REG NO:210701509

EXP 3: Map Reduce program to process a weather dataset.

AIM:

To implement MapReduce program to process a weather dataset.

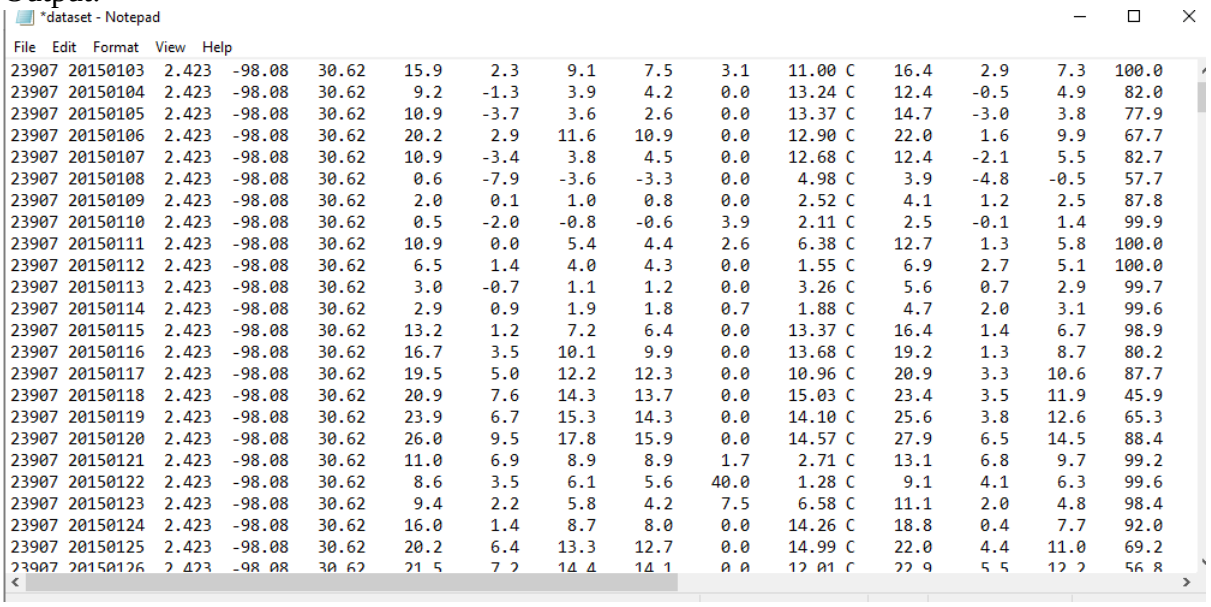
Procedure:

Step 1: Create Data File:

Create a file named "word_count_data.txt" and populate it with text data that you wish to analyse.
Login with your hadoop user.

Download the dataset (weather data)

Output:



Step 2: Mapper Logic - mapper.py:

Create a file named "mapper.py" to implement the logic for the mapper. The mapper will read input data from STDIN, split lines into words, and output each word with its count.

```
nano mapper.py
# Copy and paste the mapper.py code
```

```
#!/usr/bin/env python

import sys

# input comes from STDIN (standard input)
# the mapper will get daily max temperature and group it by month. so output will be (month,dailymax_temperature)
```

```
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    words = line.split()
    #See the README hosted on the weather website which help us understand how each
position represents a column
    month = line[10:12]
    daily_max = line[38:45]
    daily_max = daily_max.strip()
    # increase counters
    for word in words:
        # write the results to STDOUT (standard output);
        # what we output here will be go through the shuffle proess and then
        # be the input for the Reduce step, i.e. the input for reducer.py
        #
        # tab-delimited; month and daily max temperature as output
        print ('%s\t%s' % (month ,daily_max))

          .
```

Step 3: Reducer Logic - reducer.py:

Create a file named "reducer.py" to implement the logic for the reducer. The reducer will aggregate the occurrences of each word and generate the final output.

```
nano reducer.py
# Copy and paste the reducer.py code
```

reducer.py

```
#!/usr/bin/env python

from operator import itemgetter
import sys
#reducer will get the input from stdid which will be a collection of key, value(Key=month ,
value= daily max temperature)
#reducer logic: will get all the daily max temperature for a month and find max temperature
for the month
#shuffle will ensure that key are sorted(month)
current_month = None
current_max = 0
month = None

# input comes from STDIN
for line in sys.stdin:
```

```
    # remove leading and trailing whitespace
    line = line.strip()
    # parse the input we got from mapper.py
    month, daily_max = line.split('\t', 1)

    # convert daily_max (currently a string) to float
    try:
       daily_max = float(daily_max)
    except ValueError:
       # daily_max was not a number, so silently
       # ignore/discard this line
       continue

    # this IF-switch only works because Hadoop shuffle process sorts map output
    # by key (here: month) before it is passed to the reducer
    if current_month == month:
       if daily_max > current_max:
          current_max = daily_max
    else:
       if current_month:
          # write result to STDOUT
          print ('%s\t%s' % (current_month, current_max))
       current_max = daily_max
       current_month = month

 # output of the last month
 if current_month == month:
    print ('%s\t%s' % (current_month, current_max))
```

Step 4: Prepare Hadoop Environment:

Start the Hadoop daemons and create a directory in HDFS to store your data.

```
start-all.sh
```

Step 6: Make Python Files Executable:

Give executable permissions to your mapper.py and reducer.py files.

```
chmod 777 mapper.py reducer.py
```

Step 7: Run the program using Hadoop Streaming:

Download the latest hadoop-streaming jar file and place it in a location you can easily access.

