**Task 4**

**Objective**: Broadcast variables and counters

**Broadcast variables:**

Objects which will be cached on each executor in deserialized form.

**How to broadcast:**

bcast\_var = sc.broadcast(5)

**How to access:**

bcast\_var.value

**Note:** We can’t broadcast an RDD or DF.

However, we can collect an RDD as Map using collectAsMap action and broadcast it

result = rdd1.map(lambda x: (x[0],x[1])).collectAsMap()

sc.broadcast(result)

**Use case**: Map side join or lookup functionality

**Accumulators:**

Helps to aggregate across all executors.

**Initialize:**

cntr = sc.accumulator(0)

**Aggregate on executors:**

cntr.add(1)

**Read on driver:**

cntr.value

**Note:** supported data types should be added through associative and commutative operation. Can be incremented by executors but only be read by driver.

Try creating an accumulator for string data.

**Use case:** Implement counters across different stages like in MapReduce program