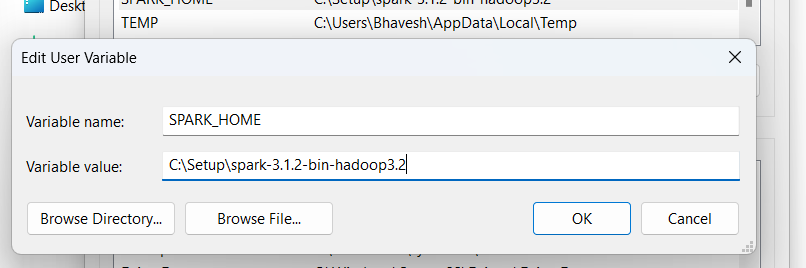
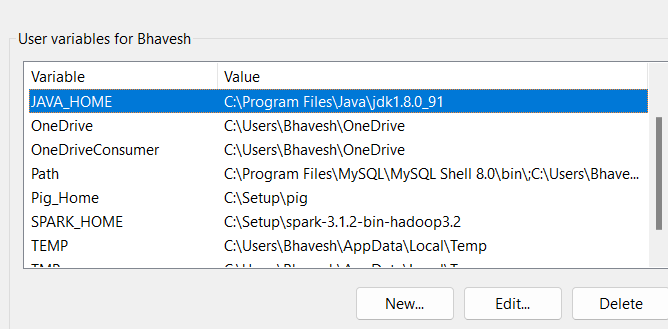
**Practical No 6**

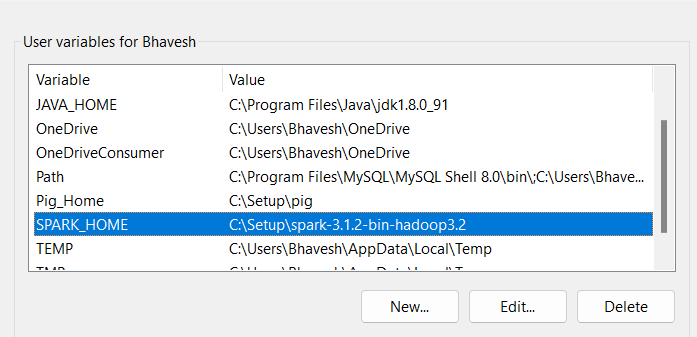
* **Set SPARK\_HOME in environment variable**



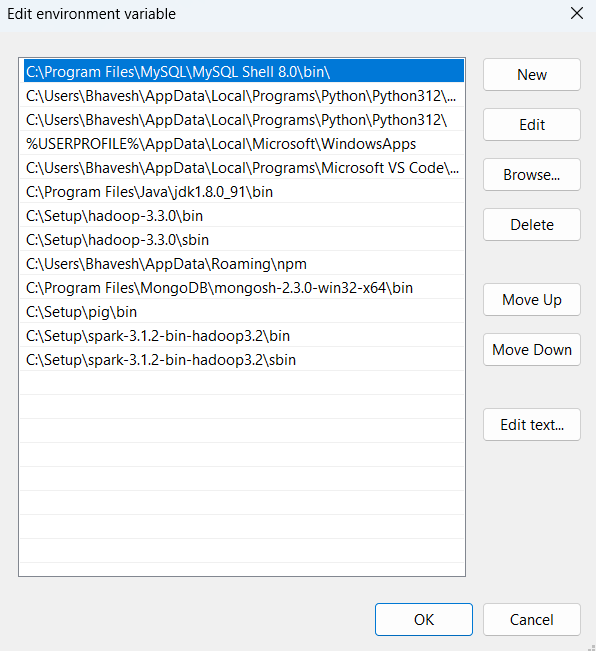
* **Set JAVA\_HOME in environment variable**



