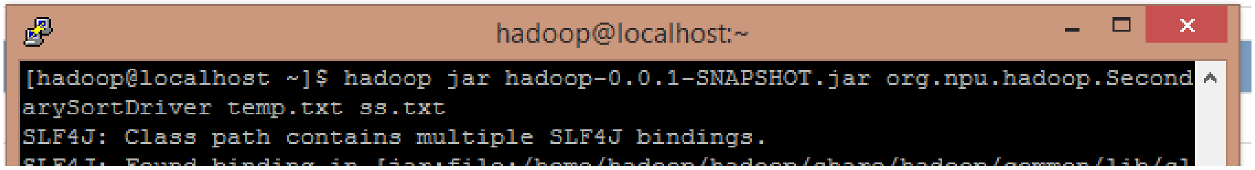
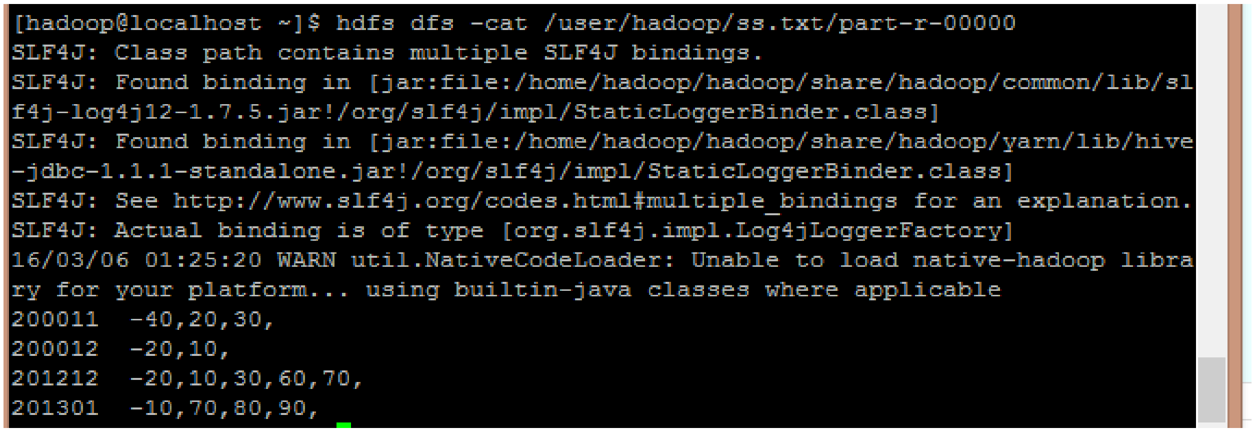
Name:Renu Mariam Mathew
Student Id:16376
Hw#3

1.step1



```
[hadoop@localhost ~]$ hadoop jar hadoop-0.0.1-SNAPSHOT.jar org.npu.hadoop.Second
arySortDriver temp.txt ss.txt
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/common/lib/sl
```

