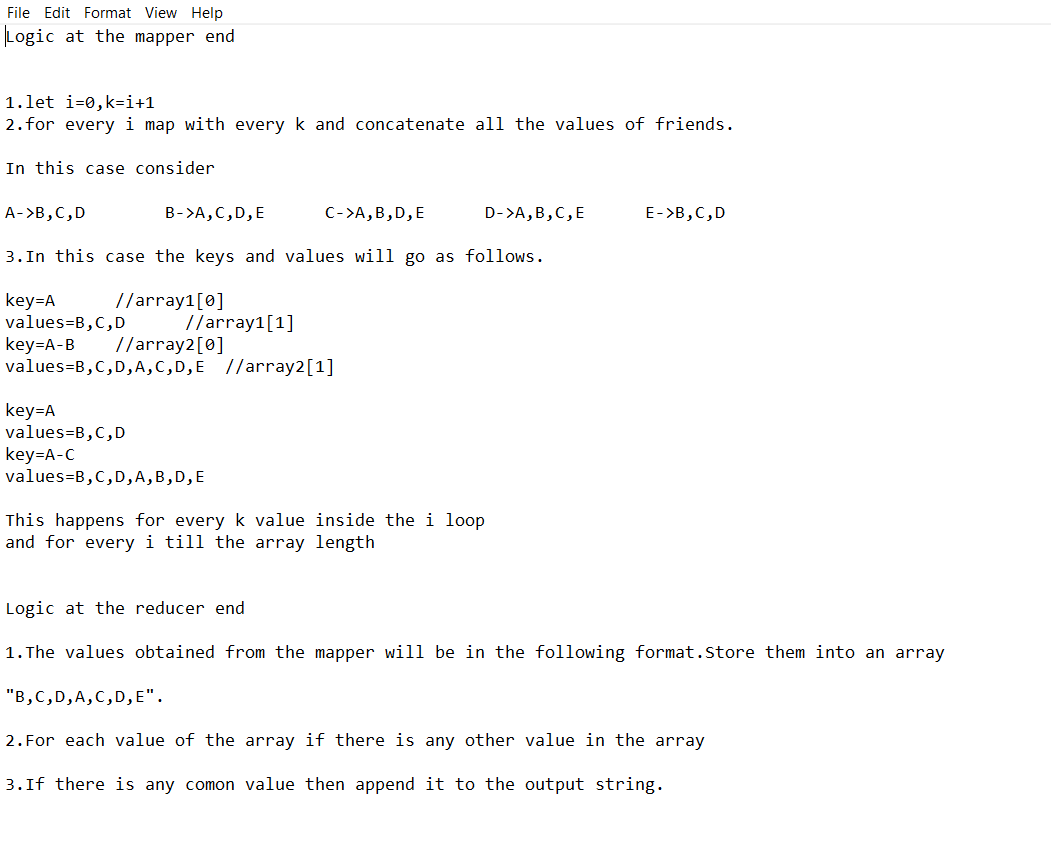
c.Map

d.Shuffle

e.Reduce

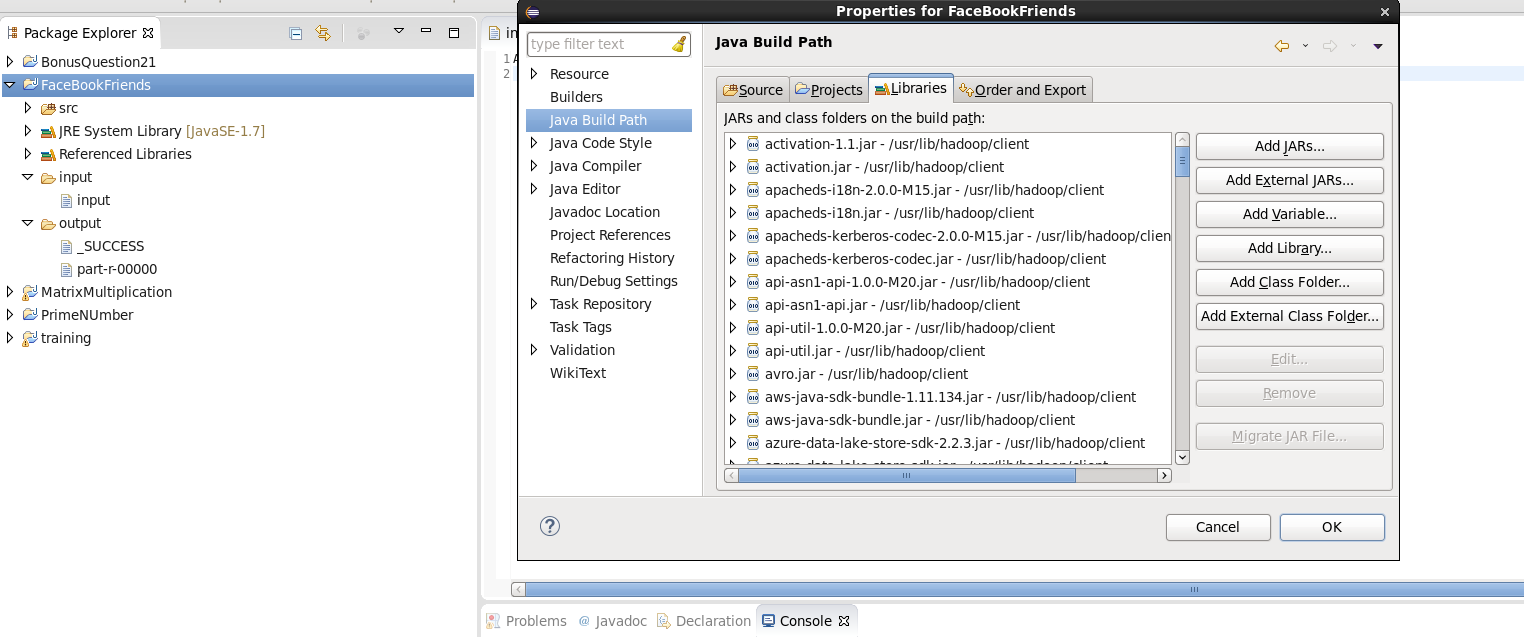
The below diagram represents how the input transforms in each phase.