Then run the program using Hadoop Streaming.

```
hadoop fs -mkdir -p /weatherdata

hadoop fs -copyFromLocal /home/sx/Downloads/dataset.txt /weatherdata

hdfs dfs -ls /weatherdata

hadoop jar /home/sx/hadoop-3.2.3/share/hadoop/tools/lib/hadoop-streaming-3.2.3.jar \
  -input /weatherdata/dataset.txt \
  -output /weatherdata/output \
  -file "/home/sx/Downloads/mapper.py" \
  -mapper "python3 mapper.py" \
  -file "/home/sx/Downloads/reducer.py" \
  -reducer "python3 reducer.py"

hdfs dfs -text /weatherdata/output/* > /home/sx/Downloads/outputfile.txt
```

Step 8: Check Output:

Check the output of the program in the specified HDFS output directory.

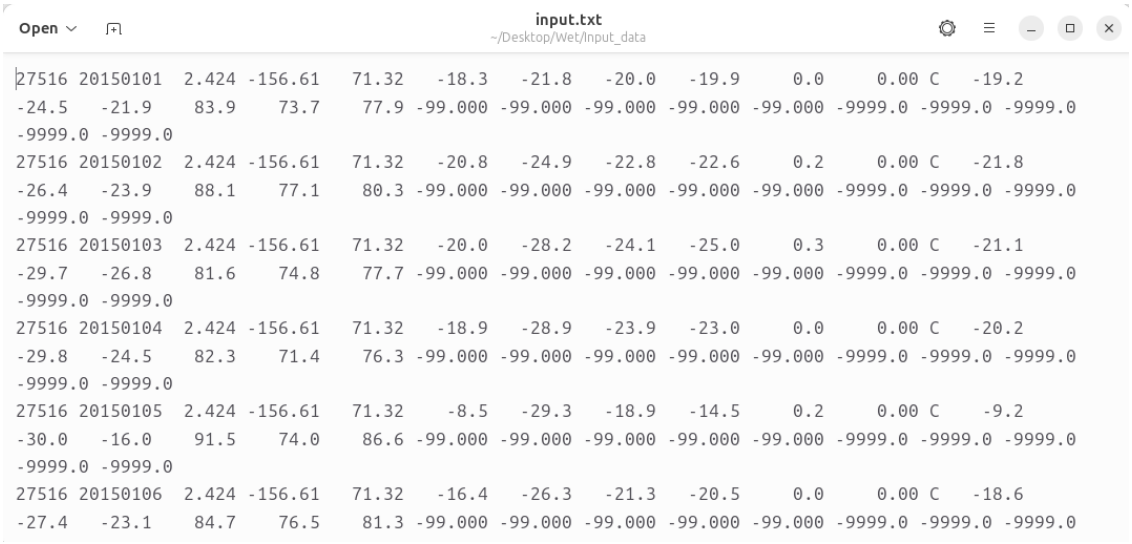
```
hdfs dfs -text /weatherdata/output/* > /home/sx/Downloads/output/
/part-00000
```

```
01      26.5
02      26.6
03      29.1
04      30.8
05      31.1
06      33.6
07      38.5
08      40.2
09      36.5
10      36.9
11      27.6
12      25.9
```

