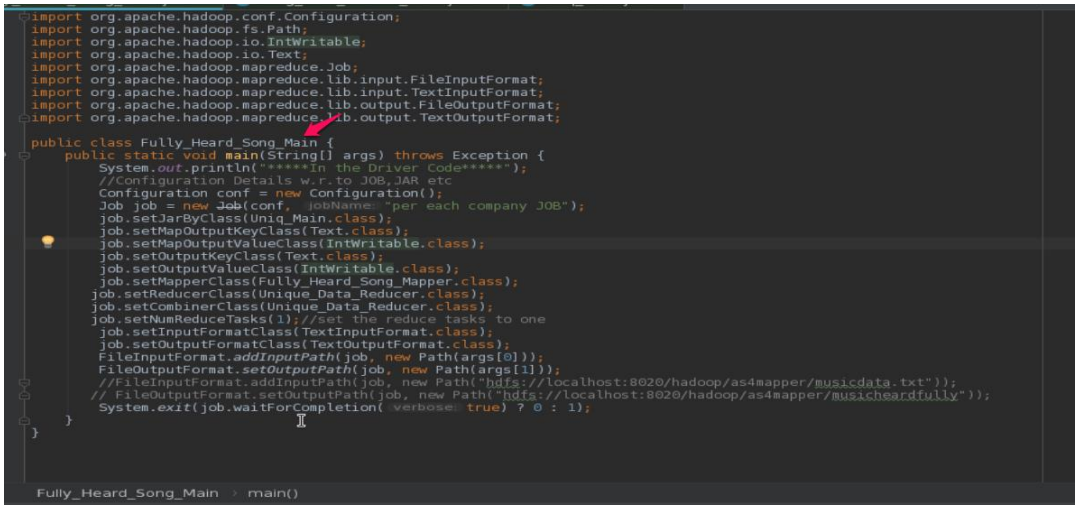
AS-4 MR CODE

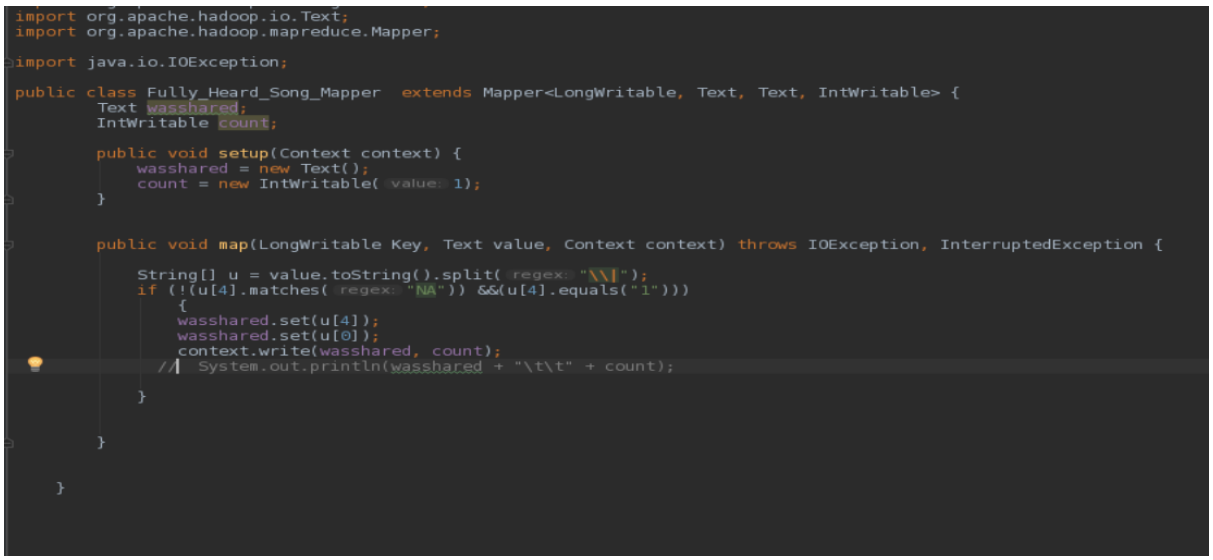
1. What are the number of times a song was heard fully.