**Algorithm:**

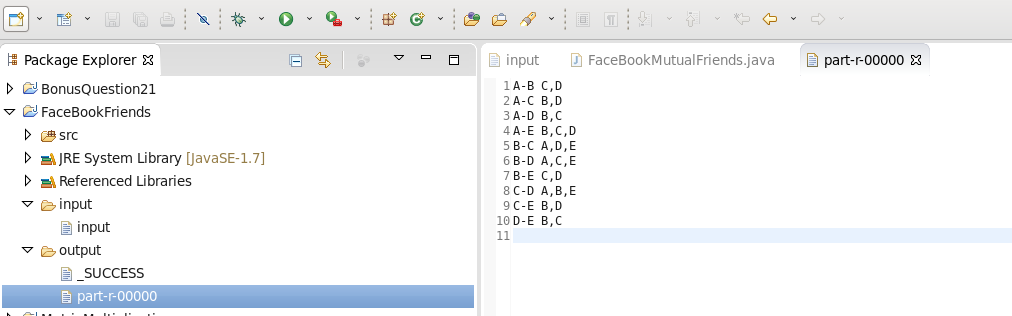


**Implementation**

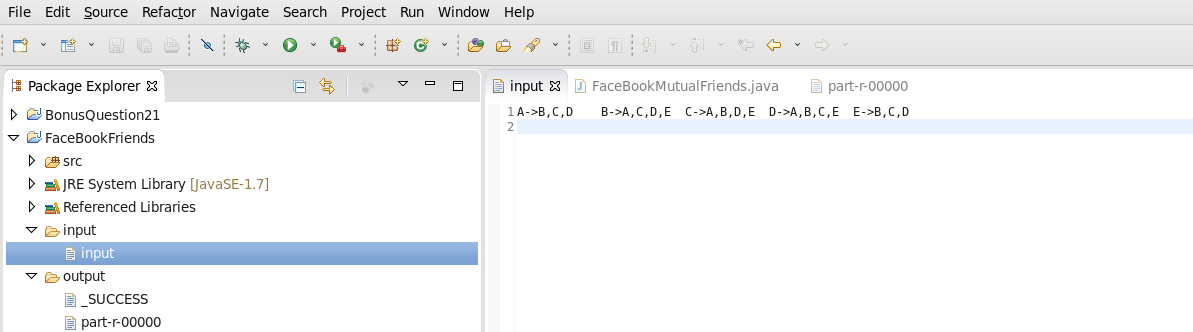
1.Create a new project and add necessary jar files.



