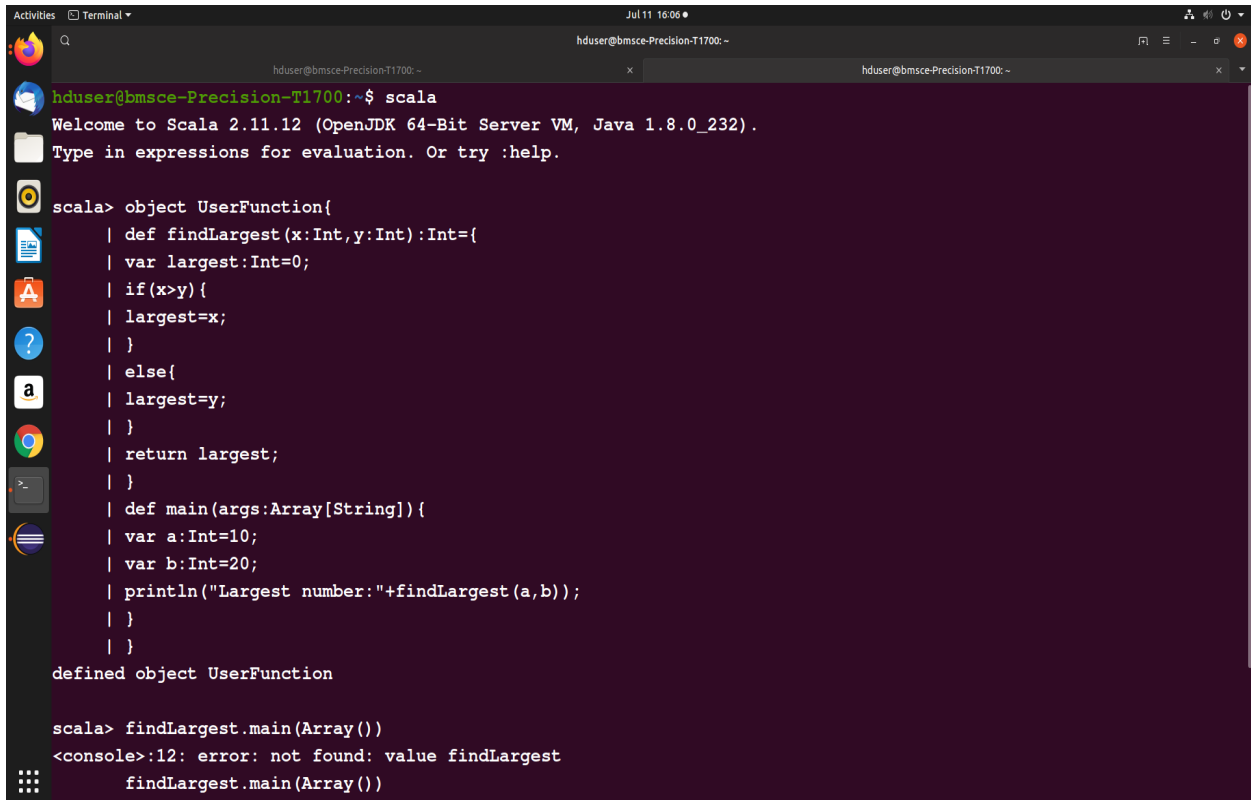


a)Scala program to create a user defined function to return the largest number among two numbers.



```
Activities Terminal Jul 11 16:06
hduser@bmsce-Precision-T1700: ~
hduser@bmsce-Precision-T1700: ~$ scala
Welcome to Scala 2.11.12 (OpenJDK 64-Bit Server VM, Java 1.8.0_232).
Type in expressions for evaluation. Or try :help.

scala> object UserFunction{
|   def findLargest(x:Int,y:Int):Int={
|   |   var largest:Int=0;
|   |   if(x>y){
|   |   |   largest=x;
|   |   }
|   |   else{
|   |   |   largest=y;
|   |   }
|   |   return largest;
|   }
|   def main(args:Array[String]){
|   |   var a:Int=10;
|   |   var b:Int=20;
|   |   println("Largest number:"+findLargest(a,b));
|   }
| }
defined object UserFunction

scala> findLargest.main(Array())
<console>:12: error: not found: value findLargest
    findLargest.main(Array())
```

```
Activities Terminal Jul 11 16:06 • hduser@bmsce-Precision-T1700: ~
| var largest:Int=0;
| if(x>y){
|   largest=x;
| }
| else{
|   largest=y;
| }
| return largest;
| }
| def main(args:Array[String]){
|   var a:Int=10;
|   var b:Int=20;
|   println("Largest number:"+findLargest(a,b));
| }
| }
defined object UserFunction

scala> findLargest.main(Array())
<console>:12: error: not found: value findLargest
    findLargest.main(Array())
    ^

scala> UserFunction.main(Array())
Largest number:20

scala>
```

b)Code: To print the sum of squares of list of given numbers

Driver Class:

```
import java.io.IOException;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;

public class SqDriver extends Configured implements Tool {
    public int run(String args[]) throws IOException
    {
        if (args.length < 2)
        {
            System.out.println("Please give valid inputs");
            return -1;
        }
    }
}
```

```

JobConf conf = new JobConf(SqDriver.class);
FileInputFormat.setInputPaths(conf, new Path(args[0]));
FileOutputFormat.setOutputPath(conf, new Path(args[1]));
conf.setMapperClass(SqMapper.class);
conf.setReducerClass(SqReducer.class);
conf.setMapOutputKeyClass(Text.class);
conf.setMapOutputValueClass(IntWritable.class);
conf.setOutputKeyClass(Text.class);
conf.setOutputValueClass(IntWritable.class);
JobClient.runJob(conf);
return 0;
}
// Main Method
public static void main(String args[]) throws Exception
{
int exitCode = ToolRunner.run(new SqDriver(), args);
System.out.println(exitCode);
}
}