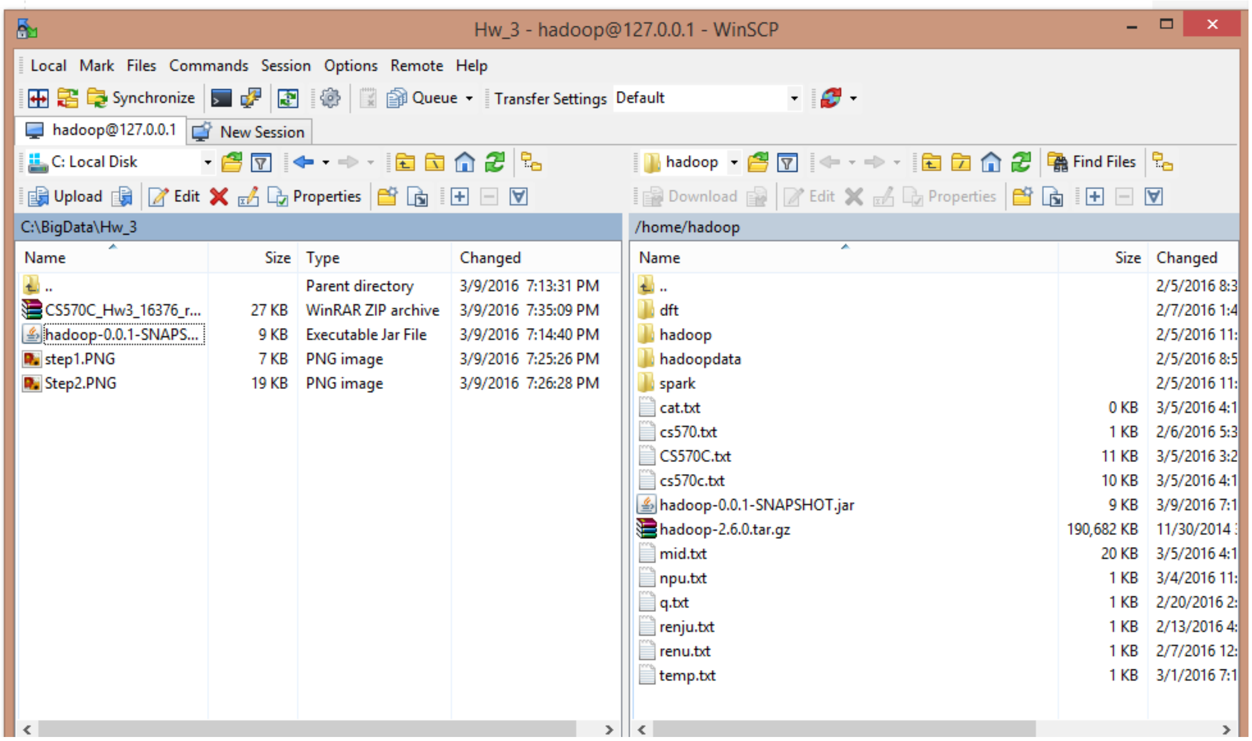
2.Step 2 output



```
[hadoop@localhost ~]$ hdfs dfs -cat /user/hadoop/ss.txt/part-r-00000
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/common/lib/sl
f4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/yarn/lib/hive
-jdbc-1.1.1-standalone.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
16/03/06 01:25:20 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
200011  -40,20,30,
200012  -20,10,
201212  -20,10,30,60,70,
201301  -10,70,80,90,
```

Name:Renu Mariam Mathew
Student Id:16376
Hw#3

3.



C:\BigData\Hw_3

Name	Size	Type	Changed
..		Parent directory	3/9/2016 7:13:31 PM
CS570C_Hw3_16376_r...	27 KB	WinRAR ZIP archive	3/9/2016 7:35:09 PM
hadoop-0.0.1-SNAPS...	9 KB	Executable Jar File	3/9/2016 7:14:40 PM
step1.PNG	7 KB	PNG image	3/9/2016 7:25:26 PM
Step2.PNG	19 KB	PNG image	3/9/2016 7:26:28 PM

/home/hadoop

Name	Size	Changed
..		2/5/2016 8:3
dft		2/7/2016 1:4
hadoop		2/5/2016 11:
hadoopdata		2/5/2016 8:5
spark		2/5/2016 11:
cat.txt	0 KB	3/5/2016 4:1
cs570.txt	1 KB	2/6/2016 5:3
CS570C.txt	11 KB	3/5/2016 3:2
cs570c.txt	10 KB	3/5/2016 4:1
hadoop-0.0.1-SNAPSHOT.jar	9 KB	3/9/2016 7:1
hadoop-2.6.0.tar.gz	190,682 KB	11/30/2014
mid.txt	20 KB	3/5/2016 4:1
npu.txt	1 KB	3/4/2016 11:
q.txt	1 KB	2/20/2016 2:
renju.txt	1 KB	2/13/2016 4:
renu.txt	1 KB	2/7/2016 12:
temp.txt	1 KB	3/1/2016 7:1