2.Create a folder structure in the eclipse with input and output folders.



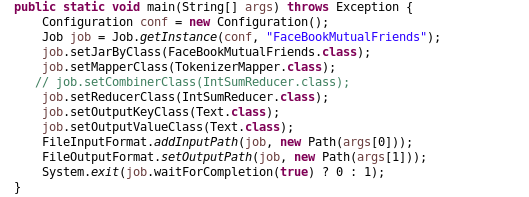
**Input**



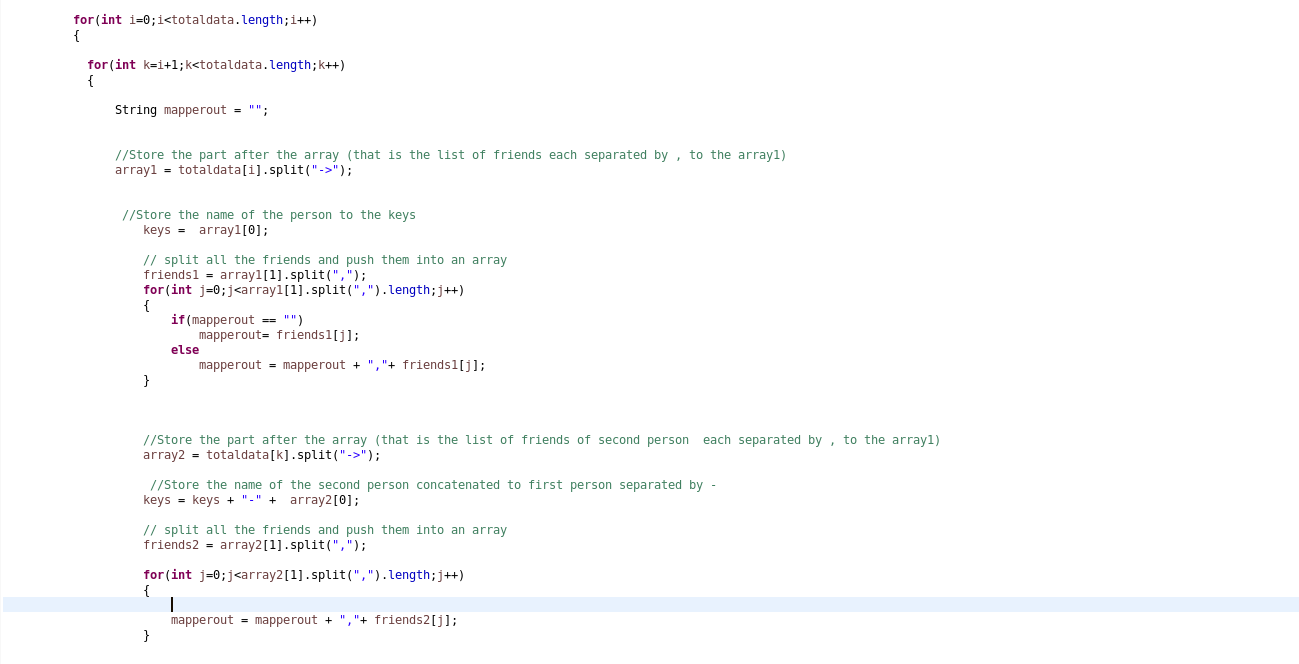
The input indicates 5 people A,B,C,D,E with their respective friends.Each are seperated by tab spaces.

