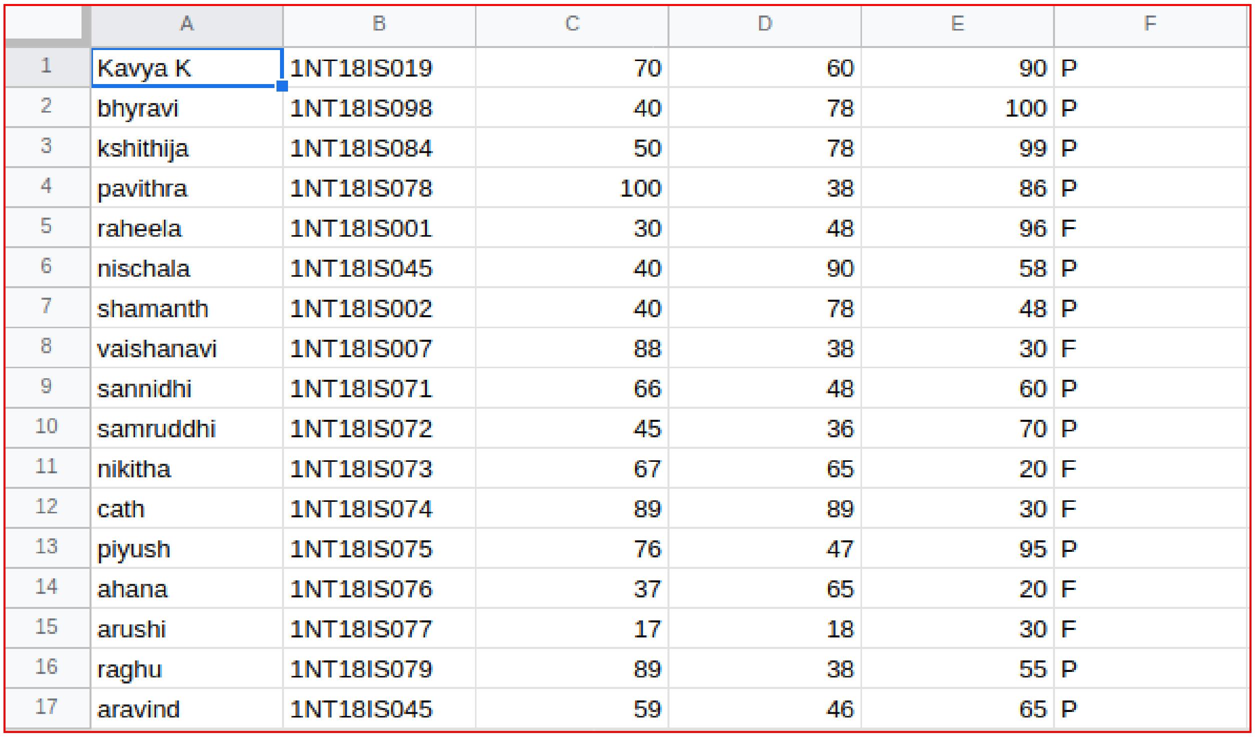
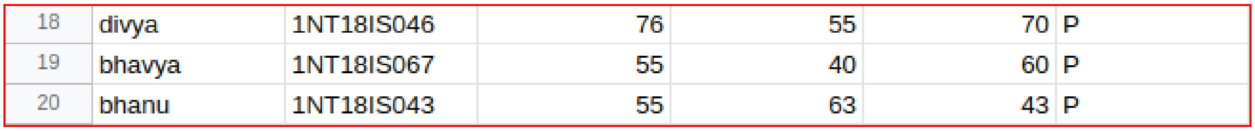
Name:Kavya K
Usn:1NT18IS078
Subject:Big data lab

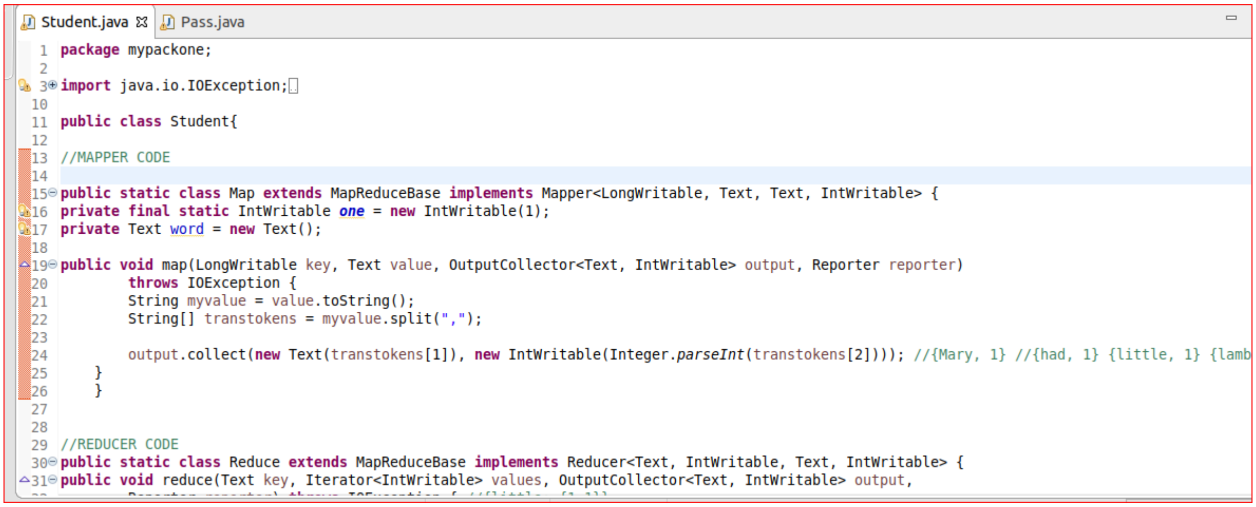
Use the Hadoop Mapreduce programming framework to compute with a program which will take the data from this .csv file and compute the following.