Driver class



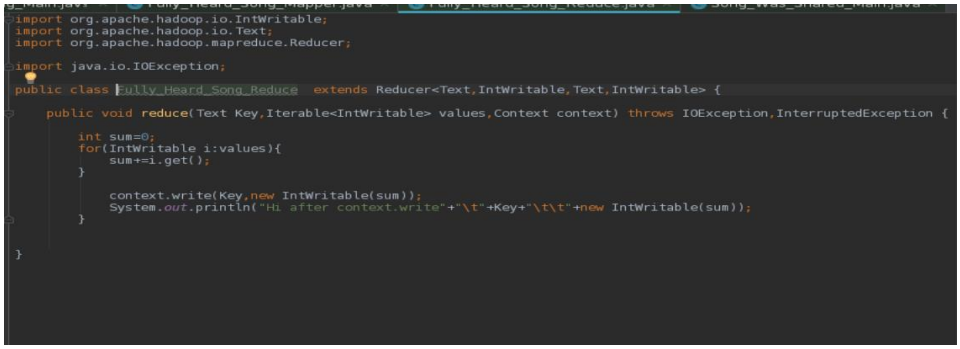
```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;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

public class Fully_Heard_Song_Main {
    public static void main(String[] args) throws Exception {
        System.out.println("****In the Driver Code*****");
        //Configuration Details w.r.to JOB,JAR etc
        Configuration conf = new Configuration();
        Job job = new Job(conf, jobName: "per each company JOB");
        job.setJarByClass(Uniq_Main.class);
        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(IntWritable.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        job.setMapperClass(Fully_Heard_Song_Mapper.class);
        job.setReducerClass(Unique_Data_Reducer.class);
        job.setCombinerClass(Unique_Data_Reducer.class);
        job.setNumReduceTasks(1);//set the reduce tasks to one
        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        //FileInputFormat.addInputPath(job, new Path("hdfs://localhost:8020/hadoop/as4mapper/musicdata.txt"));
        // FileOutputFormat.setOutputPath(job, new Path("hdfs://localhost:8020/hadoop/as4mapper/musicheardfully"));
        System.exit(job.waitForCompletion( verbose: true) ? 0 : 1);
    }
}

Fully_Heard_Song_Main › main()
```

Mapper Class



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

import java.io.IOException;

public class Fully_Heard_Song_Mapper  extends Mapper<LongWritable, Text, Text, IntWritable> {
        Text wasshared;
        IntWritable count;

        public void setup(Context context) {
            wasshared = new Text();
            count = new IntWritable( value: 1);
        }


        public void map(LongWritable Key, Text value, Context context) throws IOException, InterruptedException {

            String[] u = value.toString().split( regex: "\\|");
            if (!(u[4].matches( regex: "NA")) &&(u[4].equals("1")))
                {
                wasshared.set(u[4]);
                wasshared.set(u[0]);
                context.write(wasshared, count);
              //  System.out.println(wasshared + "\t\t" + count);

            }


        }


    }
```

Reducer Class



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

import java.io.IOException;

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

    public void reduce(Text Key,Iterable<IntWritable> values,Context context) throws IOException,InterruptedException {

        int sum=0;
        for(IntWritable i:values){
            sum+=i.get();
        }

            context.write(Key,new IntWritable(sum));
            System.out.println("Hi after context.write"+"\t"+Key+"\t\t"+new IntWritable(sum));
        }

}
```

INPUT

111115|222|0|1|0

111113|225|1|0|0

111117|223|0|1|1

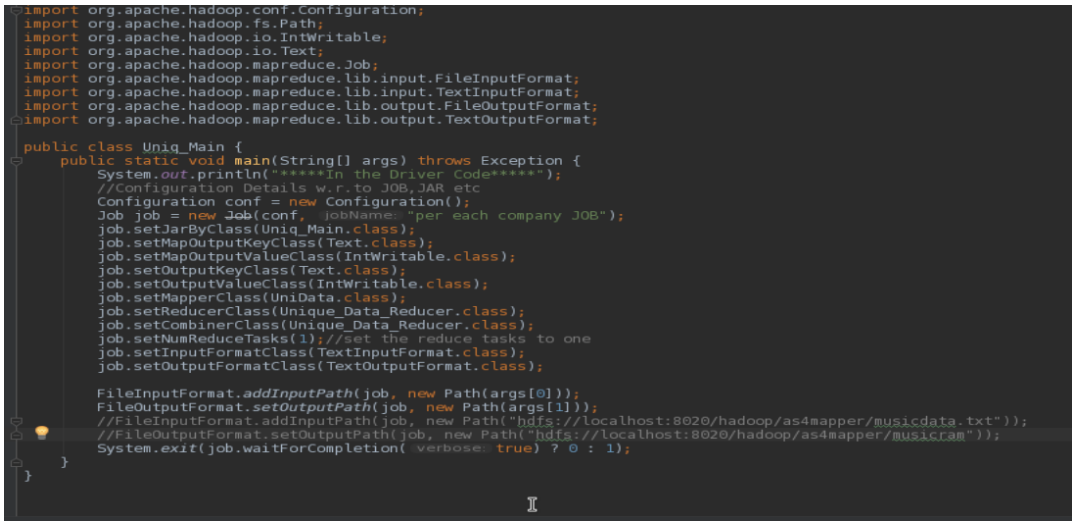
111115|225|1|0|0

OUTPUT

```
acadgild@localhost ~]$ hadoop fs -cat /hadoop/as4mapper/musicheardfully/part-r-00000
18/05/24 23:49:53 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platfor
isses where applicable
111117  1
```