* **Set path of Spark 3.x.x/bin and path of /Spark 3.x.x/sbin in path**

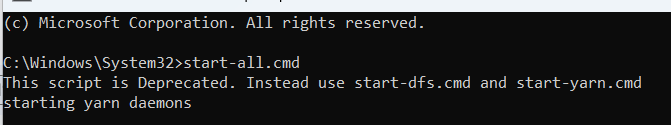


* **Set path of Java/bin in environment variable**

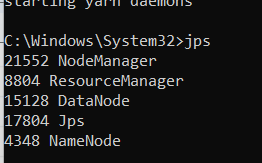


* **Open Namenode, DataNode Resource Manager and Data Manager using following**

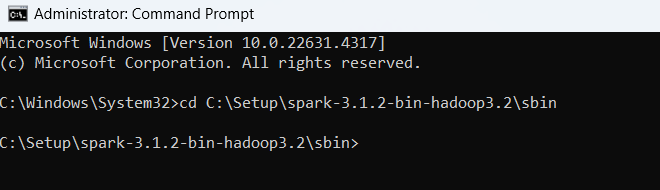
**command:- start-dfs.cmd**



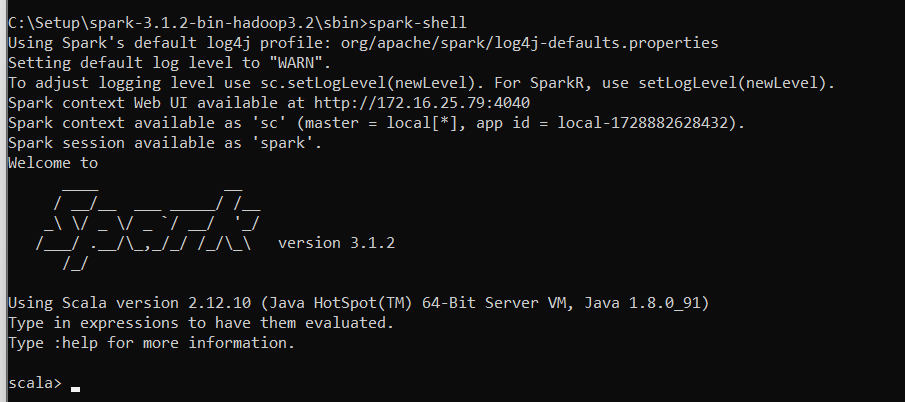
* **Check whether these 4 process are running or not**



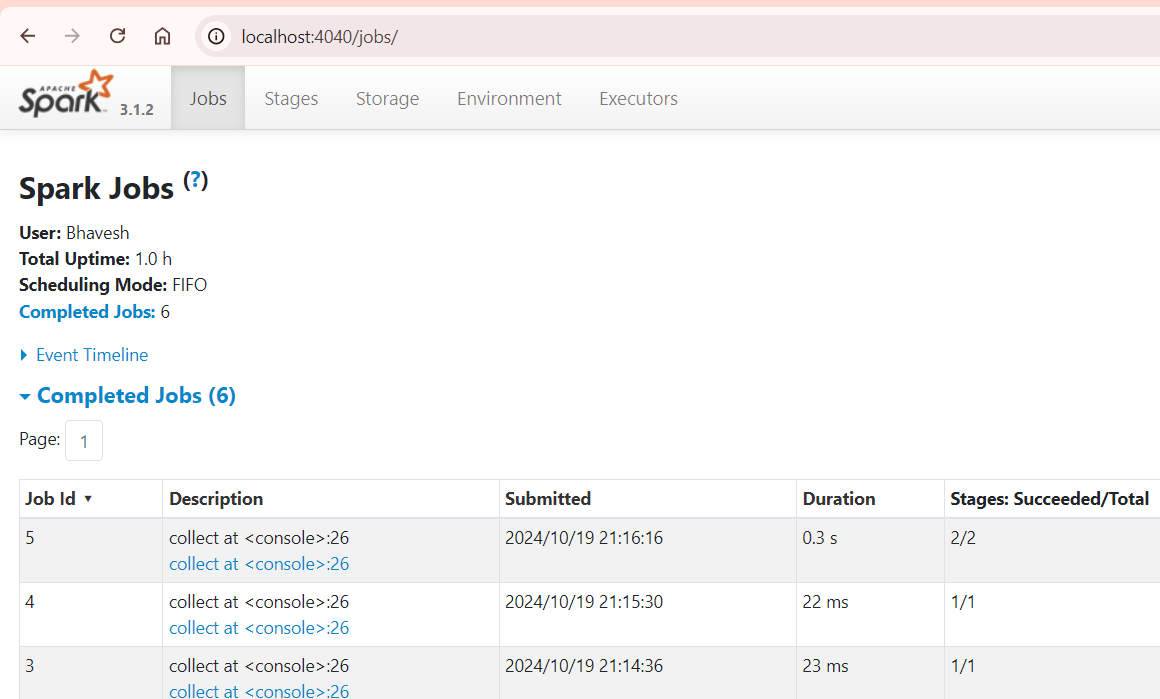
* **Change the directory filepath/Spark 3.x.x/sbin on cmd**