DATA SET:

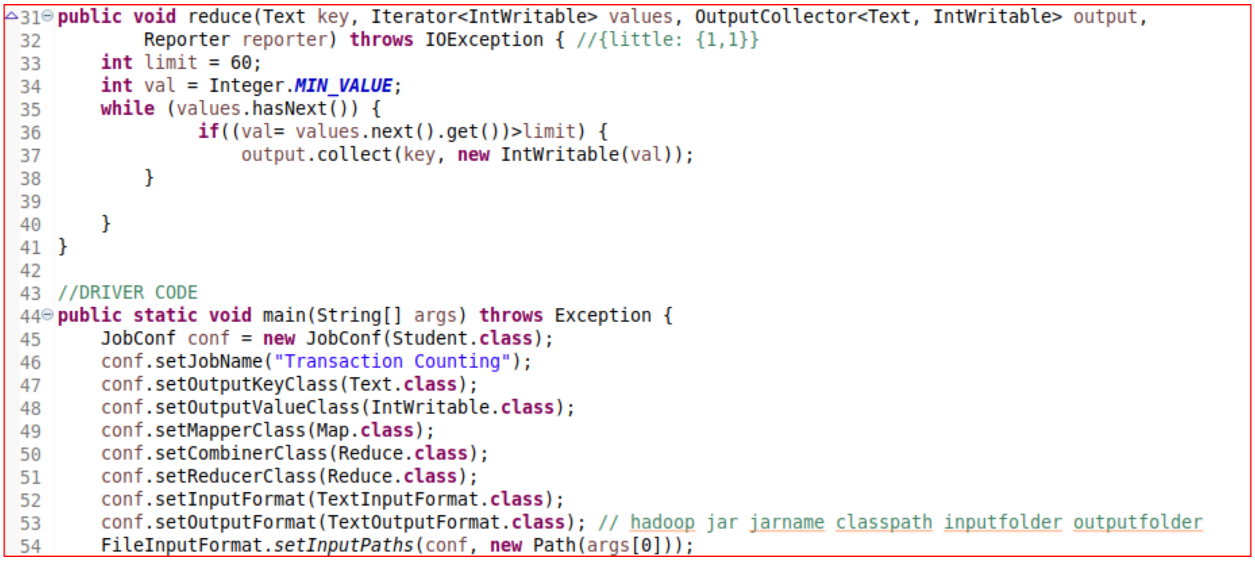



	A	B	C	D	E	F
1	Kavya K	1NT18IS019	70	60	90	P
2	bhyravi	1NT18IS098	40	78	100	P
3	kshithija	1NT18IS084	50	78	99	P
4	pavithra	1NT18IS078	100	38	86	P
5	raheela	1NT18IS001	30	48	96	F
6	nischala	1NT18IS045	40	90	58	P
7	shamanth	1NT18IS002	40	78	48	P
8	vaishanavi	1NT18IS007	88	38	30	F
9	sannidhi	1NT18IS071	66	48	60	P
10	samruddhi	1NT18IS072	45	36	70	P
11	nikitha	1NT18IS073	67	65	20	F
12	cath	1NT18IS074	89	89	30	F
13	piyush	1NT18IS075	76	47	95	P
14	ahana	1NT18IS076	37	65	20	F
15	arushi	1NT18IS077	17	18	30	F
16	raghu	1NT18IS079	89	38	55	P
17	aravind	1NT18IS045	59	46	65	P
18	divya	1NT18IS046	76	55	70	P
19	bhavya	1NT18IS067	55	40	60	P
20	bhanu	1NT18IS043	55	63	43	P

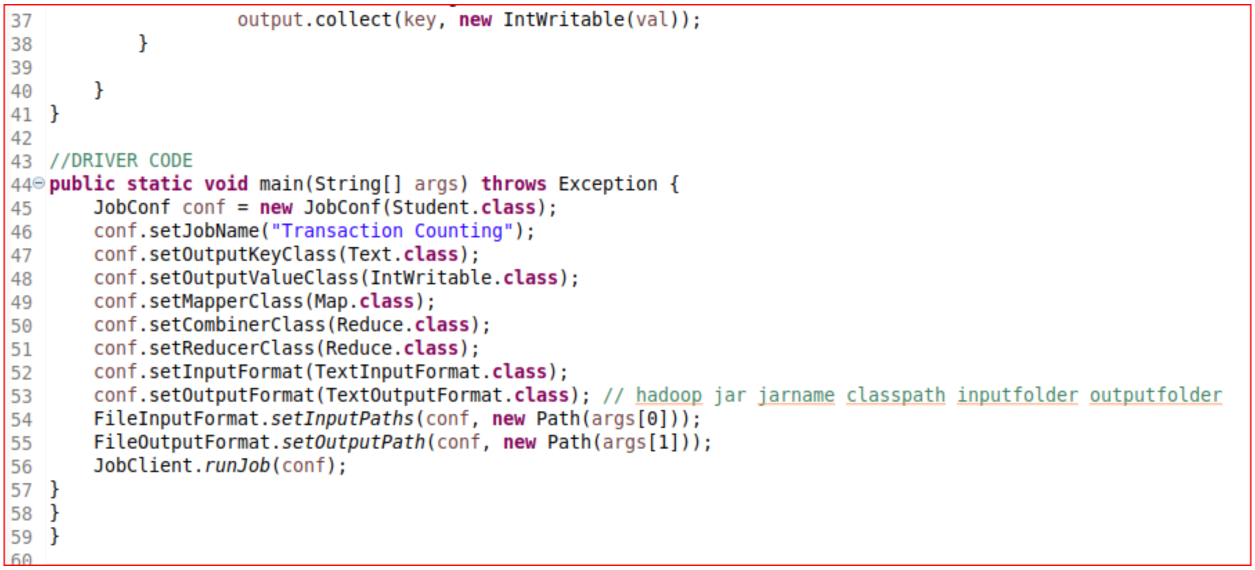
1) Total number of students who have scored more than 60 in subject1



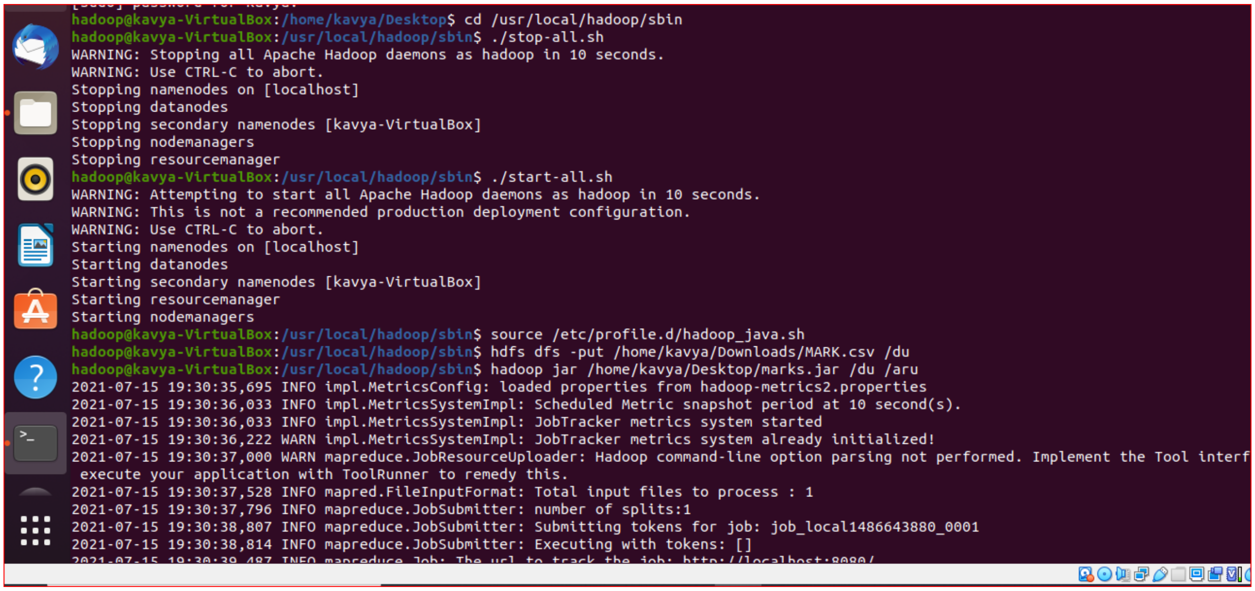
```java
package mypackone;

import java.io.IOException;

public class Student{

//MAPPER CODE

public static class Map extends MapReduceBase implements Mapper<LongWritable, Text, Text, IntWritable> {
private final static IntWritable one = new IntWritable(1);
private Text word = new Text();

public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter reporter)