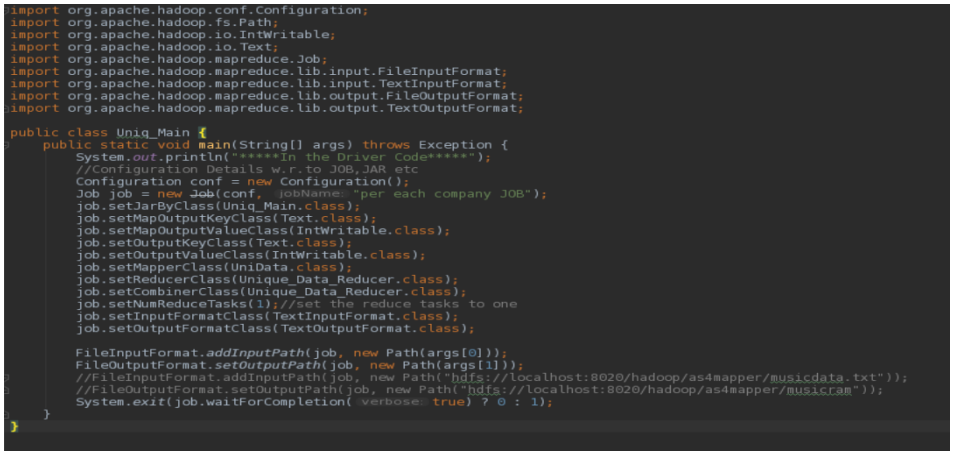
2. Find the number of unique listeners in the data set.

Driver Class



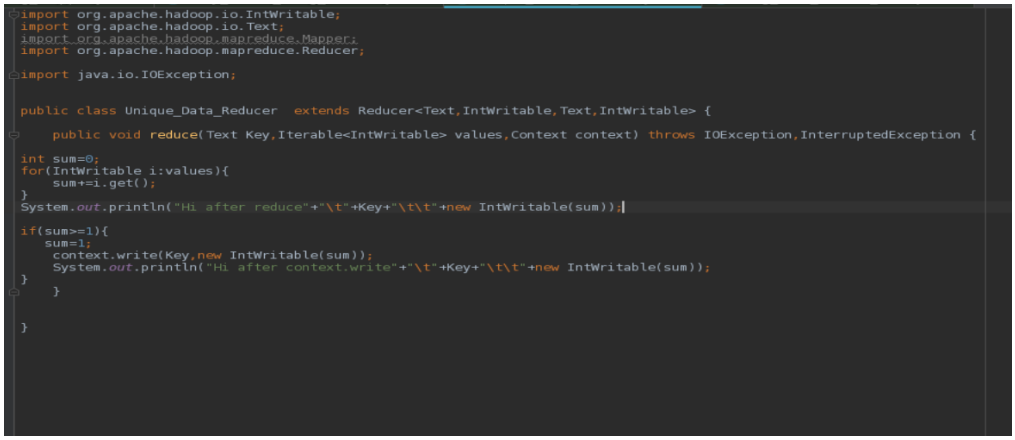
```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;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

public class Uniq_Main {
    public static void main(String[] args) throws Exception {
        System.out.println("*****In the Driver Code*****");
        //Configuration Details w.r.to JOB,JAR etc
        Configuration conf = new Configuration();
        Job job = new Job(conf,  jobName: "per each company JOB");
        job.setJarByClass(Uniq_Main.class);
        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(IntWritable.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        job.setMapperClass(UniData.class);
        job.setReducerClass(Unique_Data_Reducer.class);
        job.setCombinerClass(Unique_Data_Reducer.class);
        job.setNumReduceTasks(1);//set the reduce tasks to one
        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        //FileInputFormat.addInputPath(job, new Path("hdfs://localhost:8020/hadoop/as4mapper/musicdata.txt"));
        //FileOutputFormat.setOutputPath(job, new Path("hdfs://localhost:8020/hadoop/as4mapper/musicram"));
        System.exit(job.waitForCompletion( verbose: true) ? 0 : 1);
    }
}
```