**Program**

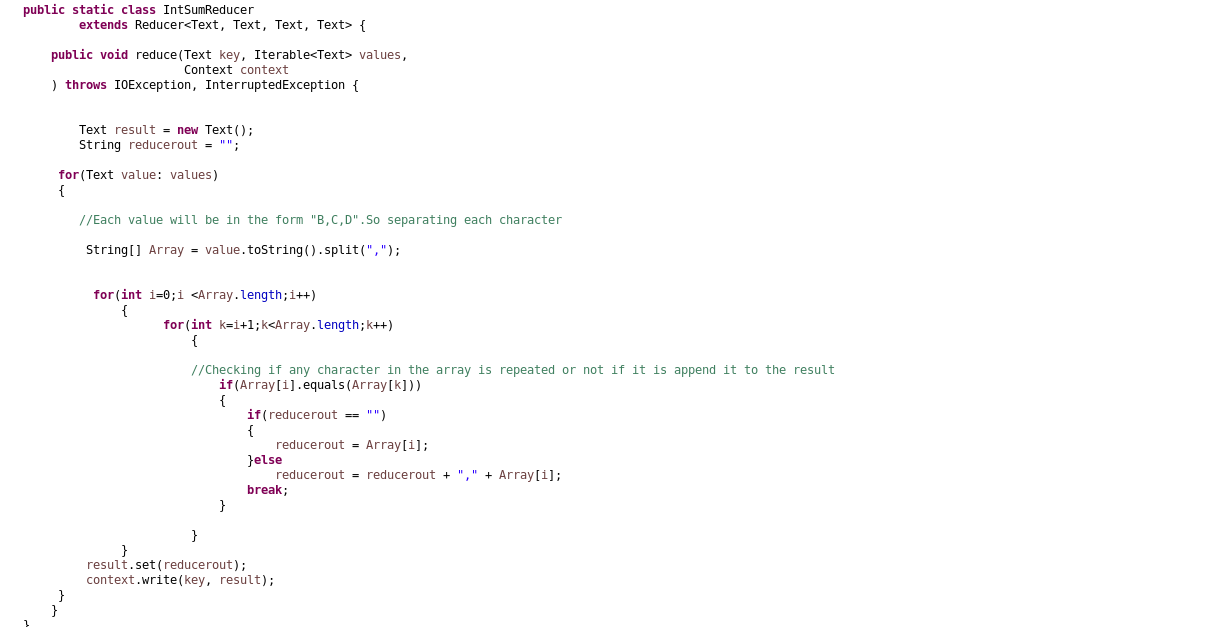
**MainMethod:**



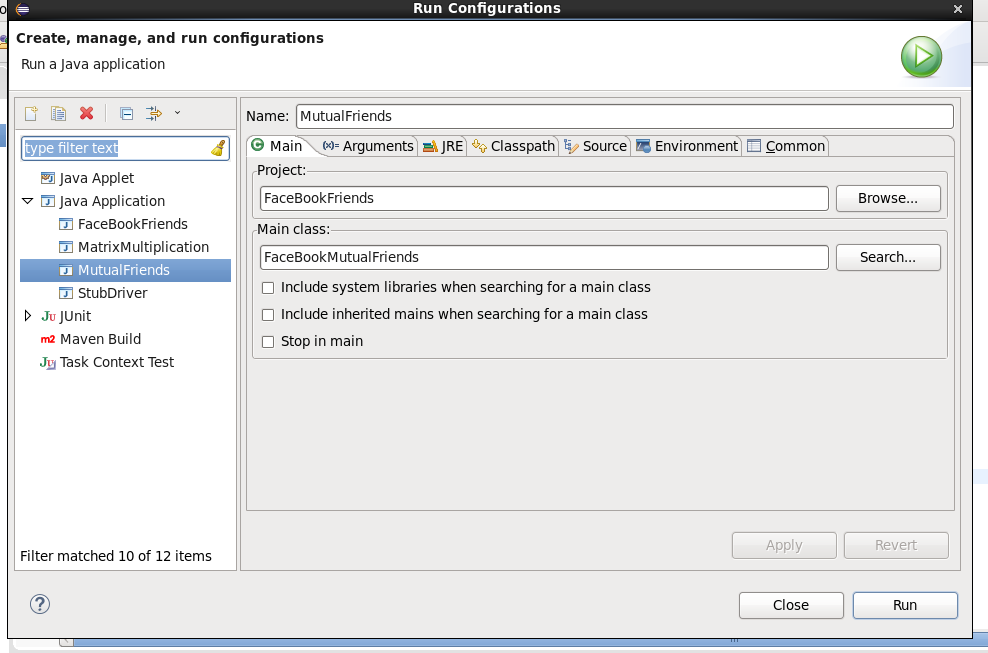
**Mappercode**



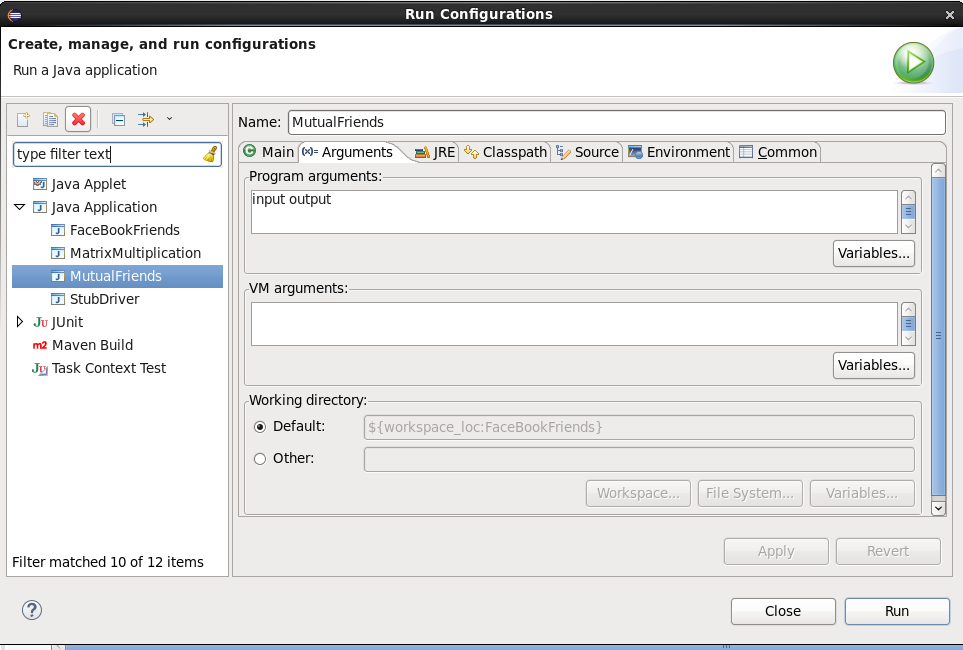
**Reducercode**



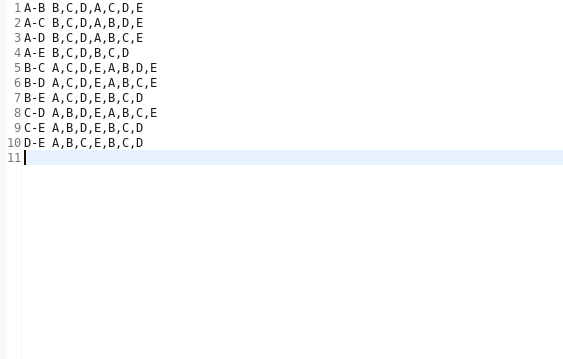
**Creating Configuration in eclipse**



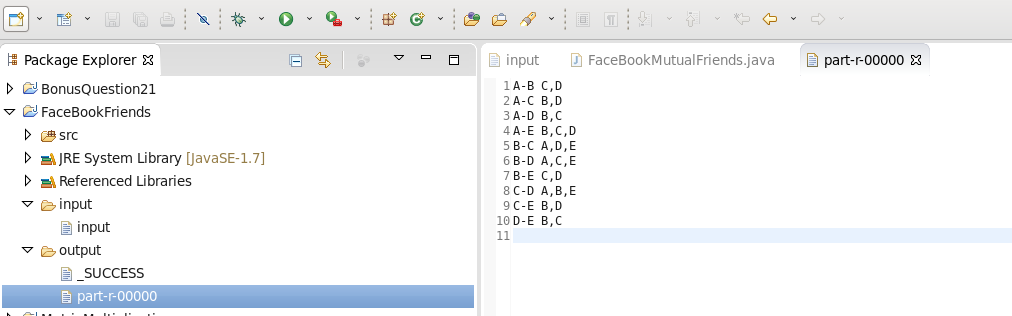
**Setting arguments input and output**



3.Run the program and the output at mapper is generated as follows.

