

Q1:

Ans:

Mapper:

```
import java.io.IOException;
```

```
import java.util.StringTokenizer;
```

```
import org.apache.hadoop.io.DoubleWritable;
```

```
import org.apache.hadoop.io.IntWritable;
```

```
import org.apache.hadoop.io.LongWritable;
```

```
import org.apache.hadoop.io.Text;
```

```
import org.apache.hadoop.mapreduce.Mapper;
```

```
public class MyMapper extends Mapper<LongWritable, Text, Text, IntWritable>
```

```
{
```

```
    @Override
```

```
    protected void map(LongWritable key, Text value,
```

```
                      org.apache.hadoop.mapreduce.Mapper.Context context)
```

```
        throws IOException, InterruptedException {
```

```
        String inputstring = value.toString();
```

```
        for(String x : inputstring.split(" "))
```

```
{
```

```
            context.write(new Text(x),new IntWritable(1));
```

```
}
```

```
}
```

```
}
```

Reducer:

```
import java.io.IOException;
```

```
import org.apache.hadoop.io.IntWritable;
```

```
import org.apache.hadoop.io.Text;
```

```
import org.apache.hadoop.mapreduce.Reducer;
```

```
import org.apache.hadoop.mapreduce.Reducer.Context;
```

```

public class MyReducer extends Reducer<Text, IntWritable, Text, IntWritable>{

    @Override
    public void reduce(Text key, Iterable<IntWritable> values, Context context)
    throws IOException, InterruptedException {
        int y = 0;
        for(IntWritable x : values)
        {
            y ++;
        }
        context.write(key, new IntWritable(y) );
    }
}

```

Driver:

```

import java.io.IOException;
import java.net.URI;
import java.net.URISyntaxException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FileSystem;
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.FileInputFormat;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class MyDriver {
    public static void main(String[] args) throws IOException,
    ClassNotFoundException, InterruptedException, URISyntaxException
    {
        Configuration conf = new Configuration();
    }
}

```

```
j.setJobName("My First Job");  
j.setJarByClass(MyDriver.class );  
j.setMapperClass(MyMapper.class );
```

```
j.setReducerClass(MyReducer.class);
```

```
j.setOutputKeyClass(Text.class);  
j.setOutputValueClass(IntWritable.class);
```

```
FileInputFormat.addInputPath(j, new Path(args[0]));  
FileOutputFormat.setOutputPath(j, new Path(args[1]));
```

```
URI uri = new URI(args[1].toString());
```

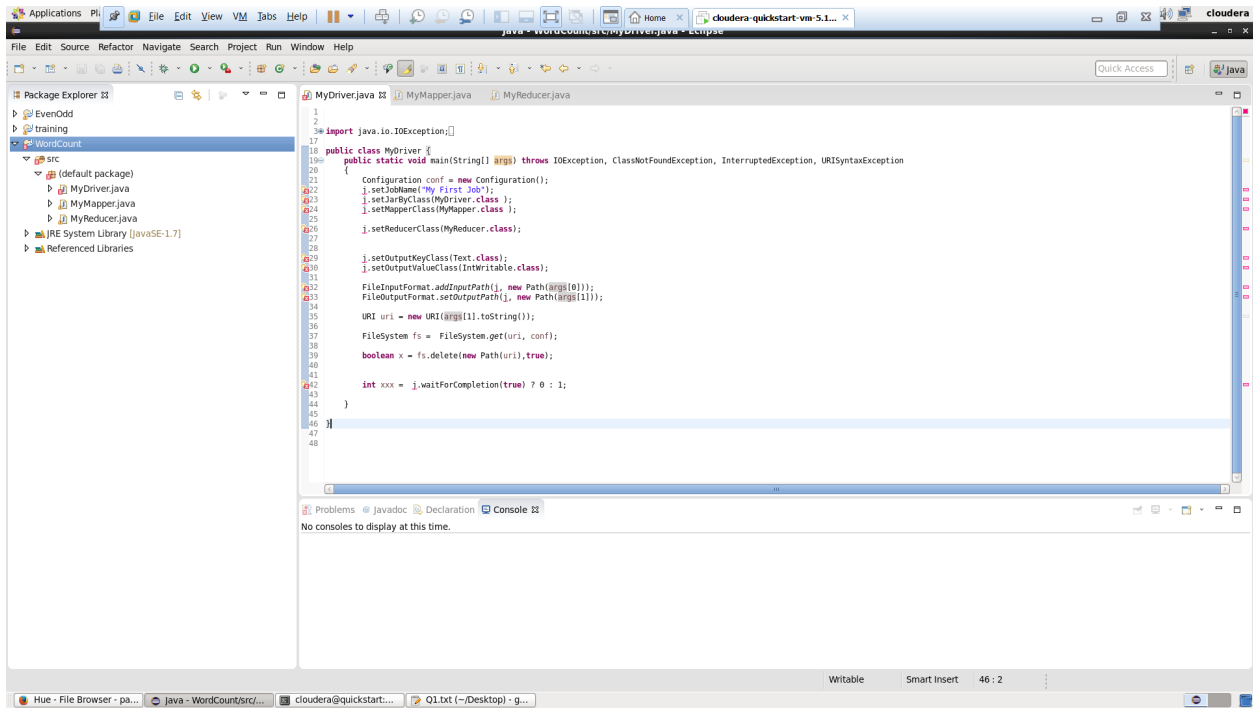
```
FileSystem fs = FileSystem.get(uri, conf);
```

```
boolean x = fs.delete(new Path(uri),true);
```

