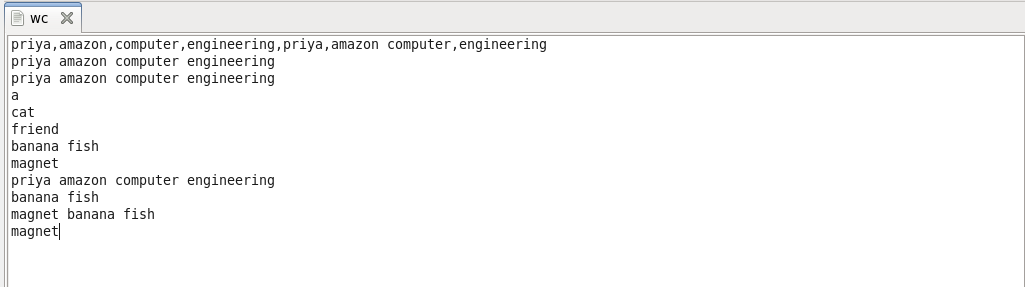
**Assignment 15.1**

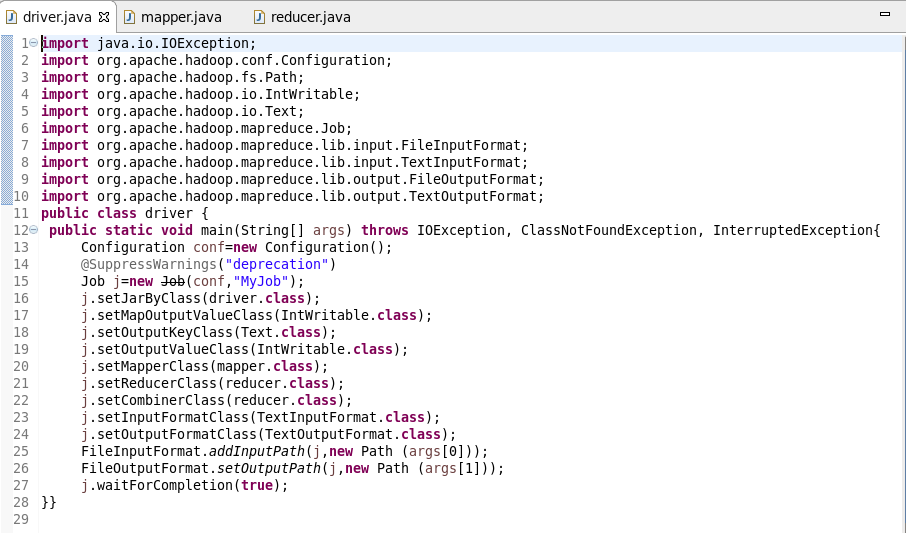
**Problem Statement:**

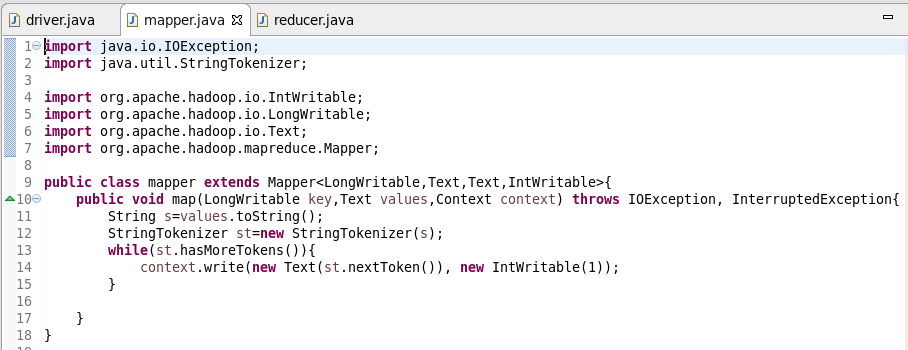
Perform wordcount operation in local mode (In windows environment). Share the code, explain the execution steps, and attach the output screenshot.

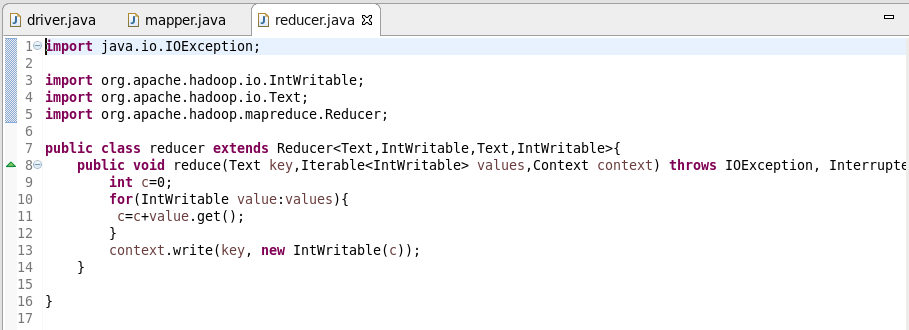
**Input:**

****

**Program:**

****

****

****

**Execution steps:**

Follow the below procedure to execute your Mapreduce programs locally in eclipse, this saves your hdfs memory and time to check your program:

1. Open eclipse

2. Create a Java Project

3. Create a new package (optional)

4. Create a new class

5. Copy your program in to that class

You need to add dependencies for running in eclipse which means few more jars need to be configured in your libraries.