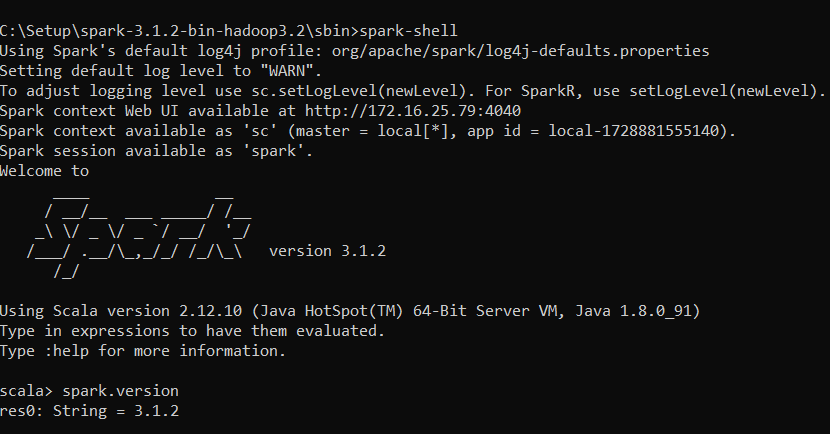
* **Run command: spark-shell**



* **Check spark jobs using following url in browser**

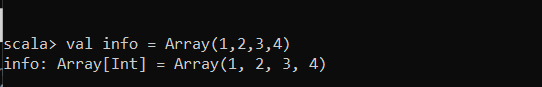
****

* **check the version of spark RDD (Resilient Distributed Datasets ) using following command:- scala> spark.version**



* **Create variable info to stored array list data**

**>val info = Array(1, 2, 3, 4)**



* **Create distinfo variable**

**val distinfo = sc.parallelize(info)**

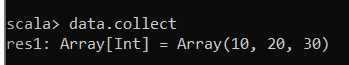


1. **Create an RDD using a parallelized collection.**

* **Create variable data**



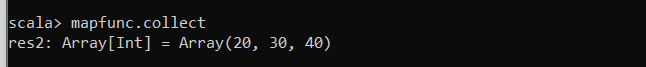
* **Display the collection data on the command prompt using command-data.collect**



* **Apply the map function and pass the expression required to perform.**
* **Create variable mapfunc**

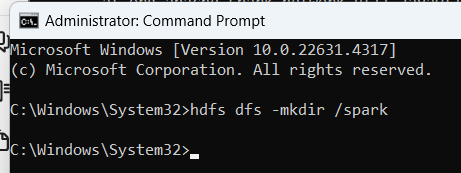


* **Display the mapfunc on the command prompt using command mapfunc.collect**.

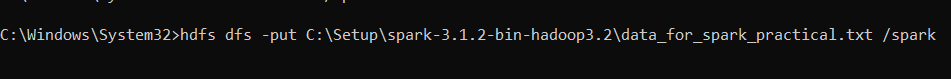
.

* **Create an RDD using a text file.**
* **Word count program using map reduce in Spark**

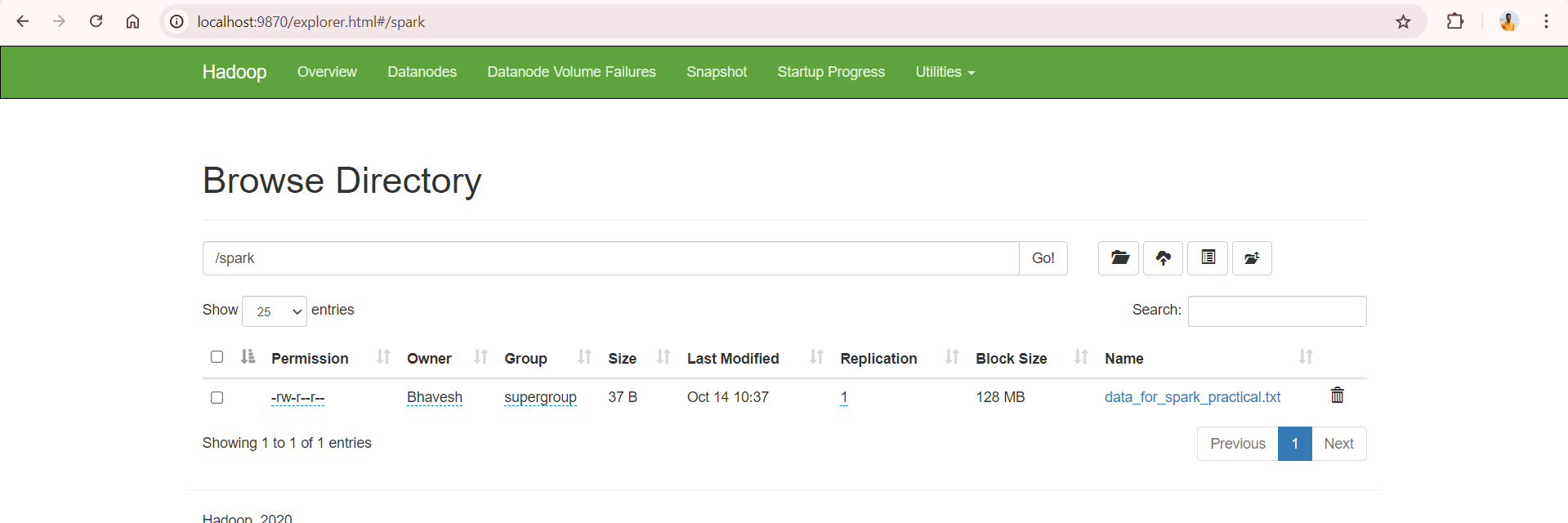
**Create a directory in HDFS, where to keep text file.**