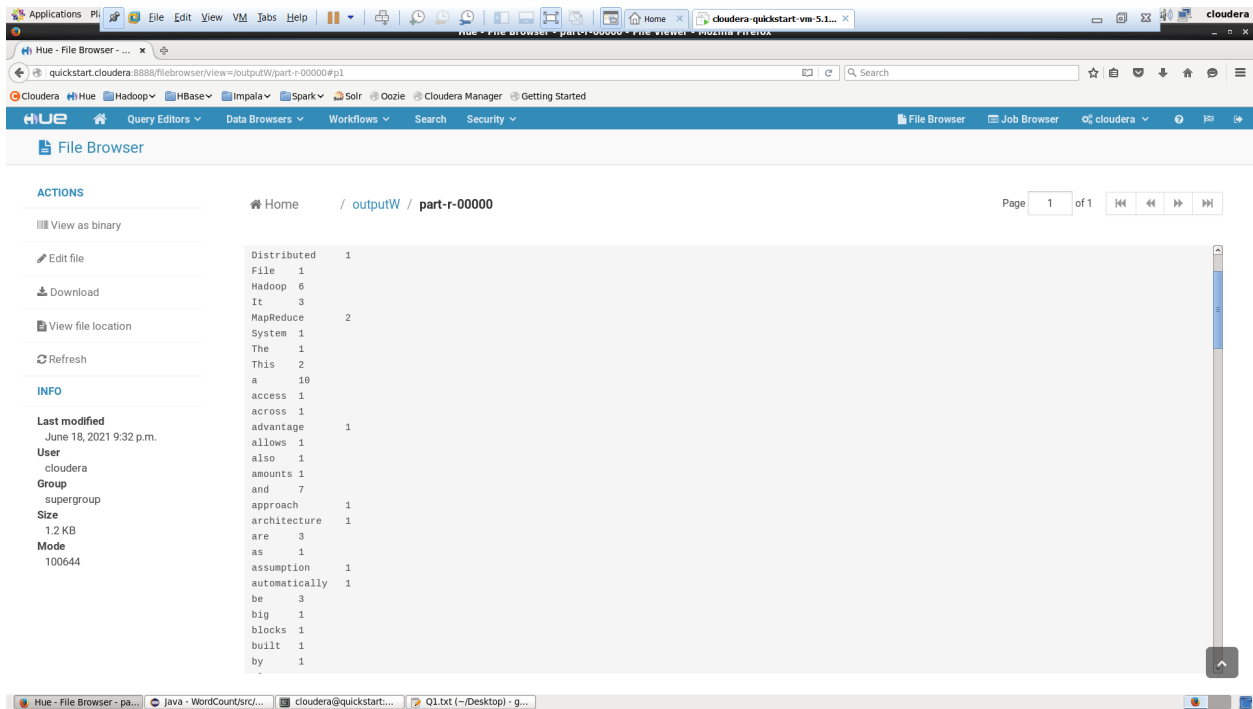
```
int xxx = j.waitForCompletion(true) ? 0 : 1;
```

```
}
```

```
}
```



OUTPUT:



Q2-

Ans:

Mapper:

```
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class MyMapper extends Mapper<LongWritable, Text, Text, IntWritable>
{

    public void map(LongWritable key, Text value, Context context)throws
    java.io.IOException, InterruptedException
    {
        String data[]=value.toString().split(",");    //data = [85 131 993 392 689....]

        for(String num:data)
        {
            int number=Integer.parseInt(num);
            if((number%2)==0)
            {
                context.write(new Text("EVEN"), new IntWritable(number));
            }

        }

    }
}
```

Reducer:

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class MyReducer extends Reducer<Text, IntWritable, Text, IntWritable>
{

    // reducer will receive key value pair as ODD [ 85 131 993 491 539 985 413.....
```

