

Q1. Count the no of black cards in deckofcards.txt

Step 1-

C:\Spark\spark-3.1.2-bin-hadoop3.2\bin\spark-shell

```
Administrator: Command Prompt - C:\Spark\spark-3.1.2-bin-hadoop3.2\bin\spark-shell
C:\Windows\System32>C:\Spark\spark-3.1.2-bin-hadoop3.2\bin\spark-shell
21/06/17 02:22:29 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://Wikhil:4040
Spark context available as 'sc' (master = local[*], app id = local-1623876762119).
Spark session available as 'spark'.
Welcome to

  ____      _
 / ___|    / \
| |  | |  / _ \
| |  | | / ___ \
| |  | |/_/   \_\
| |  | |
| |  | |
|_|  |_|      version 3.1.2

Using Scala version 2.12.10 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_101)
Type in expressions to have them evaluated.
Type :help for more information.
```

Step 2- Load deckofcards data in cards_data variable

```
val cards_data= sc.textFile("c:/Users/Nikhil Gaikwad/Downloads/deckofcards.txt")
```

```
Administrator: Command Prompt - C:\Spark\spark-3.1.2-bin-hadoop3.2\bin\spark-shell
```

```
scala> val cards_data= sc.textFile("c:/Users/Nikhil Gaikwad/Downloads/deckofcards.txt")
cards_data: org.apache.spark.rdd.RDD[String] = c:/Users/Nikhil Gaikwad/Downloads/deckofcards.txt MapPartitionsRDD[1] at textFile at <console>:24

scala> cards_data.collect()
res0: Array[String] = Array(BLACK|SPADE|2, BLACK|SPADE|3, BLACK|SPADE|4, BLACK|SPADE|5, BLACK|SPADE|6, BLACK|SPADE|7, BLACK|SPADE|8, BLACK|SPADE|9, BLACK|SPADE|10, BLACK|SPADE|11, BLACK|SPADE|12, BLACK|SPADE|13, BLACK|SPADE|14, BLACK|SPADE|15, BLACK|SPADE|16, BLACK|CLUB|1, BLACK|CLUB|2, BLACK|CLUB|3, BLACK|CLUB|4, BLACK|CLUB|5, BLACK|CLUB|6, BLACK|CLUB|7, BLACK|CLUB|8, BLACK|CLUB|9, BLACK|CLUB|10, BLACK|CLUB|11, BLACK|CLUB|12, BLACK|CLUB|13, BLACK|CLUB|14, BLACK|CLUB|15, BLACK|CLUB|16, RED|DIAMOND|1, RED|DIAMOND|2, RED|DIAMOND|3, RED|DIAMOND|4, RED|DIAMOND|5, RED|DIAMOND|6, RED|DIAMOND|7, RED|DIAMOND|8, RED|DIAMOND|9, RED|DIAMOND|10, RED|DIAMOND|11, RED|DIAMOND|12, RED|DIAMOND|13, RED|DIAMOND|14, RED|DIAMOND|15, RED|DIAMOND|16, RED|HEART|1, RED|HEART|2, RED|HEART|3, RED|HEART|4, RED|HEART|5, RED|HEART|6, RED|HEART|7, RED|HEART|8, RED|HEART|9, RED|HEART|10, RED|HEART|11, RED|HEART|12, RED|HEART|13, RED|HEART|14, RED|HEART|15, RED|HEART|16)
```

```
scala>
```

using collect we can see the data

```
cards_data.collect()
```

Step 3- Count number of rows

```
cards data.count()
```

Step 4- Filter out black cards only and count those cards

```
val black cards=cards data.filter(line => line.contains("BLACK"))
```

```
black_cards.count()
```

```
Administrator: Command Prompt - C:\Spark\spark-3.1.2-bin-hadoop3.2\bin\spark-shell
scala> val cards_data= sc.textFile("c:/Users/Nikhil Gaikwad/Downloads/deckofcards.txt")
cards_data: org.apache.spark.rdd.RDD[String] = c:/Users/Nikhil Gaikwad/Downloads/deckofcards.txt MapPartitionsRDD[1] at textFile at <console>:24

scala> cards_data.collect()
res0: Array[String] = Array(BLACK|SPADE|2, BLACK|SPADE|3, BLACK|SPADE|4, BLACK|SPADE|5, BLACK|SPADE|6, BLACK|SPADE|7, BLACK|SPADE|8, BLACK|SPADE|9, BLACK|SPADE|10, BLACK|SPADE|J, BLACK|SPADE|Q, BLACK|SPADE|K, BLACK|CLUB|2, BLACK|CLUB|3, BLACK|CLUB|4, BLACK|CLUB|5, BLACK|CLUB|6, BLACK|CLUB|7, BLACK|CLUB|8, BLACK|CLUB|9, BLACK|CLUB|10, BLACK|CLUB|J, BLACK|CLUB|Q, BLACK|CLUB|K, BLACK|CLUB|A, RED|DIAMOND|2, RED|DIAMOND|3, RED|DIAMOND|4, RED|DIAMOND|5, RED|DIAMOND|6, RED|DIAMOND|7, RED|DIAMOND|8, RED|DIAMOND|9, RED|DIAMOND|10, RED|DIAMOND|J, RED|DIAMOND|Q, RED|DIAMOND|K, RED|DIAMOND|A, RED|HEART|2, RED|HEART|3, RED|HEART|4, RED|HEART|5, RED|HEART|6, RED|HEART|7, RED|HEART|8, RED|HEART|9, RED|HEART|10, RED|HEART|J, RED|HEART|Q, RED|HEART|K, RED|HEART|A)

scala> cards_data.count()
res1: Long = 52

scala> val black_cards=cards_data.filter(line => line.contains("BLACK"))
black_cards: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[2] at filter at <console>:25

scala> black_cards.count()
res2: Long = 26

scala>
```

Q2. Count the no of black and no queen cards in deckofcards.txt

Step 1:Filter out based on condition

```
val black_queen =cards_data.filter(line => line.contains("BLACK") & line.contains("Q"))
```

Step 2:Check data and count the records

```
black_queen.collect()
```

```
black_queen.count()
```

```
scala> val black_queen =cards_data.filter(line => line.contains("BLACK") & line.contains("Q"))
black_queen: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[3] at filter at <console>:25

scala> black_queen.collect()
res3: Array[String] = Array(BLACK|SPADE|Q, BLACK|CLUB|Q)

scala> black_queen.count()
res4: Long = 2

scala>
```