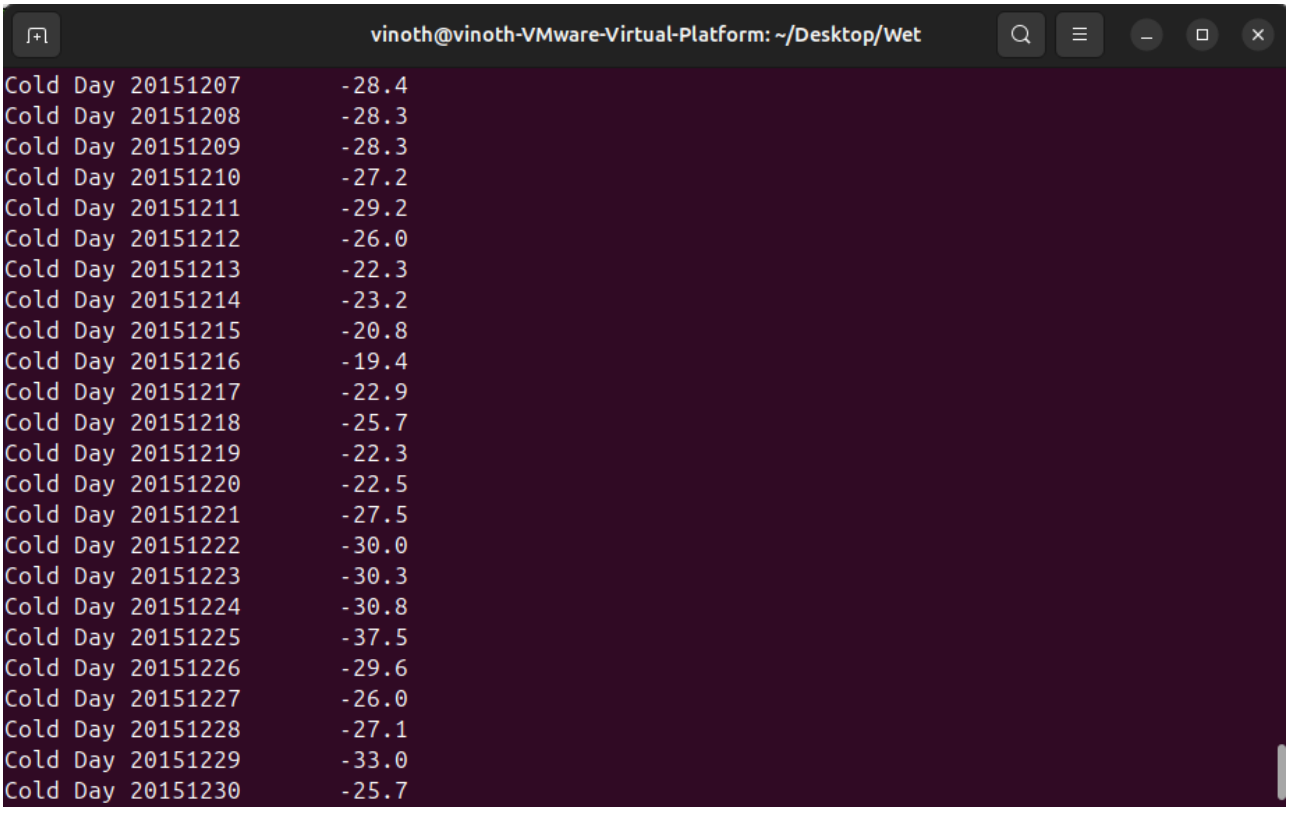
After copy and paste the above output in your local file give the below command to remove the directory from hdfs :

```
hadoop fs -rm -r /weatherdata/output
```

Output:




```
27516 20150101  2.424 -156.61   71.32   -18.3   -21.8   -20.0   -19.9     0.0     0.00 C   -19.2
-24.5   -21.9    83.9    73.7    77.9 -99.000 -99.000 -99.000 -99.000 -99.000 -9999.0 -9999.0 -9999.0
-9999.0 -9999.0
27516 20150102  2.424 -156.61   71.32   -20.8   -24.9   -22.8   -22.6     0.2     0.00 C   -21.8
-26.4   -23.9    88.1    77.1    80.3 -99.000 -99.000 -99.000 -99.000 -99.000 -9999.0 -9999.0 -9999.0
-9999.0 -9999.0
27516 20150103  2.424 -156.61   71.32   -20.0   -28.2   -24.1   -25.0     0.3     0.00 C   -21.1
-29.7   -26.8    81.6    74.8    77.7 -99.000 -99.000 -99.000 -99.000 -99.000 -9999.0 -9999.0 -9999.0
-9999.0 -9999.0
27516 20150104  2.424 -156.61   71.32   -18.9   -28.9   -23.9   -23.0     0.0     0.00 C   -20.2
-29.8   -24.5    82.3    71.4    76.3 -99.000 -99.000 -99.000 -99.000 -99.000 -9999.0 -9999.0 -9999.0
-9999.0 -9999.0
27516 20150105  2.424 -156.61   71.32    -8.5   -29.3   -18.9   -14.5     0.2     0.00 C    -9.2
-30.0   -16.0    91.5    74.0    86.6 -99.000 -99.000 -99.000 -99.000 -99.000 -9999.0 -9999.0 -9999.0
-9999.0 -9999.0
27516 20150106  2.424 -156.61   71.32   -16.4   -26.3   -21.3   -20.5     0.0     0.00 C   -18.6
-27.4   -23.1    84.7    76.5    81.3 -99.000 -99.000 -99.000 -99.000 -99.000 -9999.0 -9999.0 -9999.0
```




```
Cold Day 20151207        -28.4
Cold Day 20151208        -28.3
Cold Day 20151209        -28.3
Cold Day 20151210        -27.2
Cold Day 20151211        -29.2
Cold Day 20151212        -26.0
Cold Day 20151213        -22.3
Cold Day 20151214        -23.2
Cold Day 20151215        -20.8
Cold Day 20151216        -19.4
Cold Day 20151217        -22.9
Cold Day 20151218        -25.7
Cold Day 20151219        -22.3
Cold Day 20151220        -22.5
Cold Day 20151221        -27.5
Cold Day 20151222        -30.0
Cold Day 20151223        -30.3
Cold Day 20151224        -30.8
Cold Day 20151225        -37.5
Cold Day 20151226        -29.6
Cold Day 20151227        -26.0
Cold Day 20151228        -27.1
Cold Day 20151229        -33.0
Cold Day 20151230        -25.7
```

Result:

Thus, the program for weather dataset using Map Reduce has been executed successfully.