        throws IOException {
        String myvalue = value.toString();
        String[] transtokens = myvalue.split(",");

        output.collect(new Text(transtokens[1]), new IntWritable(Integer.parseInt(transtokens[2]))); //{Mary, 1} //{had, 1} {little, 1} {lamb
    }
    }


//REDUCER CODE
public static class Reduce extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> {
public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable> output,
        Reporter reporter) throws IOException { //{little: {1,1}}
```



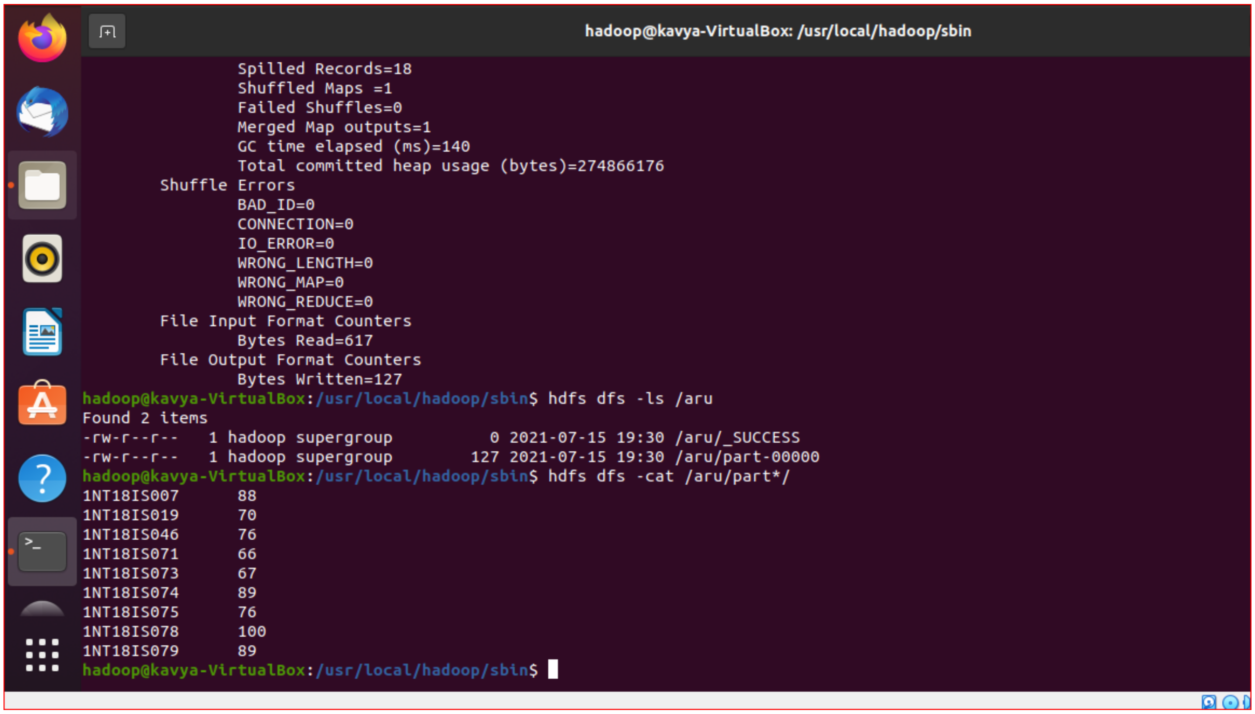
```java
public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable> output,
        Reporter reporter) throws IOException { //{little: {1,1}}
    int limit = 60;
    int val = Integer.MIN_VALUE;
    while (values.hasNext()) {
             if((val= values.next().get())>limit) {
                 output.collect(key, new IntWritable(val));
        }

    }
}

//DRIVER CODE
public static void main(String[] args) throws Exception {
    JobConf conf = new JobConf(Student.class);
    conf.setJobName("Transaction Counting");
    conf.setOutputKeyClass(Text.class);
    conf.setOutputValueClass(IntWritable.class);
    conf.setMapperClass(Map.class);
