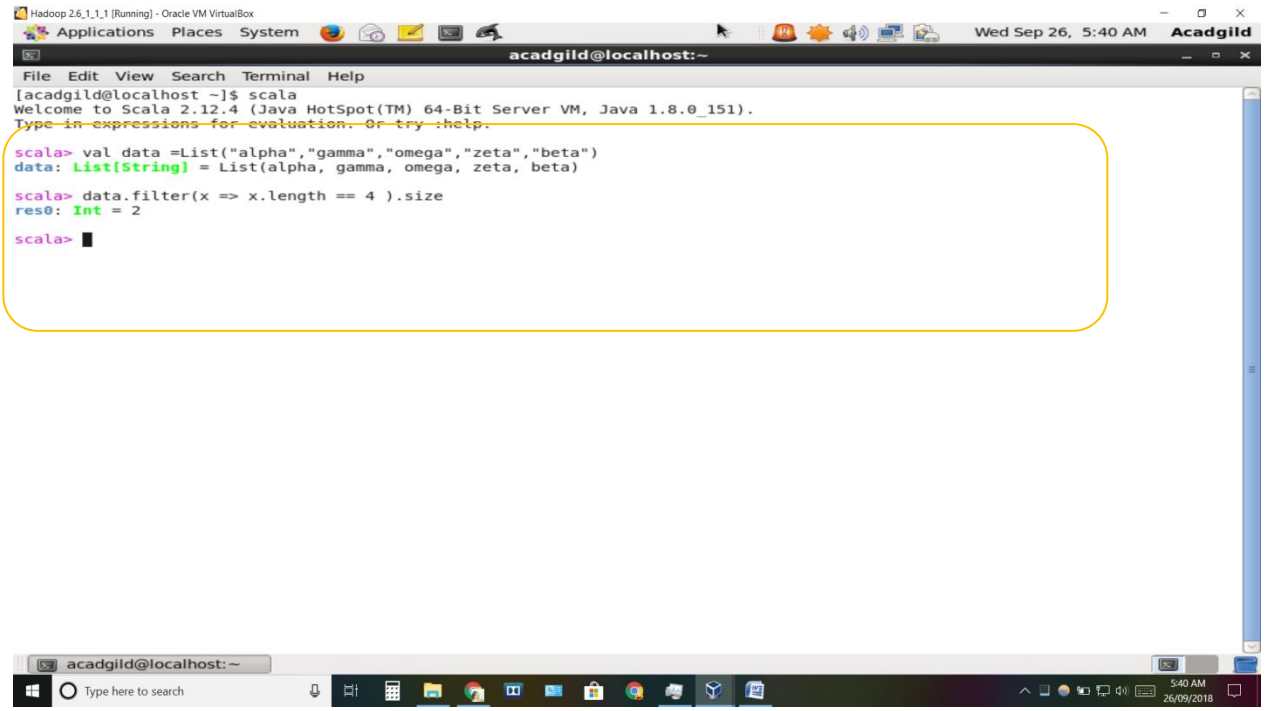
Big Data Hadoop 'Session 14 :*Scala Basics 1 Assignment 1'*

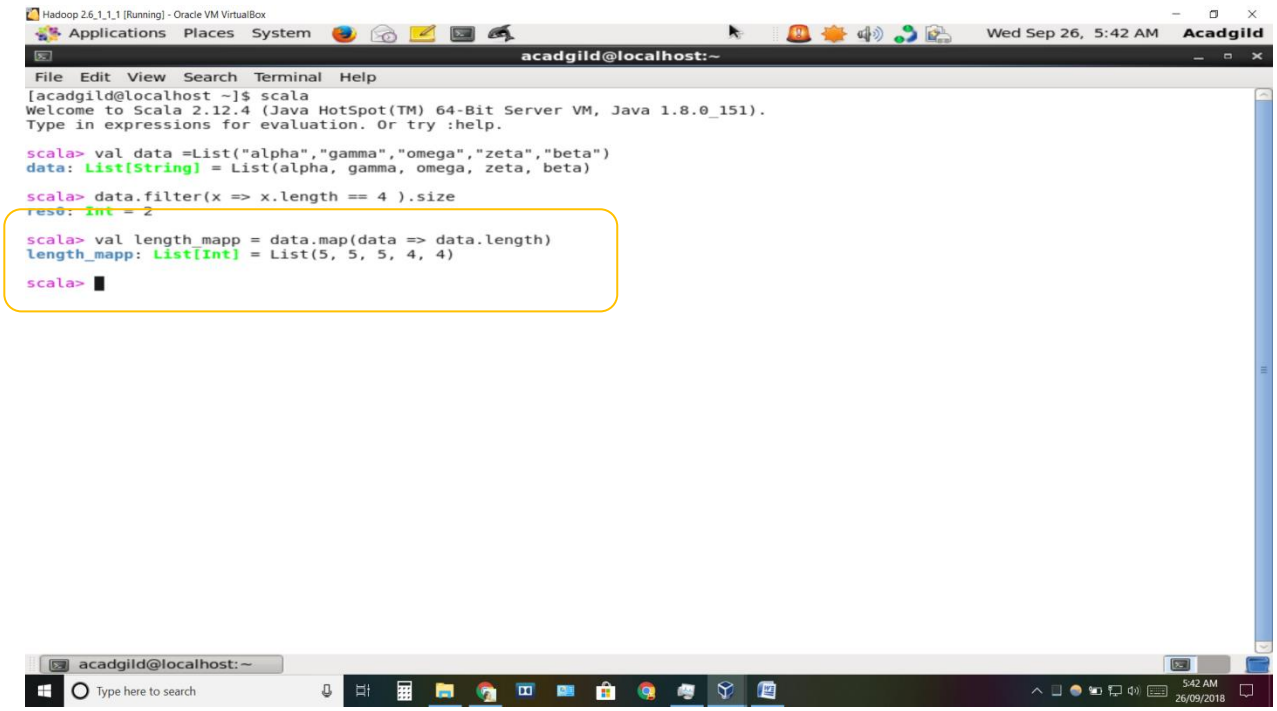
Task 1

Given a list of strings - List[String] ("alpha", "gamma", "omega", "zeta", "beta")

1) Find count of all strings with length 4.