```
//      EVEN [ 392 870 240 888 184 494 704 996 408....
```

```
    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws
IOException, InterruptedException
    {
        int sum = 0;
        if(key.equals("ODD"))
        {
            for (IntWritable value : values)
            {
                sum += value.get();
            }
        }
        else
        {
            for (IntWritable value : values)
            {
                sum += value.get();
            }
        }
        context.write(key,new IntWritable(sum));;
    }
}
```

Driver:

```
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class MyDriver
{
    public static void main(String[] args) throws
IOException,ClassNotFoundException, InterruptedException
    {
```

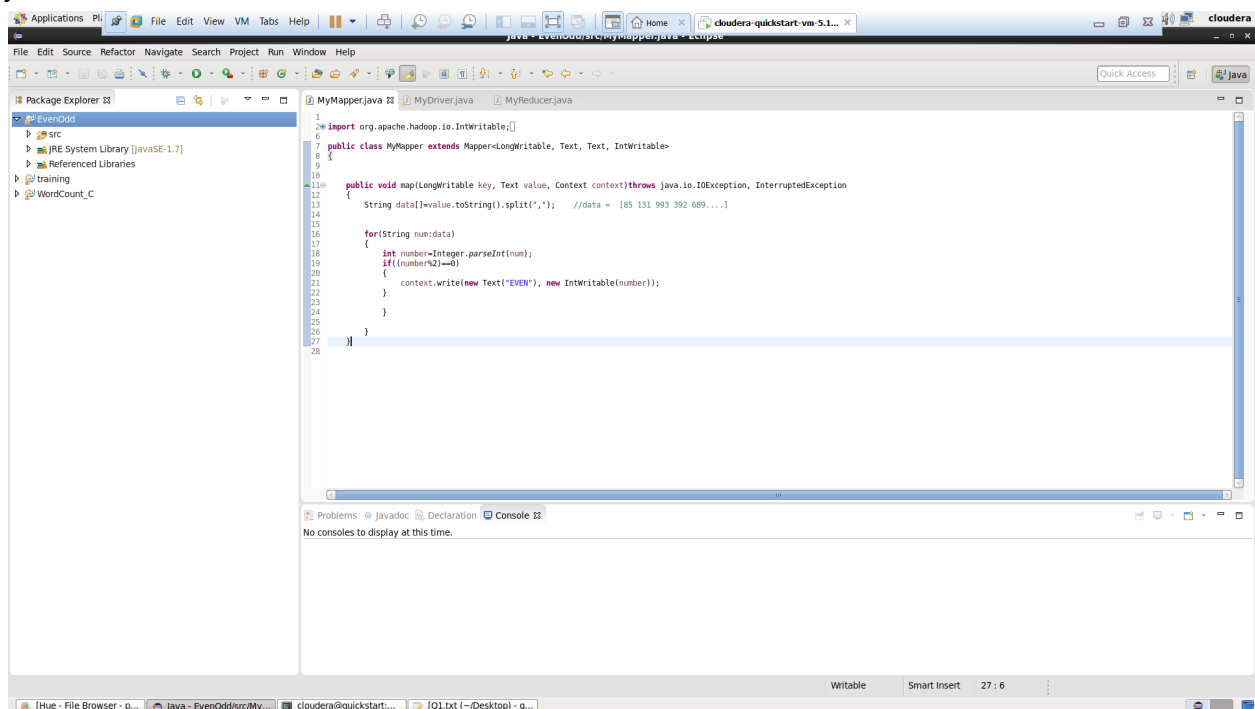
```
Configuration conf = new Configuration();
Job job = new Job(conf, "Evenodd");
```

```
job.setJarByClass(MyDriver.class);
job.setMapperClass(MyMapper.class);
job.setReducerClass(MyReducer.class);
job.setMapOutputKeyClass(Text.class);
job.setMapOutputValueClass(IntWritable.class);
```

```
FileInputFormat.addInputPath(job, new Path(args[0]));
FileOutputFormat.setOutputPath(job, new Path(args[1]));
```

```
System.exit(job.waitForCompletion(true) ? 0 : 1);
```

```
}
}
```



OUTPUT:

The screenshot displays the Hue File Browser interface within a web browser window. The browser's address bar shows the URL: `quickstart.cloudera:8888/filebrowser/view=outputE/part-r-00000`. The Hue interface includes a top navigation bar with various tool icons and a sidebar on the left with the following sections:

- ACTIONS**
 - View as binary
 - Edit file
 - Download
 - View file location
 - Refresh
- INFO**
 - Last modified**: June 18, 2021 8:58 p.m.
 - User**: cloudera
 - Group**: supergroup
 - Size**: 12 bytes
 - Mode**: 100644

The main content area shows the file details for `part-r-00000` under the path `outputE`. The file content is displayed as a single line: `EVEN 559536`. The interface also features a top status bar with the Hue logo and navigation links, and a bottom taskbar with open applications.