    conf.setCombinerClass(Reduce.class);
    conf.setReducerClass(Reduce.class);
    conf.setInputFormat(TextInputFormat.class);
    conf.setOutputFormat(TextOutputFormat.class); // hadoop jar jarname classpath inputfolder outputfolder
    FileInputFormat.setInputPaths(conf, new Path(args[0]));
```



```java
                output.collect(key, new IntWritable(val));
        }

    }
}

//DRIVER CODE
public static void main(String[] args) throws Exception {
    JobConf conf = new JobConf(Student.class);
    conf.setJobName("Transaction Counting");
    conf.setOutputKeyClass(Text.class);
    conf.setOutputValueClass(IntWritable.class);
    conf.setMapperClass(Map.class);
    conf.setCombinerClass(Reduce.class);
    conf.setReducerClass(Reduce.class);
    conf.setInputFormat(TextInputFormat.class);
    conf.setOutputFormat(TextOutputFormat.class); // hadoop jar jarname classpath inputfolder outputfolder
    FileInputFormat.setInputPaths(conf, new Path(args[0]));
    FileOutputFormat.setOutputPath(conf, new Path(args[1]));
    JobClient.runJob(conf);
}
}
}
```

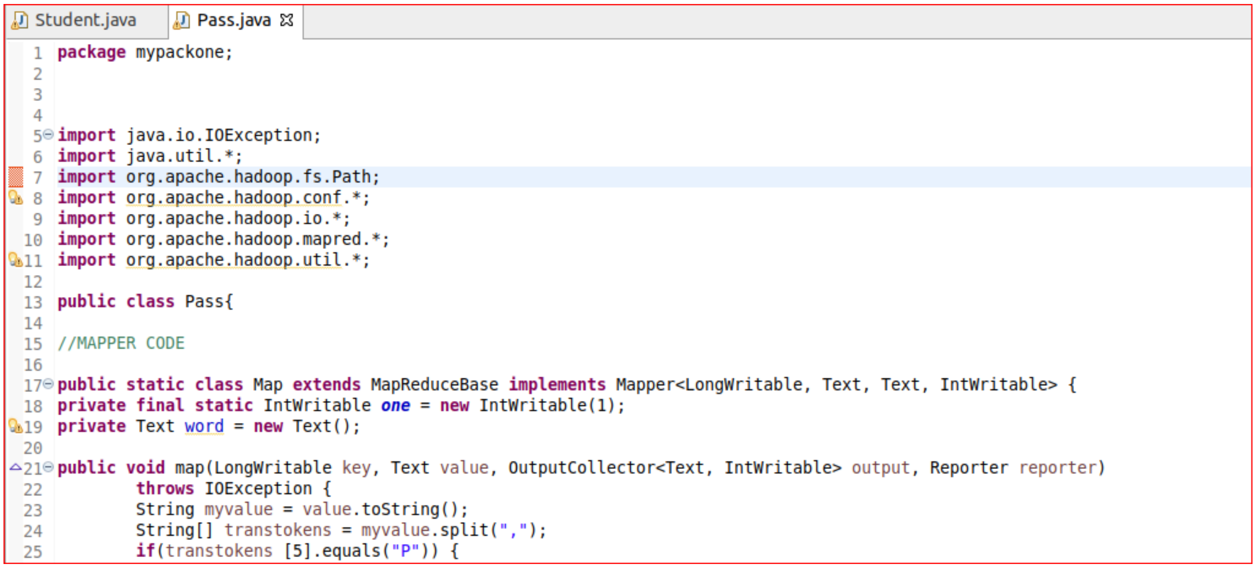


```
[sudo] password for kavya:
hadoop@kavya-VirtualBox:/home/kavya/Desktop$ cd /usr/local/hadoop/sbin
