# EX.NO.10

**IMPLEMENT A MAPREDUCE PROGRAM TO PROCESS A WEATHER DATASET**

# AIM:

To implement a MapReduce python program to process a weather dataset in Hadoop.

# PROCEDURE:

1. Open command prompt as administrator and start the Hadoop by using the command:

# start-all.cmd

1. Create a new directory in the Hadoop file systems using the command:

# hadoop fs -mkdir /weather

1. Upload the input text file into the weather directory using the command:

# hadoop fs -put

**C:/Users/mercy/OneDrive/Documents/DataAnalytics/WeatherPrediction/sa mple\_weather.txt /weather**

1. Create the mapper and reducer files.
2. To execute the files with Hadoop streaming run the following command:

# hadoop jar C:/hadoop-3.3.6/share/hadoop/tools/lib/hadoop-streaming- 3.3.6.jar ^ -file C:/Users/mercy/Documents/DataAnalytics/WeatherPrediction/mapper.py

**^ -file C:/Users/mercy/Documents/DataAnalytics/WeatherPrediciton/reducer.py**

# ^ -input /weather/sample\_weather.txt ^ -output /weather/output ^ -mapper "python mapper.py" ^ -reducer "python reducer.py"

**MAPPER.PY:**

#!C:/ProgramData/chocolatey/bin/python3.exe import sys

def map1():

for line in sys.stdin:

tokens = line.strip().split() if len(tokens) < 13:

continue

station = tokens[0] if "STN" in station:

continue

date\_hour = tokens[2] temp = tokens[3]

dew = tokens[4] wind = tokens[12]

if temp == "9999.9" or dew == "9999.9" or wind == "999.9": continue

hour = int(date\_hour.split("\_")[-1])

date = date\_hour[:date\_hour.rfind("\_")-2] if 4 < hour <= 10:

section = "section1" elif 10 < hour <= 16:

section = "section2" elif 16 < hour <= 22:

section = "section3" else:

section = "section4"

key\_out = f"{station}\_{date}\_{section}" value\_out = f"{temp} {dew} {wind}" print(f"{key\_out}\t{value\_out}")

if name == " main ":

map1()

# REDUCER.PY:

#!C:/ProgramData/chocolatey/bin/python3.exe import sys

def reduce1():

current\_key = None

sum\_temp, sum\_dew, sum\_wind = 0, 0, 0

count = 0

for line in sys.stdin:

key, value = line.strip().split("\t")

temp, dew, wind = map(float, value.split()) if current\_key is None:

current\_key = key

if key == current\_key:

sum\_temp += temp sum\_dew += dew sum\_wind += wind count += 1

else:

avg\_temp = sum\_temp / count avg\_dew = sum\_dew / count avg\_wind = sum\_wind / count

print(f"{current\_key}\t{avg\_temp} {avg\_dew} {avg\_wind}") current\_key = key

sum\_temp, sum\_dew, sum\_wind = temp, dew, wind count = 1

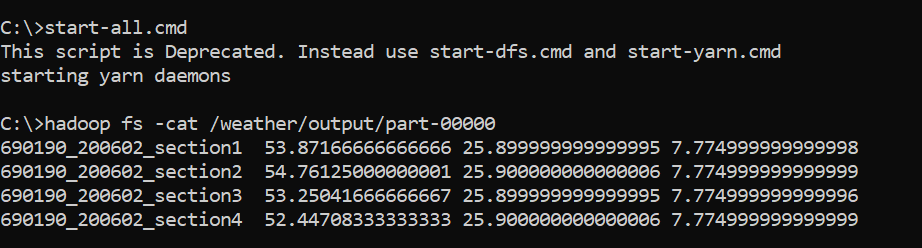
if current\_key is not None:

avg\_temp = sum\_temp / count avg\_dew = sum\_dew / count avg\_wind = sum\_wind / count

print(f"{current\_key}\t{avg\_temp} {avg\_dew} {avg\_wind}") if name == " main ":

reduce1()

# OUTPUT:



**RESULT:**

Thus the implementation of the MapReduce python program to process a weather dataset in Hadoop is executed successfully.