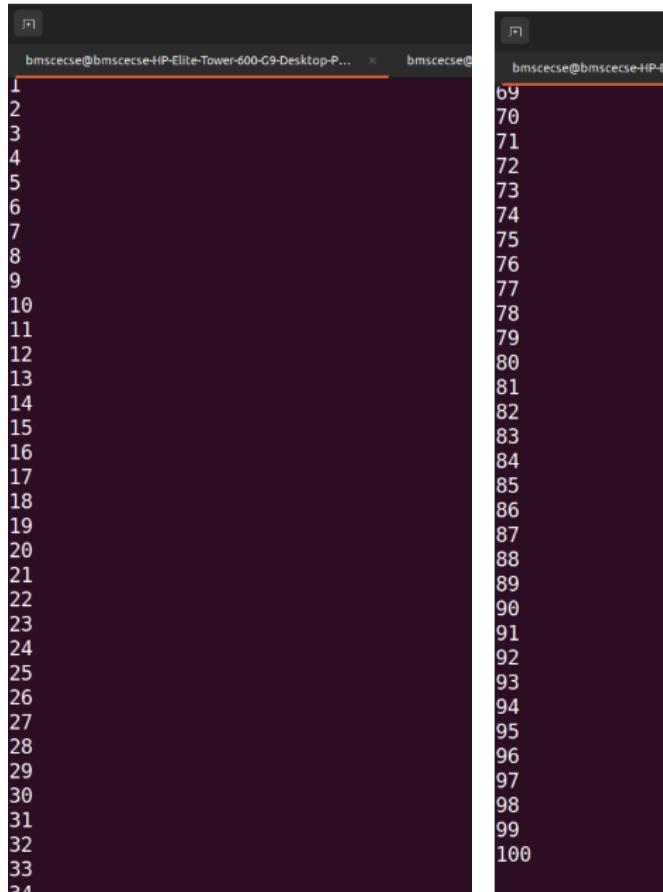


Lab 9: Hadoop, Spark

Code with Output:

Write a Scala program to print numbers from 1 to 100 using for loop

```
object pi {  
  def main(args: Array[String]): Unit = {  
    for(counter <- 1 to 100)  
      print(counter + " ")  
    println()  
  }  
}
```



The image displays two terminal windows side-by-side, showing the output of a Scala program. The left terminal window shows the first 34 lines of output, and the right terminal window shows the remaining lines from 69 to 100. The output consists of numbers from 1 to 100, printed in a single line with spaces between them.

Left Terminal Output (Lines 1-34):

```
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
```

Right Terminal Output (Lines 69-100):

```
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
```

Using RDD and FlatMap

count how many times each word appears in a file and write out a list of words whose count is strictly greater than 4 using Spark

```
bmscscse@bmscscse-HP-Elite-Tower-600-G9-Desktop-PC: ~  
res26: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[28] at map at  
<console>:26  
  
scala> .reduceByKey(_ + _)  
res27: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[29] at reduceByKey  
at <console>:26  
  
scala> .filter(_._2 > 4)  
res28: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[30] at filter  
at <console>:26  
  
scala> result.collect().foreach(println)  
Spark is fast. Spark is powerful. Spark is amazing. Spark is fast. Spark rocks.  
Fast spark processing.  
  
scala> res22: org.apache.spark.rdd.RDD[(String, Int)]  
res30: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[23] at filter  
at <console>:26  
  
scala> res22.collect().foreach(println)  
(spark,6)  
  
scala> █
```

**For a given Text file, Create a
Map Reduce program to sort the content in an alphabetic order listing only
top 10 maximum occurrences of words.(Hadoop Program)**

```
hadoop@bmscscse-HP-Elite-Tower-600-G9-Desktop-PC: ~/De...  
hadoop@bmscscse-HP-Elite-Tower-600-G9-Desktop-PC: ~/De...  
CONNECTION=0  
IO_ERROR=0  
WRONG_LENGTH=0  
WRONG_MAP=0  
WRONG_REDUCE=0  
File Input Format Counters  
  Bytes Read=1052  
File Output Format Counters  
  Bytes Written=63  
2025-05-20 15:28:53,269 INFO streaming.StreamJob: Output directory: /user/hadoop  
/output  
hadoop@bmscscse-HP-Elite-Tower-600-G9-Desktop-PC: ~/Desktop$ hadoop fs -cat /user  
/hadoop/output/part-00000  
a 5  
and 7  
apple 4  
data 8  
hadoop 5  
how 5  
is 8  
map 5  
of 5  
word 5  
hadoop@bmscscse-HP-Elite-Tower-600-G9-Desktop-PC: ~/Desktop$ █
```