hadoop@kavya-VirtualBox:/usr/local/hadoop/sbin$ ./stop-all.sh
WARNING: Stopping all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: Use CTRL-C to abort.
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [kavya-VirtualBox]
Stopping nodemanagers
Stopping resourcemanager
hadoop@kavya-VirtualBox:/usr/local/hadoop/sbin$ ./start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [kavya-VirtualBox]
Starting resourcemanager
Starting nodemanagers
hadoop@kavya-VirtualBox:/usr/local/hadoop/sbin$ source /etc/profile.d/hadoop_java.sh
hadoop@kavya-VirtualBox:/usr/local/hadoop/sbin$ hdfs dfs -put /home/kavya/Downloads/MARK.csv /du
hadoop@kavya-VirtualBox:/usr/local/hadoop/sbin$ hadoop jar /home/kavya/Desktop/marks.jar /du /aru
2021-07-15 19:30:35,695 INFO impl.MetricsConfig: loaded properties from hadoop-metrics2.properties
2021-07-15 19:30:36,033 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2021-07-15 19:30:36,033 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2021-07-15 19:30:36,222 WARN impl.MetricsSystemImpl: JobTracker metrics system already initialized!
2021-07-15 19:30:37,000 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interf
 execute your application with ToolRunner to remedy this.