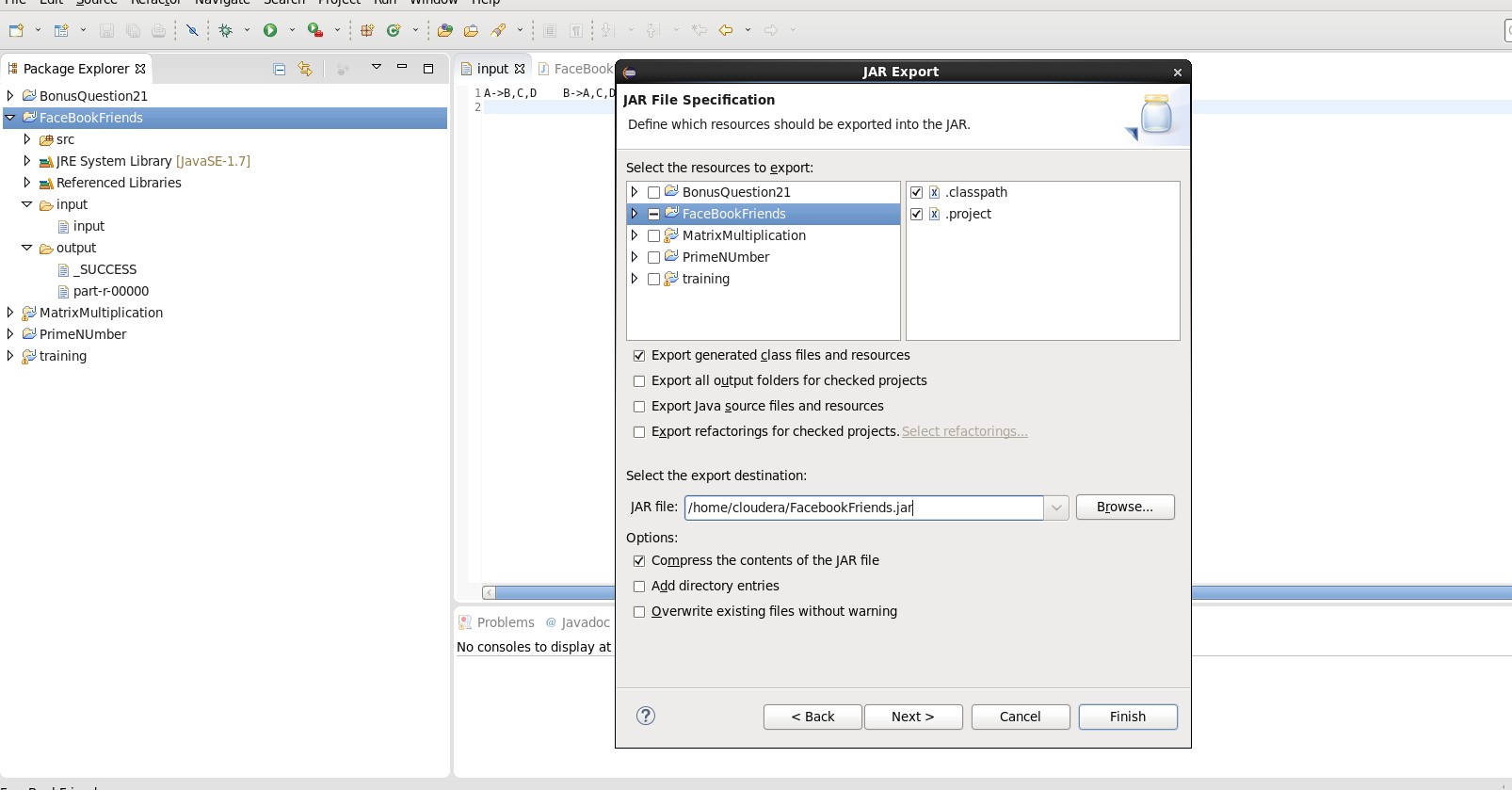


4.The final output is generated as follows.