Mapper Class



```
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.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class Uniq_Main {
    public static void main(String[] args) throws Exception {
        System.out.println("*****In the Driver Code*****");
        //Configuration Details w.r.to JOB,JAR etc
        Configuration conf = new Configuration();
        Job job = new Job(conf,  jobName: "per each company JOB");
        job.setJarByClass(Uniq_Main.class);
        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(IntWritable.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        job.setMapperClass(UniData.class);
        job.setReducerClass(Unique_Data_Reducer.class);
        job.setCombinerClass(Unique_Data_Reducer.class);
        job.setNumReduceTasks(1);//set the reduce tasks to one
        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        //FileInputFormat.addInputPath(job, new Path("hdfs://localhost:8020/hadoop/as4mapper/musicdata.txt"));
        //FileOutputFormat.setOutputPath(job, new Path("hdfs://localhost:8020/hadoop/as4mapper/musicram"));
        System.exit(job.waitForCompletion( verbose: true) ? 0 : 1);
    }
}
```

Reducer Class



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

import java.io.IOException;


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

    public void reduce(Text Key,Iterable<IntWritable> values,Context context) throws IOException,InterruptedException {

int sum=0;
for(IntWritable i:values){
    sum+=i.get();
}
System.out.println("Hi after reduce"+"\t"+Key+"\t\t"+new IntWritable(sum));

if(sum>=1){
   sum=1;
    context.write(Key,new IntWritable(sum));
    System.out.println("Hi after context.write"+"\t"+Key+"\t\t"+new IntWritable(sum));
}
    }


}
```

INPUT

111115|222|0|1|0

111113|225|1|0|0

111117|223|0|1|1

111115|225|1|0|0

OUTPUT OF UNIQUE DATA

```
[acadgild@localhost ~]$ hadoop fs -cat /hadoop/as4mapper/musicram/part-r-00000
18/05/25 00:01:30 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
111113  1
111115  1
111117  1
```

3. What are the number of times a song was shared.

Driver Class

```
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.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class Song_Was_Shared_Main {
    public static void main(String[] args) throws Exception {
        System.out.println("*****In the Driver Code*****");
        //Configuration Details w.r.to JOB,JAR etc
        Configuration conf = new Configuration();
        Job job = new Job(conf,  jobName: "per each company JOB");
        job.setJarByClass(Uniq_Main.class);
        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(IntWritable.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        job.setMapperClass(Song_Was_Shared_Mapper.class);
        job.setReducerClass(Song_Was_Shared_Reducer.class);
        job.setCombinerClass(Song_Was_Shared_Reducer.class);
        job.setNumReduceTasks(1);//set the reduce tasks to one
        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
       //  FileInputFormat.addInputPath(job, new Path("hdfs://localhost:8020/hadoop/as4mapper/musicdata.txt"));
        // FileOutputFormat.setOutputPath(job, new Path("hdfs://localhost:8020/hadoop/as4mapper/musicwasshared"));
        System.exit(job.waitForCompletion( verbose: true) ? 0 : 1);
    }
}
```

Mapper Class

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

import java.io.IOException;

public class Song_Was_Shared_Mapper extends Mapper<LongWritable, Text, Text, IntWritable> {
    Text fullsong;
    IntWritable count;

    public void setup(Context context) {
        fullsong = new Text();
        count = new IntWritable( value: 1);
    }


    public void map(LongWritable Key, Text value, Context context) throws IOException, InterruptedException {

        String[] u = value.toString().split( regex: "\\|");
        if (!(u[2].matches( regex: "NA")) &&(u[2].equals("1")))
        {
            fullsong.set(u[2]);
            fullsong.set(u[0]);
            context.write(fullsong, count);
            System.out.println(fullsong + "\t\t" + count);

        }


    }

}
```

Reducer Class

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

import java.io.IOException;

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

    public void reduce(Text Key,Iterable<IntWritable> values,Context context) throws IOException,InterruptedException {

        int sum=0;
        for(IntWritable i:values){
            sum+=i.get();
        }

        context.write(Key,new IntWritable(sum));
        System.out.println("Hi after context.write"+"\t"+Key+"\t\t"+new IntWritable(sum));
    }


}
```

INPUT

111115|222|0|1|0

111113|225|1|0|0

111117|223|0|1|1

111115|225|1|0|0

OUTPUT

```
18/05/25 00:07:38 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-jav
asses where applicable
111113  1
111115  1
```