2021-07-15 19:30:37,528 INFO mapred.FileInputFormat: Total input files to process : 1
2021-07-15 19:30:37,796 INFO mapreduce.JobSubmitter: number of splits:1
2021-07-15 19:30:38,807 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1486643880_0001
2021-07-15 19:30:38,814 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-07-15 19:30:39,487 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
```

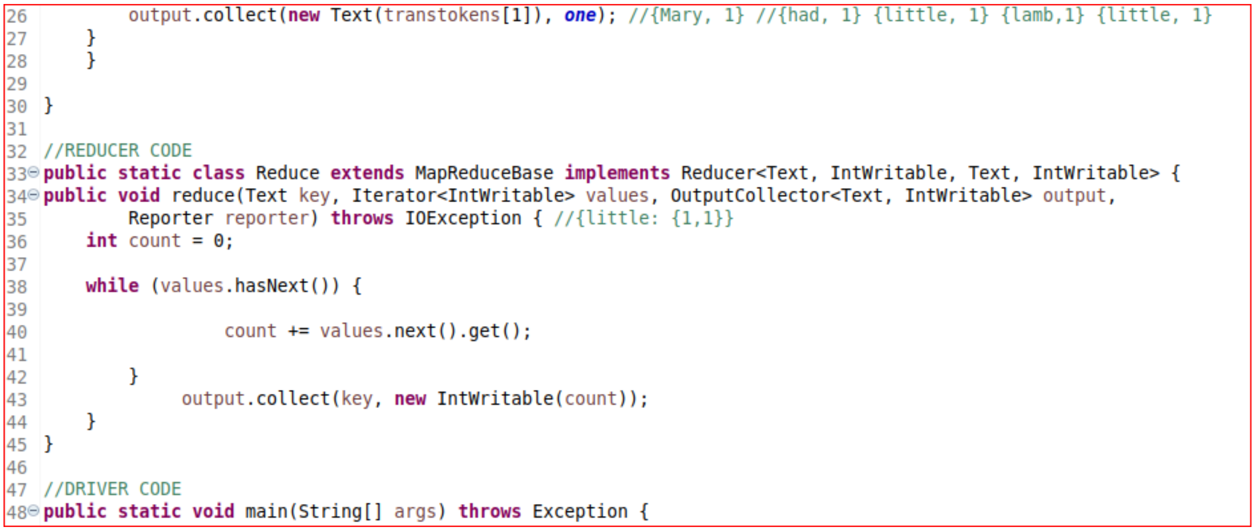


```
                Spilled Records=18
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=140
                Total committed heap usage (bytes)=274866176
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=617
        File Output Format Counters
                Bytes Written=127
hadoop@kavya-VirtualBox:/usr/local/hadoop/sbin$ hdfs dfs -ls /aru
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2021-07-15 19:30 /aru/_SUCCESS
-rw-r--r--   1 hadoop supergroup        127 2021-07-15 19:30 /aru/part-00000
hadoop@kavya-VirtualBox:/usr/local/hadoop/sbin$ hdfs dfs -cat /aru/part*/
1NT18IS007      88
1NT18IS019      70
1NT18IS046      76
1NT18IS071      66
1NT18IS073      67
1NT18IS074      89
1NT18IS075      76
1NT18IS078      100
1NT18IS079      89
hadoop@kavya-VirtualBox:/usr/local/hadoop/sbin$ 
```

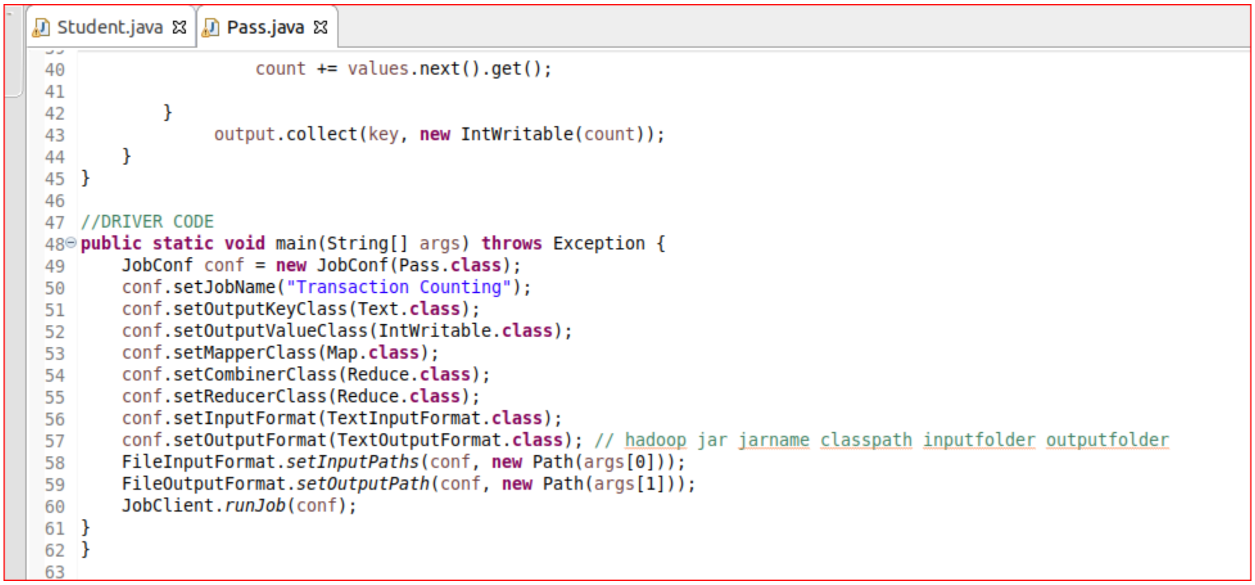
2)Total number of students who have passed in all subjects.