**Steps to view the output in HUE Browser**

5.Now export the project in to jar file in the eclipse



6.Now load the input data into the hdfs using the command

"hadoop fs -put facebookinput.txt project1/input"

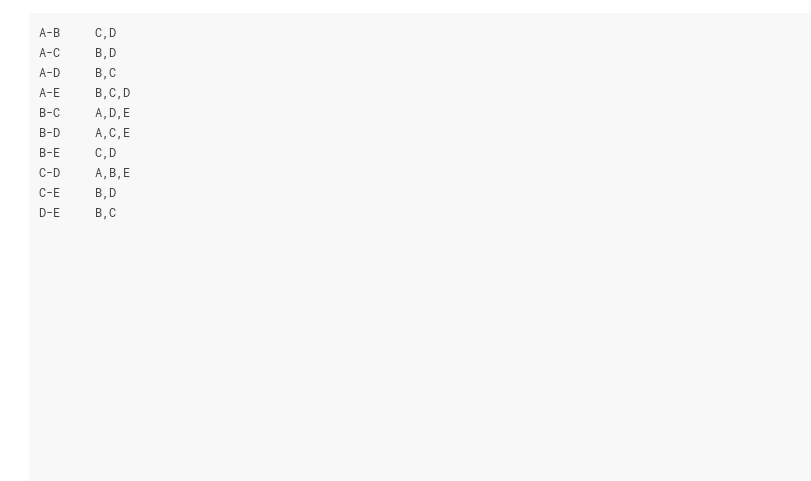
7.The input in hue broswer looks as follows.



8.Now run the jar file that is generated in the eclipse using the following command.

"hadoop jar FacebookFriends.jar FaceBookMutualFriends project1/input project1/output"

9.Then the output can be viewed in the hue browser as follows.



**Challenges faced (Question4- Part-4)**

1.The coding part was smooth

**Github Link**

<https://github.com/RoshiniVarada/BDP_Projects/tree/master/Case1-FacebookMutualFriends>

**Link for Commands:**

<https://github.com/RoshiniVarada/BDP_Projects/blob/master/Case1-FacebookMutualFriends/Commands/commands>

**Link for Sourcecode**

<https://github.com/RoshiniVarada/BDP_Projects/blob/master/Case1-FacebookMutualFriends/Sourcecode/FaceBookMutualFriends.java>

**Link for YouTube Video**

<https://youtu.be/KmpgsL1s_So>

**Team Members Links**

**Use-case1 -FacebookMutualFriends**

Project-link -https://github.com/RoshiniVarada/BDP\_Projects/tree/master/Case1-FacebookMutualFriends

Wiki-link-https://github.com/RoshiniVarada/BDP\_Projects/wiki/Case1-FacebookMutualFriends

**Use-case2-YouTubeAnalysis**

Project-link-https://github.com/RoshiniVarada/BDP\_Projects/tree/master/Case2-YouTubeAnalysis

Wiki-link-https://github.com/RoshiniVarada/BDP\_Projects/wiki/Case2-YouTubeAnalysis

**Use-case3- Hive\_UseCase**

Project-link- https://github.com/RoshiniVarada/BDP\_Projects/tree/master/Case3\_HiveUseCase

Wiki-link- https://github.com/RoshiniVarada/BDP\_Projects/wiki/Case3\_Hive\_Usecase

**Use-case4- Solr\_Usecase**

Project-link- https://github.com/RoshiniVarada/BDP\_Projects/tree/master/Case4\_Solr%20usecase

Wiki-link- https://github.com/RoshiniVarada/BDP\_Projects/wiki/Case4-Solr