Assignment 14.1
Scala Basics 1

__Task 1 :__

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

- Find count of all strings with length 4.

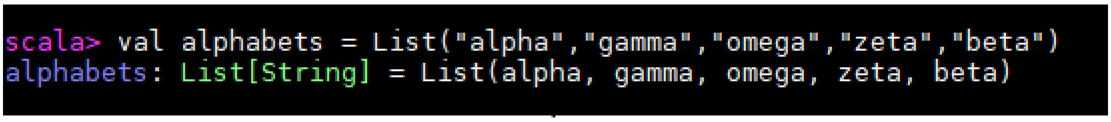
- Convert the list of string to a list of integers, where each string is mapped to its corresponding length.

- Find count of all strings which contain alphabet ‘m’.

- Find the count of all strings which start with the alphabet ‘a’.

We have created an alphabets List as shown below :

val alphabets = List("alpha","gamma","omega","zeta","beta")

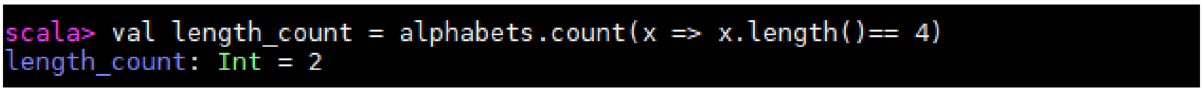


```
scala> val alphabets = List("alpha","gamma","omega","zeta","beta")
alphabets: List[String] = List(alpha, gamma, omega, zeta, beta)
```

1) Find count of all strings with length 4.

In **alpahabets** List, there are two elements with length as 4. They are **“zeta”** and **“beta”**.
So count of these elements in List are **2**.

val length_count = alphabets.count(x => x.length() == 4)



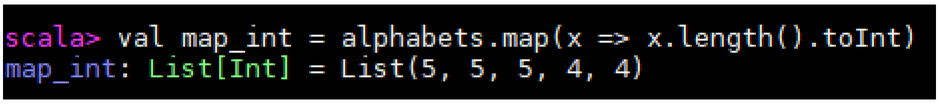
```
scala> val length_count = alphabets.count(x => x.length()== 4)
length_count: Int = 2
```

Here we have used **count** and **length** methods to find out number of elements having length 4.
So it gives result : **2**

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

Here we have used **map** method to map list of string to list of integers and **length** method to find out length of corresponding element and **toInt** method to convert it to integer.
So it gives below result :

val map_int = alphabets.map(x => x.length().toInt)



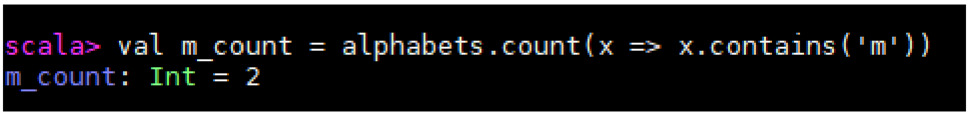
```
scala> val map_int = alphabets.map(x => x.length().toInt)
map_int: List[Int] = List(5, 5, 5, 4, 4)
```

Assignment 14.1
Scala Basics 1

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

There are two elements **“gamma”** and **“omega”** in alphabets list. So count is 2.

val m_count = alphabets.count(x => x.contains(‘m’))

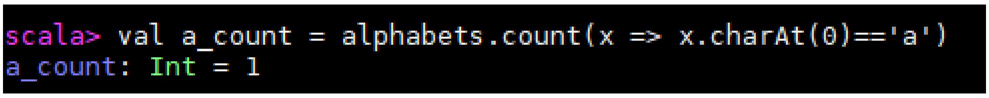


```
scala> val m_count = alphabets.count(x => x.contains('m'))
m_count: Int = 2
```

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

There is only one element **“alpha”** which start with alphabet ‘a’ in alphabets list. So count is **1**.

val a_count = alphabets.count(x => x.charAt(0)==’a’)



```
scala> val a_count = alphabets.count(x => x.charAt(0)=='a')
a_count: Int = 1
```

Assignment 14.1
Scala Basics 1

Task 2 :

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

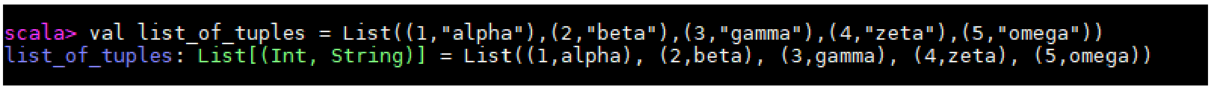
Example - ((1, ‘alpha’), (2, ‘beta’), (3, ‘gamma’), (4, ‘zeta’), (5, ‘omega’))