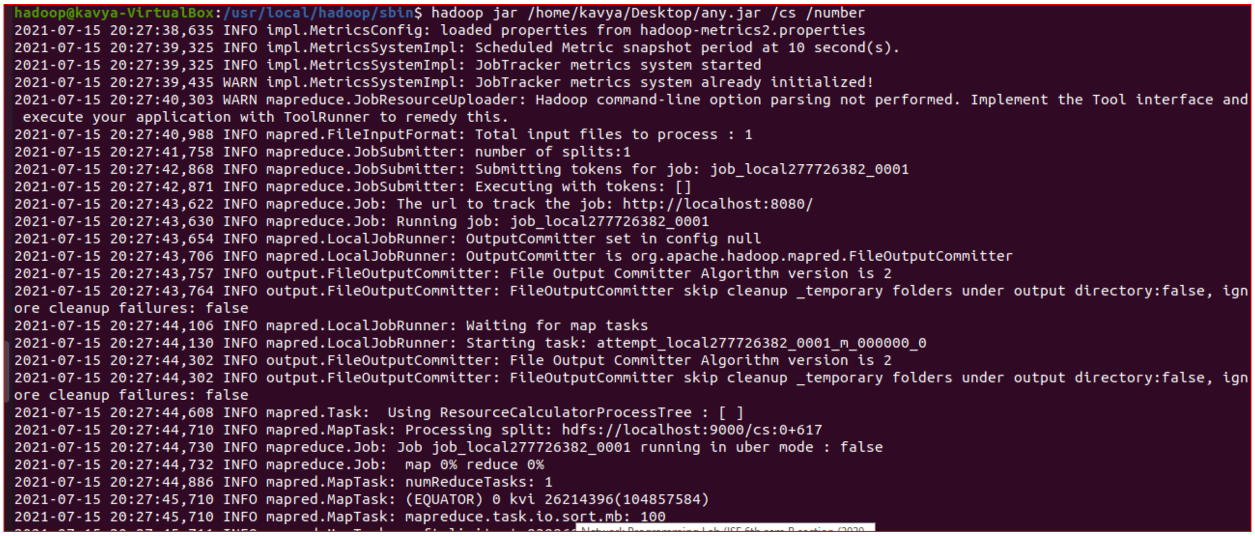
```
package mypackone;



import java.io.IOException;
import java.util.*;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.*;
import org.apache.hadoop.util.*;

public class Pass{

//MAPPER CODE

public static class Map extends MapReduceBase implements Mapper<LongWritable, Text, Text, IntWritable> {
private final static IntWritable one = new IntWritable(1);
private Text word = new Text();

public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter reporter)
        throws IOException {
        String myvalue = value.toString();
        String[] transtokens = myvalue.split(",");
        if(transtokens [5].equals("P")) {
```



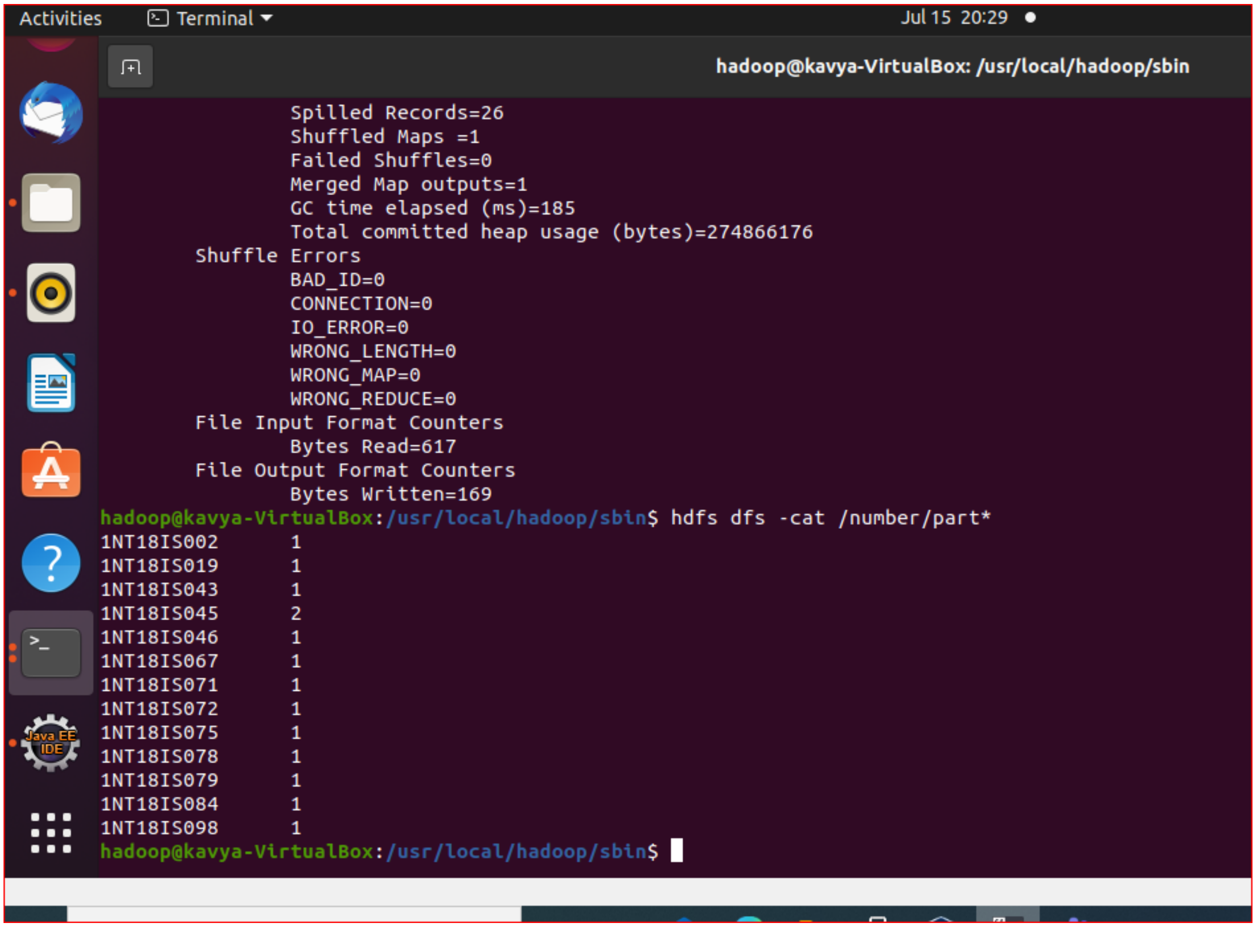
```
        output.collect(new Text(transtokens[1]), one); //{Mary, 1} //{had, 1} {little, 1} {lamb,1} {little, 1}
    }
    }

}

//REDUCER CODE
public static class Reduce extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> {
public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable> output,
        Reporter reporter) throws IOException { //{little: {1,1}}
    int count = 0;

    while (values.hasNext()) {

                 count += values.next().get();

        }
             output.collect(key, new IntWritable(count));
    }
}

//DRIVER CODE
public static void main(String[] args) throws Exception {
```



```
                 count += values.next().get();

        }
             output.collect(key, new IntWritable(count));
    }
}

//DRIVER CODE
public static void main(String[] args) throws Exception {
    JobConf conf = new JobConf(Pass.class);
    conf.setJobName("Transaction Counting");
    conf.setOutputKeyClass(Text.class);
    conf.setOutputValueClass(IntWritable.class);
    conf.setMapperClass(Map.class);
    conf.setCombinerClass(Reduce.class);
    conf.setReducerClass(Reduce.class);
    conf.setInputFormat(TextInputFormat.class);
    conf.setOutputFormat(TextOutputFormat.class); // hadoop jar jarname classpath inputfolder outputfolder
    FileInputFormat.setInputPaths(conf, new Path(args[0]));
    FileOutputFormat.setOutputPath(conf, new Path(args[1]));
    JobClient.runJob(conf);
}
}
```



```
hadoop@kavya-VirtualBox:/usr/local/hadoop/sbin$ hadoop jar /home/kavya/Desktop/any.jar /cs /number
2021-07-15 20:27:38,635 INFO impl.MetricsConfig: loaded properties from hadoop-metrics2.properties
2021-07-15 20:27:39,325 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2021-07-15 20:27:39,325 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2021-07-15 20:27:39,435 WARN impl.MetricsSystemImpl: JobTracker metrics system already initialized!