All the jars present in the lib folder of the common directory of hadoop.

Hadoop common 1.2.1 jar (Need to be imported externally)

**To add the jar files**

Right click on the project–>Build Path–>Configure Build Path–>Libraries–>Add External Jars–>open your hadoop folder–>share–>hadoop–>common–>lib–>

Add all the jars in lib folder

Then you need to add another external jar for dependencies i.e., hadoop-core-1.2.1 jar

Download that jar file from the below linkhttps://drive.google.com/file/d/0ByJLBTmJojjzM2IwU1FPdmExLUE/view?usp=sharing

After downloading you need to add this jar in to your libraries.

Now you are ready to run your program in eclipse,

**To run**

Right click on the project–>Run as–>Run configurations–>main

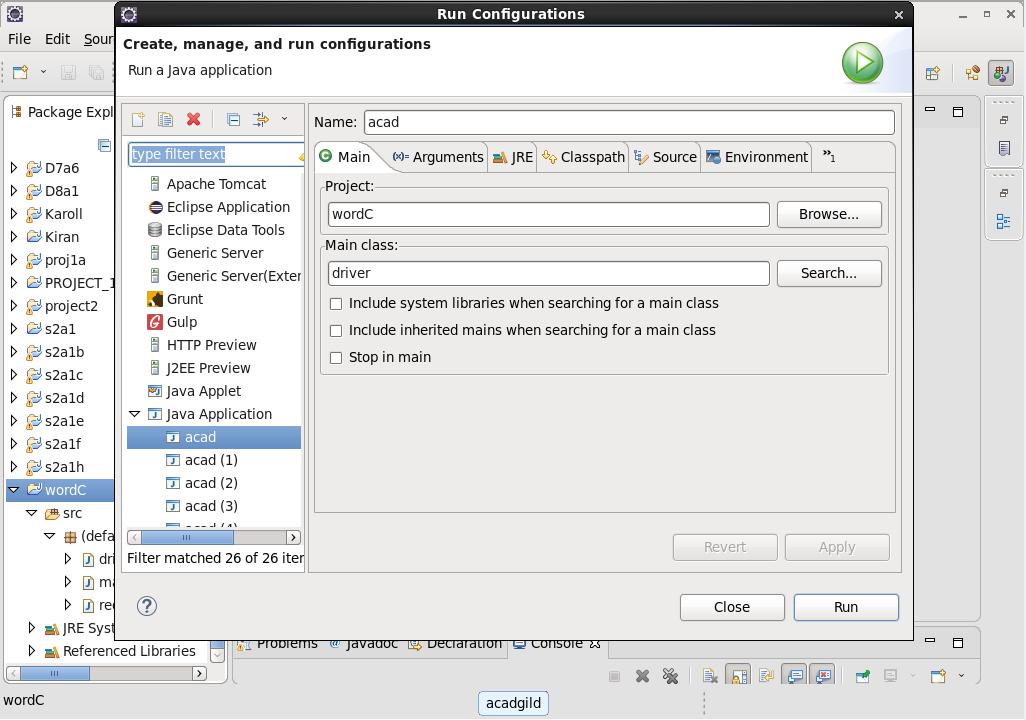
In main you need to select your project and main class correctly