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

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

We have created a list of tuples as shown below :

val list_of_tuples = List((1,"alpha"),(2,"beta"),(3,"gamma"),(4,"zeta"),(5,"omega"))



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

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

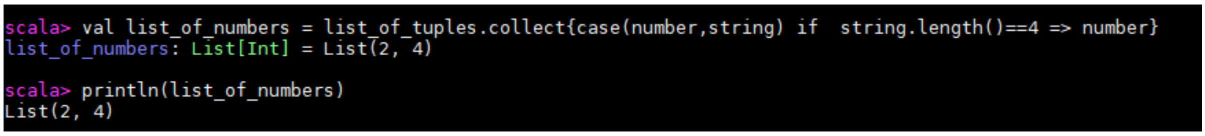
There are two ways to do this :
a) By using collect
b) By using filter and map together

a) By using collect :

We have used **collect** method which is a combination of **filter** and **map** methods and print numbers :

val list_of_numbers = list_of_tuples.collect{case(number,string) if string.length()==4 => number}

It gives output as **List(2,4).**



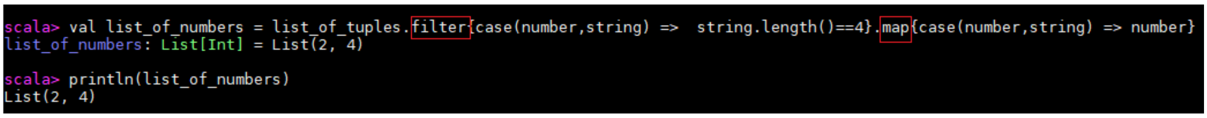
```
scala> val list_of_numbers = list_of_tuples.collect{case(number,string) if  string.length()==4 => number}
list_of_numbers: List[Int] = List(2, 4)

scala> println(list_of_numbers)
List(2, 4)
```

Assignment 14.1
Scala Basics 1

b) By using filter and map together :

Here we have used **filter** method first to filter out values having length of string as 4.
Then we have used **map** method to fetch only number as output

val list_of_numbers = list_of_tuples.filter{case(number,string) => string.length()==4}.map{case(number,string) => number}

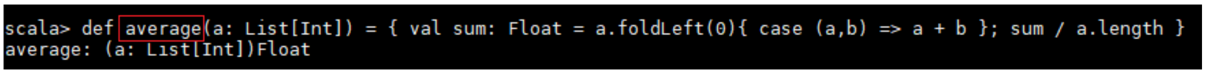


```
scala> val list_of_numbers = list_of_tuples.filter{case(number,string) =>  string.length()==4}.map{case(number,string) => number}
list_of_numbers: List[Int] = List(2, 4)

scala> println(list_of_numbers)
List(2, 4)
```

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

Average method is not a readily available, as it is not a built-in scala function.
Hence we have created an **average** method as shown below by using **foldleft** function which is using 0 as first value and starts from the left side and iterates to the right till the last element in the list:

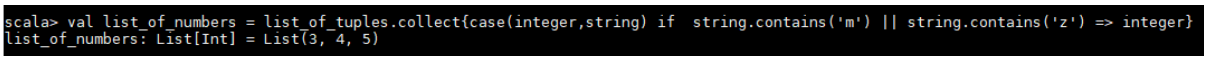
def average(a: List[Int]) = { val sum: Float = a.foldLeft(0){ case (a,b) => a + b }; sum / a.length }



```
scala> def average(a: List[Int]) = { val sum: Float = a.foldLeft(0){ case (a,b) => a + b }; sum / a.length }
average: (a: List[Int])Float
```

Then we have created **list_of_numbers** as List. Then we have List of numbers whose corresponding string contains either character ‘m’ or ‘z’ as shown below.

val list_of_numbers = list_of_tuples.collect{case(integer,string) if string.contains('m') || string.contains('z') => integer}

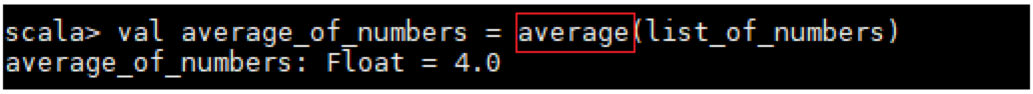


```
scala> val list_of_numbers = list_of_tuples.collect{case(integer,string) if  string.contains('m') || string.contains('z') => integer}
list_of_numbers: List[Int] = List(3, 4, 5)
```

Then we have used this **average** method which we have created to apply it on **list_of_numbers**.

As we are applying average method on List(3,4,5), it is giving average as 4.0

val average_of_numbers = average(list_of_numbers)



```
scala> val average_of_numbers = average(list_of_numbers)
average_of_numbers: Float = 4.0
```