2021-07-15 20:27:40,303 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and
 execute your application with ToolRunner to remedy this.
2021-07-15 20:27:40,988 INFO mapred.FileInputFormat: Total input files to process : 1
2021-07-15 20:27:41,758 INFO mapreduce.JobSubmitter: number of splits:1
2021-07-15 20:27:42,868 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local277726382_0001
2021-07-15 20:27:42,871 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-07-15 20:27:43,622 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2021-07-15 20:27:43,630 INFO mapreduce.Job: Running job: job_local277726382_0001
2021-07-15 20:27:43,654 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2021-07-15 20:27:43,706 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapred.FileOutputCommitter
2021-07-15 20:27:43,757 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2021-07-15 20:27:43,764 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ign
ore cleanup failures: false
2021-07-15 20:27:44,106 INFO mapred.LocalJobRunner: Waiting for map tasks
2021-07-15 20:27:44,130 INFO mapred.LocalJobRunner: Starting task: attempt_local277726382_0001_m_000000_0
2021-07-15 20:27:44,302 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2021-07-15 20:27:44,302 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ign
ore cleanup failures: false
2021-07-15 20:27:44,608 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2021-07-15 20:27:44,710 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/cs:0+617
2021-07-15 20:27:44,730 INFO mapreduce.Job: Job job_local277726382_0001 running in uber mode : false
2021-07-15 20:27:44,732 INFO mapreduce.Job:  map 0% reduce 0%
2021-07-15 20:27:44,886 INFO mapred.MapTask: numReduceTasks: 1
2021-07-15 20:27:45,710 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2021-07-15 20:27:45,710 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
```



```
                Spilled Records=26
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=185
                Total committed heap usage (bytes)=274866176
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=617
        File Output Format Counters
                Bytes Written=169
hadoop@kavya-VirtualBox:/usr/local/hadoop/sbin$ hdfs dfs -cat /number/part*
1NT18IS002      1
1NT18IS019      1
1NT18IS043      1
1NT18IS045      2
1NT18IS046      1
1NT18IS067      1
1NT18IS071      1
1NT18IS072      1
1NT18IS075      1
1NT18IS078      1
1NT18IS079      1
1NT18IS084      1
1NT18IS098      1
hadoop@kavya-VirtualBox:/usr/local/hadoop/sbin$
```