4.Code was changed only in the DateTemperaturePair class only

,

I have highlighted the code where I have changed

```
 package org.npu.hadoop;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.io.Writable;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.WritableComparable;


import java.io.DataInput;

import java.io.DataOutput;

import java.io.IOException;
```

Name:Renu Mariam Mathew
Student Id:16376
Hw#3

```
/**
 * The DateTemperaturePair class enable us to represent a
 * composite type of (yearMonth, day, temperature). To persist
 * a composite type (actually any data type) in Hadoop, it has
 * to implement the org.apache.hadoop.io.Writable interface.
 *
 * To compare composite types in Hadoop, it has to implement
 * the org.apache.hadoop.io.WritableComparable interface.
 *
 *
 */
public class DateTemperaturePair
    implements Writable, WritableComparable<DateTemperaturePair> {

    private Text yearMonth = new Text();
    private Text day = new Text();
    private IntWritable temperature = new IntWritable();



    public DateTemperaturePair() {
    }

    public DateTemperaturePair(String yearMonth, String day, int temperature) {
        this.yearMonth.set(yearMonth);
        this.day.set(day);
        this.temperature.set(temperature);
    }


    public static DateTemperaturePair read(DataInput in) throws IOException {
        DateTemperaturePair pair = new DateTemperaturePair();
        pair.readFields(in);
        return pair;
    }


    public void write(DataOutput out) throws IOException {
        yearMonth.write(out);
        day.write(out);
        temperature.write(out);
    }


    public void readFields(DataInput in) throws IOException {
        yearMonth.readFields(in);
        day.readFields(in);
        temperature.readFields(in);
    }


    public int compareTo(DateTemperaturePair pair) {
        int compareValue = this.yearMonth.compareTo(pair.getYearMonth());
        if (compareValue == 0) {
            compareValue = temperature.compareTo(pair.getTemperature());
        }
         return compareValue;       // to sort ascending
```

Name:Renu Mariam Mathew
Student Id:16376
Hw#3

```
//have commented the descending order
    //return -1*compareValue;     // to sort descending
  }


  public Text getYearMonthDay() {
    return new Text(yearMonth.toString()+day.toString());
  }


  public Text getYearMonth() {
    return yearMonth;
  }


  public Text getDay() {
    return day;
  }


  public IntWritable getTemperature() {
    return temperature;
  }


  public void setYearMonth(String yearMonthAsString) {
    yearMonth.set(yearMonthAsString);
  }


  public void setDay(String dayAsString) {
    day.set(dayAsString);
  }
```

Name:Renu Mariam Mathew
Student Id:16376
Hw#3

```
public void setTemperature(int temp) {
  temperature.set(temp);
}

@Override
public boolean equals(Object o) {
  if (this == o) {
    return true;
  }
  if (o == null || getClass() != o.getClass()) {
    return false;
  }

  DateTemperaturePair that = (DateTemperaturePair) o;
  if (temperature != null ? !temperature.equals(that.temperature) : that.temperature != null) {
    return false;
  }
  if (yearMonth != null ? !yearMonth.equals(that.yearMonth) : that.yearMonth != null) {
    return false;
  }

  return true;
}

@Override
public int hashCode() {
        int result = yearMonth != null ? yearMonth.hashCode() : 0;

        result = 31 * result + (temperature != null ? temperature.hashCode() : 0);

        return result;

    }


    @Override

    public String toString() {

            StringBuilder builder = new StringBuilder();

            builder.append("DateTemperaturePair{yearMonth=");

            builder.append(yearMonth);

            builder.append(", day=");

            builder.append(day);

            builder.append(", temperature=");

            builder.append(temperature);

            builder.append("}");

            return builder.toString();

    }

}
```