Problem Statement

Create a dataframe with 1 to 100 and save as parquet file.

Solution:

Importing the scala and spark packages

```
scala> import scala.collection.mutable.ListBuffer
import scala.collection.mutable.ListBuffer

scala> import org.apache.spark.sql._
import org.apache.spark.sql._

scala> import sqlContext.implicits._
import sqlContext.implicits._

scala>
```

Defining a dataList 1 to 100

```
scala> def dataList(n:ListBuffer[Integer])=
     | {
     | var i =1
     | while(i<=100)
     | {
     | n+=i
     | i+=1
     | }
     | }
dataList: (n: scala.collection.mutable.ListBuffer[Integer])Unit

scala>
```

Defining a ListBuffer object of integers,Invoke UDF dataList and pass ListBuffer object as an argument and converting ListBuffer to List and creating RDD with list of integers

```
scala> var myDataList = new ListBuffer[Integer]()
myDataList: scala.collection.mutable.ListBuffer[Integer] = ListBuffer()

scala>

scala> dataList(myDataList)

scala>

scala> val myDataList_show = myDataList.toList
myDataList_show: List[Integer] = List(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24,
25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 5
6, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87
, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100)

scala> val pdataRDD = sc.parallelize(myDataList_show)
pdataRDD: org.apache.spark.rdd.RDD[Integer] = ParallelCollectionRDD[231] at parallelize at <console>:70

scala>
```

Creating Dataframe with the help of RDD defined above

```
scala> val pdataDF = sqlContext.createDataFrame(pdataRDD.map(Tuple1.apply)).toDF("Column")
pdataDF: org.apache.spark.sql.DataFrame = [Column: int]
```

Saving the Dataframe as Parquet File

```
scala> pdataDF.saveAsParquetFile("datafile.parquet")
warning: there were 1 deprecation warning(s); re-run with -deprecation for details

scala>
```

To read the data that was saved as a parquet file, use the **read** method with data type: parquet and filename (given before) as parameter: *[filename]*.parquet. readFile

```
scala> val readFile = sqlContext.read.parquet("datafile.parquet")
readFile: org.apache.spark.sql.DataFrame = [Column: int]
```

use the .show() method with readFile to display the contents of the file.

```
scala> readFile.show()
+------+
|Column|
+------+
|     1|
|     2|
|     3|
|     4|
|     5|
|     6|
|     7|
|     8|
|     9|
|    10|
|    11|
|    12|
|    13|
|    14|
|    15|
|    16|
|    17|
|    18|
|    19|
|    20|
+------+
only showing top 20 rows

scala>
```

Use .show(100, false) to view the entire data in the file.

```
scala> readFile.show(100,false)
+------+
|Column|
+------+
|1     |
|2     |
|3     |
|4     |
|5     |
|6     |
|7     |
|8     |
|9     |
|10    |
|11    |
|12    |
|13    |
|14    |
|15    |
|16    |
|17    |
|18    |
|19    |
|20    |
|21    |
|22    |
|23    |
|24    |
|25    |
|26    |
|27    |
|28    |
|29    |
|30    |
|31    |
|32    |
|33    |
|34    |
|35    |
|36    |
|37    |
|38    |
|39    |
|40    |
|41    |
|42    |
|43    |
|44    |
|45    |
|46    |
|47    |
|48    |
|49    |
|50    |
|51    |
|52    |
|53    |
|54    |
|55    |
|56    |
|57    |
|58    |
|59    |
|60    |
|61    |
|62    |
|63    |
|64    |
|65    |
|66    |
|67    |
|68    |
|69    |
|68    |
|69    |
|70    |
|71    |
|72    |
|73    |
|74    |
|75    |
|76    |
|77    |
|78    |
|79    |
|80    |
|81    |
|82    |
|83    |
|84    |
|85    |
|86    |
|87    |
|88    |
|89    |
|90    |
|91    |
|92    |
|93    |
|94    |
|95    |
|96    |
|97    |
|98    |
|99    |
|100   |
+------+

scala>
```

Showing dataFile.parquet file and its contents