20/05/25

Lab 9 :→ Scala program to print numbers from 1 to 100 using a for loop

```
scala > for (i <- 1 to 100) {
        println(i)
        }
```

1
2
3
4
:
:

→ 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

```
scala > val textFile = sc.textFile("/home/bmscecse/text.txt")

scala > val words = textFile.flatMap(line => line.split("\\W+"))

scala > val wordPairs = words.map(word => (word.toLowerCase, 1))

scala > val wordCounts = wordPairs.reduceByKey(_ + _)
```

PAGE NO :
DATE :

scala> val frequentWords = wordCounts.filter{case(_,count) => count > 4}

scala> frequentWords.collect().foreach(println)

(Hello, 5)
(Hi, 6)

→ For a given Text file, create a MapReduce program to sort the content in an alphabetic order listing only top 10 maximum occurences of words (Hadoop program)

→ Mapper.py

```
#!/usr/bin/env python3
import sys
import re

for line in sys.stdin:
    line = line.strip().lower()
    words = re.findall(r'\b[a-z]+\b', line)
    for word in words:
        print(f"{word}\t1")
```

PAGE NO :
DATE :

Reducer.py

```
#!/usr/bin/env python3

import sys
from collections import defaultdict

word_count = defaultdict(int)

for line in sys.stdin:
    line = line.strip()
    if not line:
        continue
    word, count = line.split('\t')
    word_count[word] += int(count)

top_words = sorted(word.count.items(),
 key = lambda x: (-x[1], x[0]))[:10].

top_words = sorted(top_words,
        key = lambda x: x[0])

for word, count in top_words:
    print(f"{word}\t{count}")
```

→ hadoop jar /home/bmscse/had.jar \
- input /home/bmscse/inp.txt \
- output /home/bmscse/output_dir \
- mapper mapper.py \
- reducer reducer.py \
- file mapper.py \
- file reducer.py.

2015