```

Mapper Class

```

import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;

import org.apache.hadoop.mapred.Reporter;
public class SqMapper extends MapReduceBase implements
Mapper<LongWritable, Text, Text, IntWritable> {
// Map function
public void map(LongWritable key, Text value, OutputCollector<Text,
IntWritable> output, Reporter rep) throws IOException
{

String line = value.toString();
// Splitting the line on spaces
for (String num : line.split(" "))
{
int n = Integer.parseInt(num);
int square = n * n;
output.collect(new Text("sum"), new IntWritable(square));

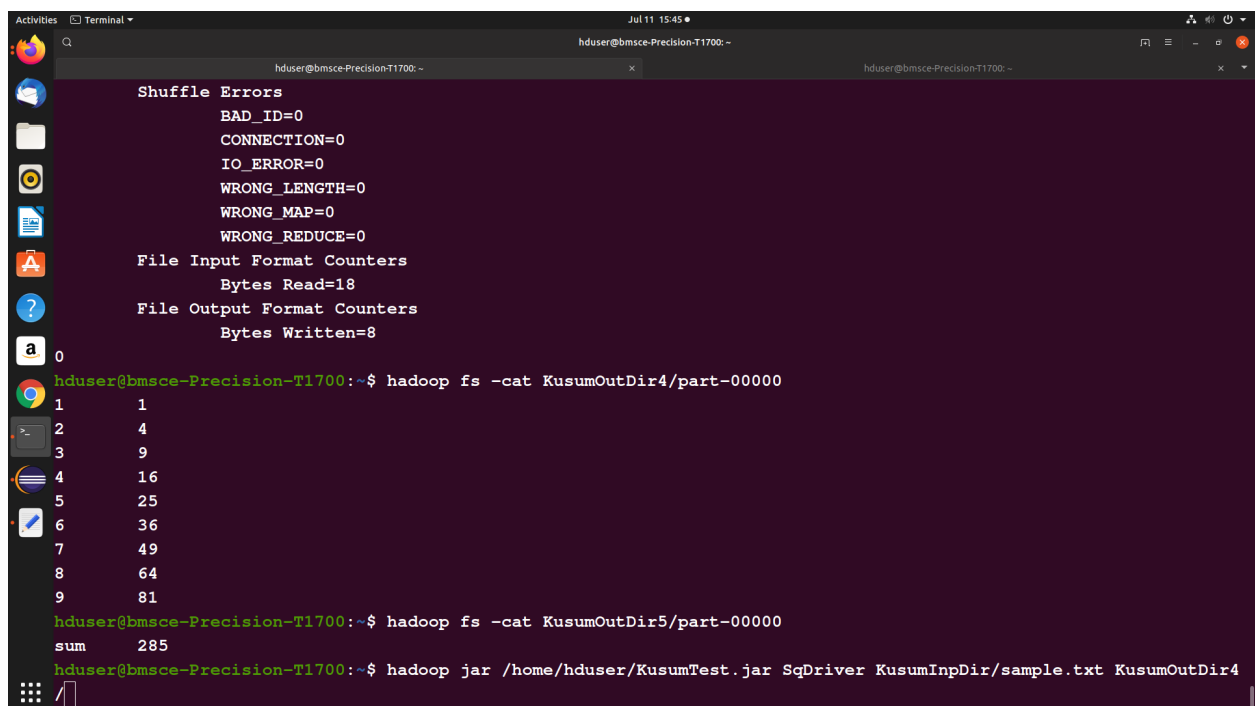
}
}
}

```

Reducer Class

```
import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;

public class SqReducer extends MapReduceBase implements
Reducer<Text,IntWritable, Text, IntWritable> {
// Reduce function
public void reduce(Text key, Iterator<IntWritable> value,
OutputCollector<Text, IntWritable> output,
Reporter rep) throws IOException
{
int sum = 0;
while (value.hasNext())
{
IntWritable i = value.next();
sum += i.get();
}
output.collect(key, new IntWritable(sum));
}
}
```



The screenshot shows a terminal window with the following content:

```
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=18
File Output Format Counters
Bytes Written=8
0
hduser@bmsce-Precision-T1700:~$ hadoop fs -cat KusumOutDir4/part-00000
1
2 4
3 9
4 16
5 25
6 36
7 49
8 64
9 81
hduser@bmsce-Precision-T1700:~$ hadoop fs -cat KusumOutDir5/part-00000
sum 285
hduser@bmsce-Precision-T1700:~$ hadoop jar /home/hduser/KusumTest.jar SqDriver KusumInpDir/sample.txt KusumOutDir4
```