|  |  |
| --- | --- |
| **Ex. No:** | **PySpark** |
|  |

**Aim: To understand map values of RDD in spark**

**Program:**

Q1

from pyspark import SparkContext

sc = SparkContext("local", "Friends Average")

data = sc.textFile("friends\_test.csv").map(lambda line: line.split(","))

age\_friends\_rdd = data.map(lambda x: (int(x[2]), int(x[3])))

combined = age\_friends\_rdd.combineByKey(

    lambda friends: (friends, 1),

    lambda acc, friends: (acc[0] + friends, acc[1] + 1),

    lambda acc1, acc2: (acc1[0] + acc2[0], acc1[1] + acc2[1])

)

average\_friends\_rdd = combined.mapValues(lambda x: x[0] / x[1])

results = average\_friends\_rdd.collect()

for age, avg\_friends in results:

    print(f"Age: {age}, Average Number of Friends: {avg\_friends}")

sc.stop()

Q2

from pyspark import SparkContext

sc = SparkContext("local", "Temperature Analysis")

data = sc.textFile("temp.csv").zipWithIndex().filter(lambda x: x[1] != 0).map(lambda x: x[0].split(","))

temp\_rdd = data.map(lambda x: (x[0], x[1], x[2], int(x[3])))

tmin\_rdd = temp\_rdd.filter(lambda x: x[2] == "TMIN")

overall\_min\_temp = tmin\_rdd.map(lambda x: x[3]).min()

print(f"Overall Minimum Temperature: {overall\_min\_temp}")

min\_temp\_per\_itemID = tmin\_rdd.map(lambda x: (x[0], x[3])).reduceByKey(lambda a, b: min(a, b))

print("Minimum Temperature for each ItemID:")

for itemID, min\_temp in min\_temp\_per\_itemID.collect():

    print(f"ItemID: {itemID}, Minimum Temperature: {min\_temp}")

min\_temp\_per\_stationID = tmin\_rdd.map(lambda x: (x[1], x[3])).reduceByKey(lambda a, b: min(a, b))

print("Minimum Temperature for each StationID:")

for stationID, min\_temp in min\_temp\_per\_stationID.collect():

    print(f"StationID: {stationID}, Minimum Temperature: {min\_temp}")

sc.stop()

Q3

from pyspark import SparkContext

sc = SparkContext("local", "Temperature Analysis")

data = sc.textFile("temp.csv").zipWithIndex().filter(lambda x: x[1] != 0).map(lambda x: x[0].split(","))

temp\_rdd = data.map(lambda x: (x[0], x[1], x[2], int(x[3])))

tmax\_rdd = temp\_rdd.filter(lambda x: x[2] == "TMAX")

overall\_max\_temp = tmax\_rdd.map(lambda x: x[3]).max()

print(f"Overall Maximum Temperature: {overall\_max\_temp}")

max\_temp\_per\_itemID = tmax\_rdd.map(lambda x: (x[0], x[3])).reduceByKey(lambda a, b: max(a, b))

print("Maximum Temperature for each ItemID:")

for itemID, max\_temp in max\_temp\_per\_itemID.collect():

    print(f"ItemID: {itemID}, Maximum Temperature: {max\_temp}")

max\_temp\_per\_stationID = tmax\_rdd.map(lambda x: (x[1], x[3])).reduceByKey(lambda a, b: max(a, b))

print("Maximum Temperature for each StationID:")

for stationID, max\_temp in max\_temp\_per\_stationID.collect():

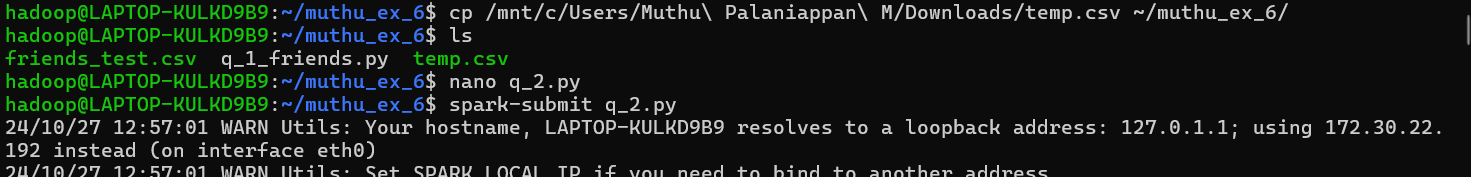
    print(f"StationID: {stationID}, Maximum Temperature: {max\_temp}")

sc.stop()

**Output:**

A screenshot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA computer screen with white text

Description automatically generatedA screen shot of a computer

Description automatically generated

**Result: The required output is achieved using pyspark**