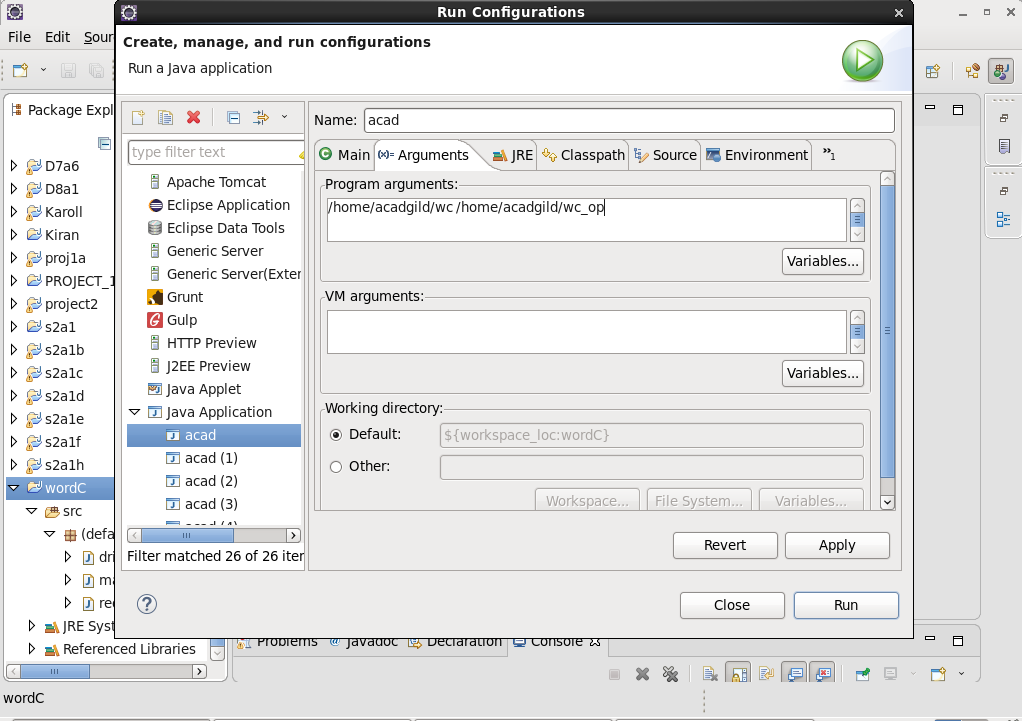
Then move into the Arguments tab

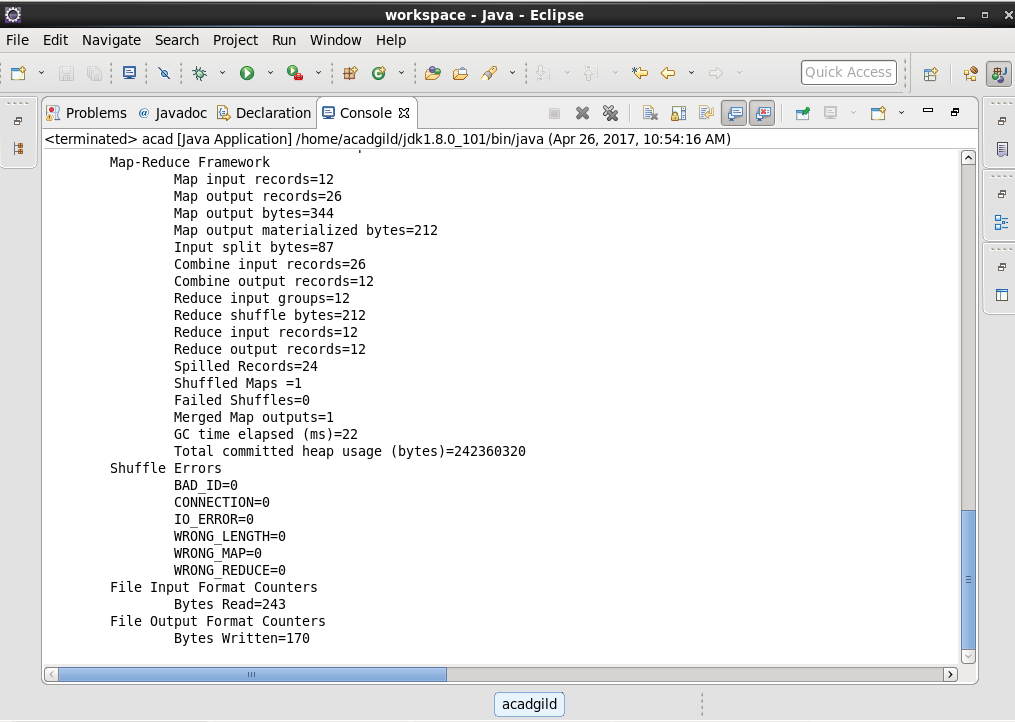
Here you need to give your input file path and output file path separated by Tab space

Now click on Run then your program will start running and you can track the status in console after the whole process you can see that an output file will be created in your specified folder.

Inside that folder you can see a part file and a success file which indicates that you have executed your program successfully in eclipse locally.

****

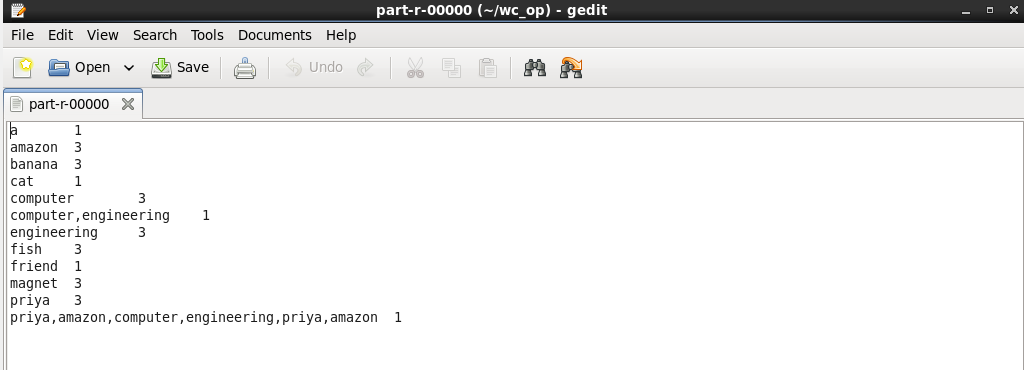
****

****

**Output:**

The output folder is generated which holds the part file.

****

****