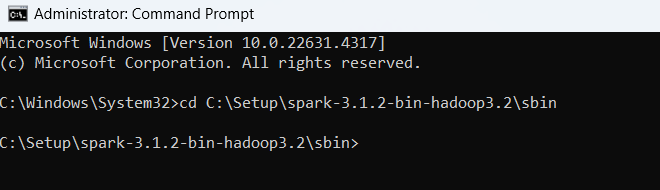
* **Upload the data\_for\_spark\_practical.txt file on HDFS in the specific directory**.



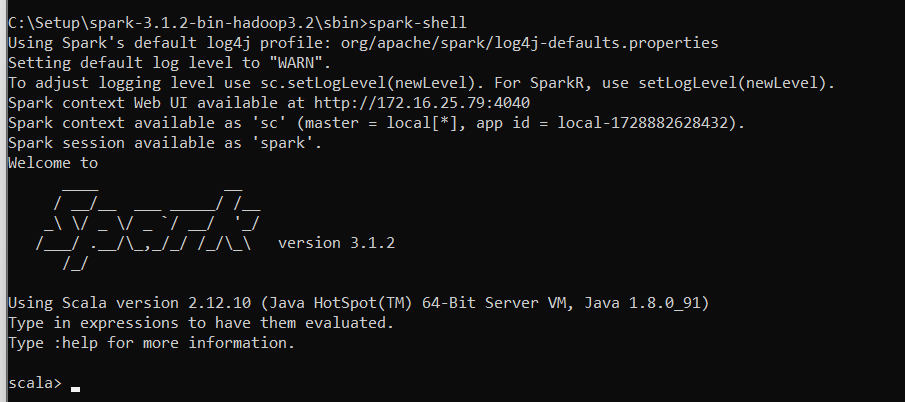
* **Check http://localhost:9870/ and change spark directory**



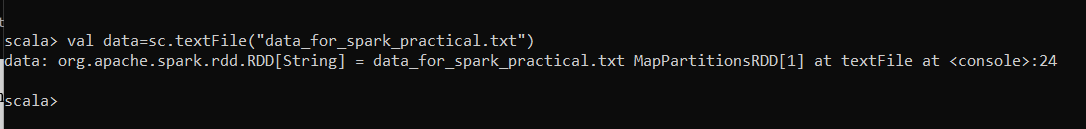
* **Change directory to Spark**



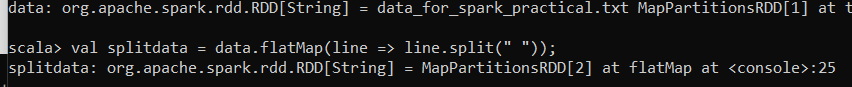
* **Start spark-shell**



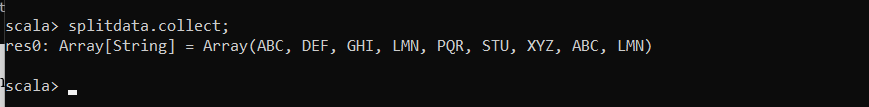
* **create an RDD by using the following command.**
* **val data=sc.textFile("data\_for\_spark\_practical.txt")**



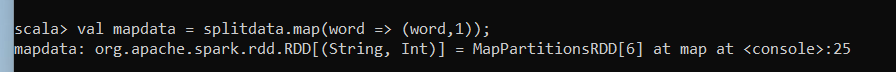
* **Splitting data:**
* **Create variable splitdata**



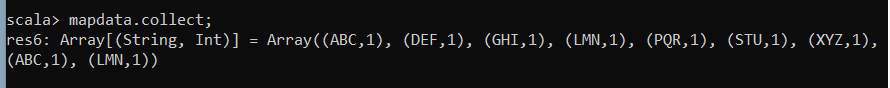
* **Display the splitdata on the command prompt using commandsplitdata.collect**



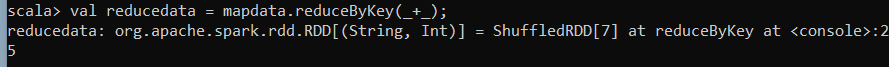
* **Map Key or value**
* **Create variable mapdata**



* **Display the mapdata on the command prompt using command mapdata.collect**



* **Reduce the data**
* **Create variable reduce data**



* **Display the reduce data on the command prompt using command reducedata.collect**