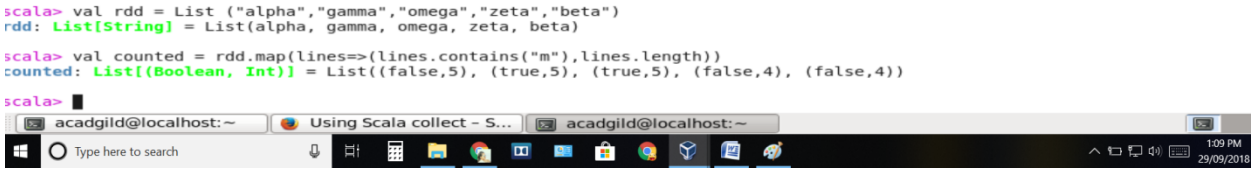


2) Convert the list of string to a list of integers, where each string is mapped to its Corresponding length.



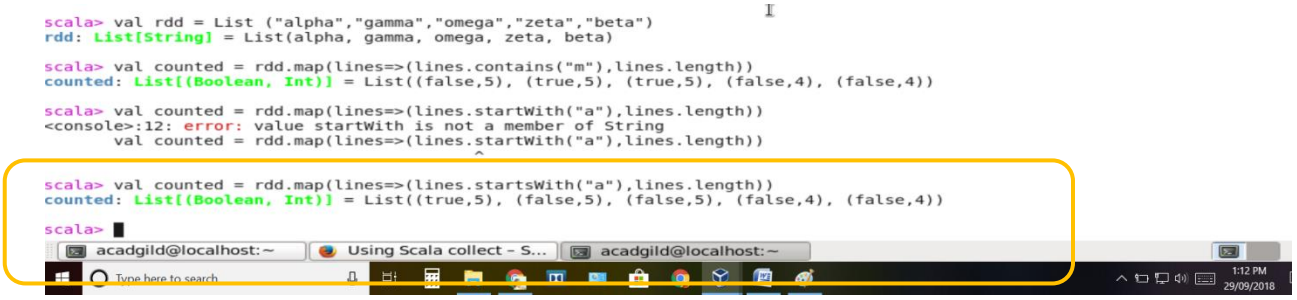
3) Find count of all strings which contain alphabet 'm'.

This shows two value contains letter 'm' i.e 'gamma' and 'omega'



4) Find the count of all strings which start with the alphabet 'a'.

This shows one value startsWith a i.e alpha



Task 2

1) Create a list of tuples, where the 1st element of the tuple is an int and the second element is a string.

Example - ((1, ‘alpha’), (2, ‘beta’), (3, ‘gamma’), (4, ‘zeta’), (5, ‘omega’))

- For the above list, print the numbers where the corresponding string length is 4.

Here 2nd and 4th element length is equal to 4

```
scala> val input : List [(Int,String)] = List ((1,"alpha"),(2,"beta"),(3,"gamma"),(4,"zeta"),(5,"omega"))
input: List[(Int, String)] = List((1,alpha), (2,beta), (3,gamma), (4,zeta), (5,omega))

scala> input.collect{ case (int, string) if string.length == 4 => int}
res0: List[Int] = List(2, 4)

scala> input.filter{ case (int,string) = > string.length == 4 }
<console>:1: error: '=>' expected but '=' found.
       input.filter{ case (int,string) = > string.length == 4 }
                                         ^

scala> input.filter{ case (int,string) => string.length == 4 }.map {case (int ,string) => int}
res1: List[Int] = List(2, 4)

scala>
```

2) find the average of all numbers, where the corresponding string contains alphabet ‘m’ or alphabet ‘z’

```
Applications  Places  System
acadgild@localhost:~
File  Edit  View  Search  Terminal  Help
[acadgild@localhost ~]$ scala
Welcome to Scala 2.12.4 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_151).
Type in expressions for evaluation. Or try :help.

scala> val input : List [(Int,String)] = List ((1,"alpha"),(2,"beta"),(3,"gamma"
),(4,"zeta"),(5,"omega"))
input: List[(Int, String)] = List((1,alpha), (2,beta), (3,gamma), (4,zeta), (5,o
mega))

scala> val rdd1 = input.map(lines=>(lines._1,lines._2.length,lines._2)).filter(v
alues=>(values._3.contains("m")) ||(values._3.contains("z")))
rdd1: List[(Int, Int, String)] = List((3,5,gamma), (4,4,zeta), (5,5,omega))

scala> val rdd2 = rdd1.map(values=>values._1)
rdd2: List[Int] = List(3, 4, 5)

scala> val rdd3 = rdd2.reduce(_+_)/3
rdd3: Int = 4

scala>
```