Questions and Results

1) Describe what are Accumulators and Broadcast Variables in Spark and when to use these shared variables?

   Accumulators

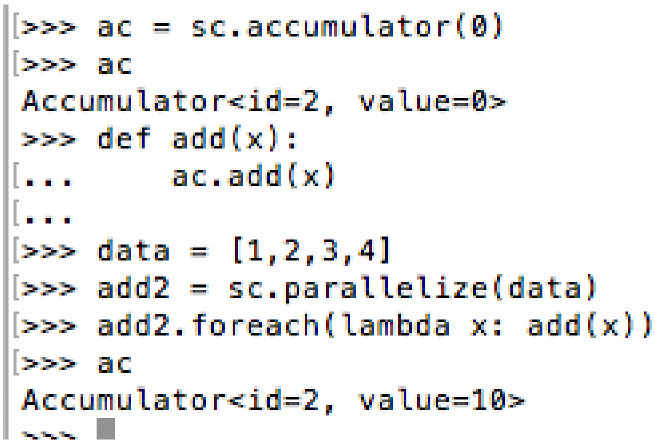
   Accumulators are variables that are only “added” to through an associative operation and can therefore, be efficiently supported in parallel. They can be used to implement counters (as in MapReduce) or sums. Spark natively supports accumulators of numeric types, and programmers can add support for new types.

   Broadcast Variables

   This is useful to reduce the overhead of sending the same variable or data set to all tasks. This is a read only variable which ensures same value is read across all tasks.

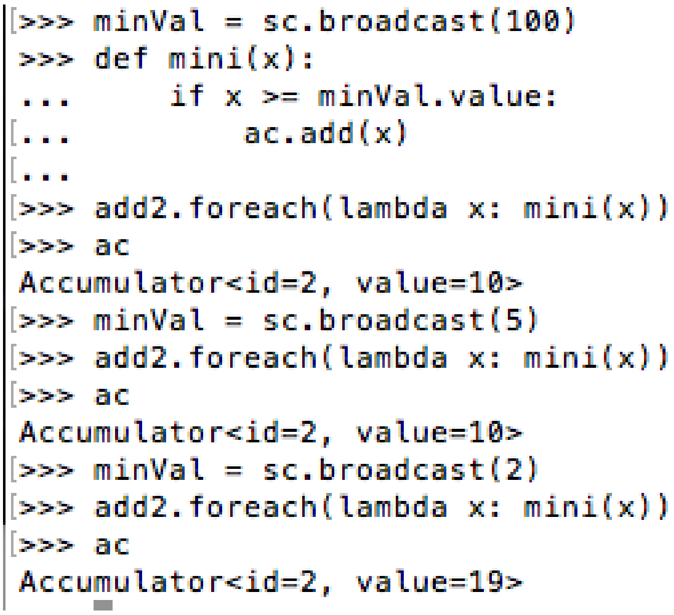
2) Write a Python/Java/Scala code demonstrating the use of Accumulators and Shared Variables in Spark. The code doesn't have to be complex, you can choose simple examples for your submission.

Accumulators



```
[>>> ac = sc.accumulator(0)
[>>> ac
Accumulator<id=2, value=0>
>>> def add(x):
[...     ac.add(x)
[...
[>>> data = [1,2,3,4]
[>>> add2 = sc.parallelize(data)
[>>> add2.foreach(lambda x: add(x))
[>>> ac
Accumulator<id=2, value=10>
>>> 
```

Broadcast Variables



```
>>> minVal = sc.broadcast(100)
>>> def mini(x):
...     if x >= minVal.value:
...         ac.add(x)
...
>>> add2.foreach(lambda x: mini(x))
>>> ac
Accumulator<id=2, value=10>
>>> minVal = sc.broadcast(5)
>>> add2.foreach(lambda x: mini(x))
>>> ac
Accumulator<id=2, value=10>
>>> minVal = sc.broadcast(2)
>>> add2.foreach(lambda x: mini(x))
>>> ac
Accumulator<id=2, value=19>
```