

Lab 9:-

1) Write a Scala Program to print Numbers from 1 to 100 using for loop

```
object Print Numbers {
  def main(args: Array[String]): Unit = {
    for (i <- 1 to 100) {
      printLn(i)
    }
  }
}
Print Numbers.main(Array())
```

2) Create MapReduce Program to sort the content in an alphabet

Write out list of words whose count is strictly greater than 1 using spark

```
val data = sc.textFile("spark data.txt")
data.collect();
val split data = data.flatMap(line =>
  line.split(" "));
split data.collect();
val map data = split data.map(word => (word, 1));
map data.collect();
val reduce data = map data.reduceByKey(_+_);
reduce data.collect();
```

Output

spark: 5

is: 6

3) Mapper.py

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

```
import sys
```

```
import re
```

```
for line in sys.stdin:
```

```
    line = line.strip().lower()
```

```
    words = re.findall(r'([a-z]+)', line)
```

```
    for word in words:
```

```
        print("{} {}".format(word, 1))
```

### 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(' ')
```

```
    word_count[word] += int(count)
```

```
top_words = sorted(word_count.items(),
```

```
                    key=lambda x: x[1], reverse=True)
```

```
top_words = sorted(top_words,
```

```
                    key=lambda x: x[1])
```

```
for word, count in top_words:
```

```
    print("{} {}".format(word, count))
```



hadoop jar / home / bin / hbase / hadoop.jar  
input / home / bin / hbase / input.txt  
- output / home / bin / hbase / output\_dir  
- mapper - mapper.py  
- reducer - reducer.py

20/5/25