

## TASK 1:

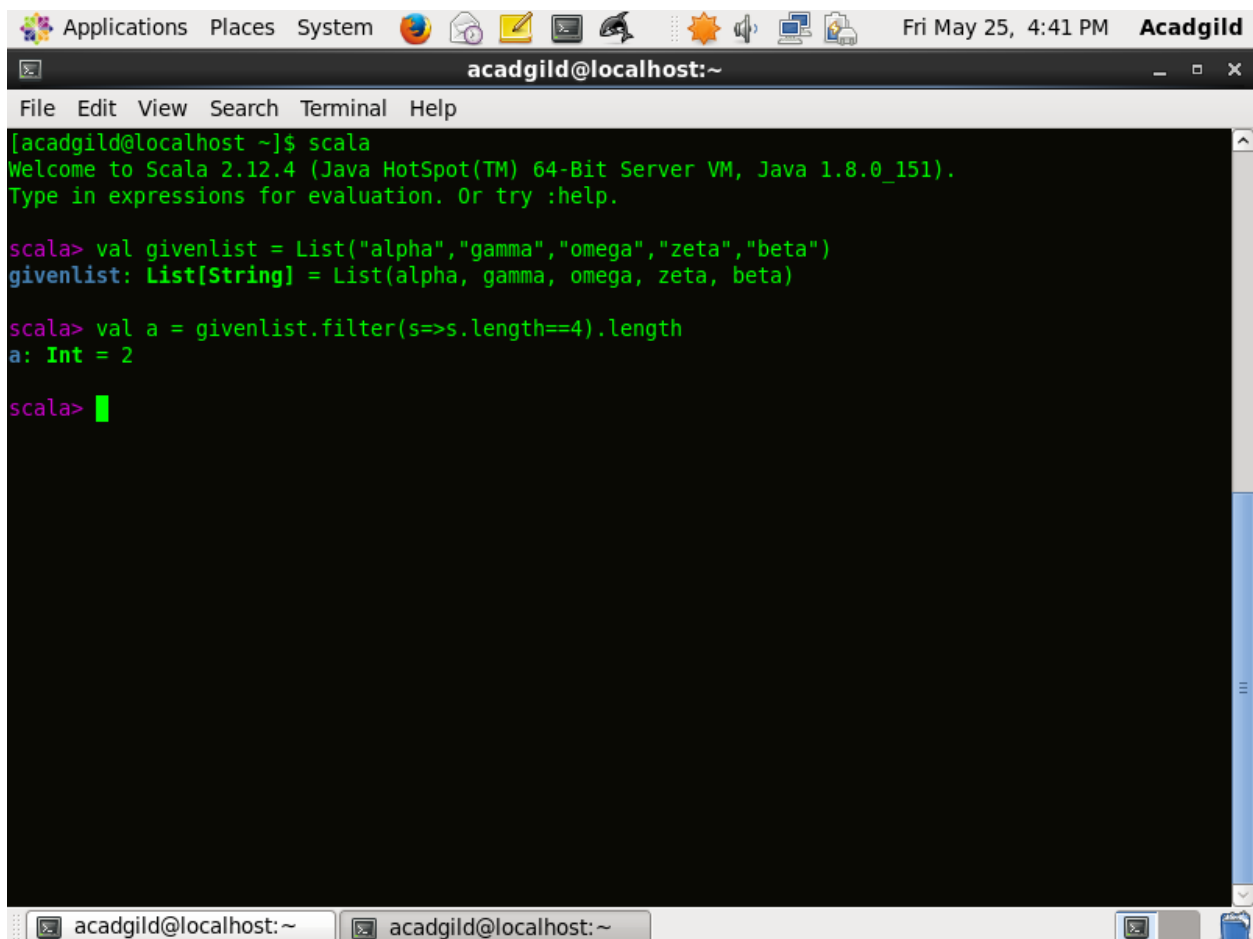
Using scala shell to do the assignment. To start the shell just give the command 'scala'

Usint List keyword to make a new list

Filter function is used on the list with condition that length of string is equal to 4

Using length function on the output of filter function to find the number of stringwith length 4

As one can see the value of 'a' variable is 2 which is our answer

A screenshot of a Linux desktop environment. The top panel shows application icons and the system clock (Fri May 25, 4:41 PM). The main window is a terminal titled 'acadgild@localhost:~'. The terminal output shows the following commands and results:

```
[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 givenlist = List("alpha","gamma","omega","zeta","beta")
givenlist: List[String] = List(alpha, gamma, omega, zeta, beta)

scala> val a = givenlist.filter(s=>s.length==4).length
a: Int = 2

scala>
```

The terminal window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The bottom of the window shows a taskbar with two terminal icons and a system tray with a volume icon and a network icon.

Using variable countmap to store the mapped output of the length of the strings in the list by using the map function and storing the length by using the length function

```
scala> val countmap = givenlist.map(s=>s.length)
countmap: List[Int] = List(5, 5, 5, 4, 4)

scala> println(countmap)
List(5, 5, 5, 4, 4)

scala> █
```

Again using the filter function to check the condition of if string contains character 'm' by using count function on the incoming strings and checking if each character is 'm' and checking if the total count is not equal to zero.

Finally applying length function on the filtered output to get the desired result

```
scala> val b = givenlist.filter(s=>s.count(_=='m')!=0).length
b: Int = 2

scala> println(b)
2

scala> █
```

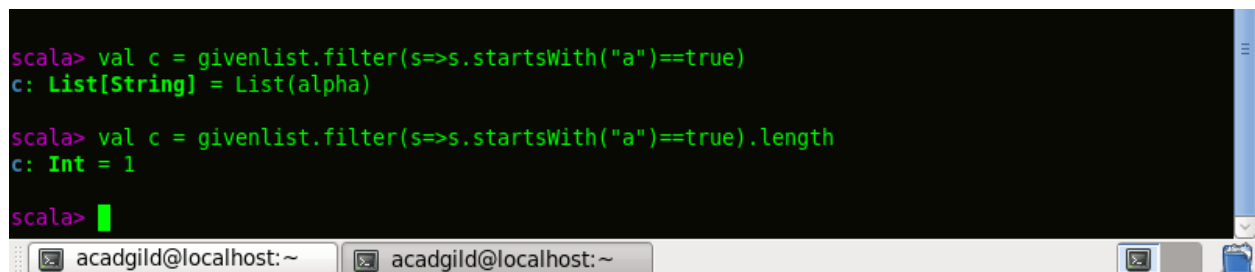
Similar to the previous example, using `startsWith` function on string instead of `count`

And then applying `length` function

```
scala> val c = givenlist.filter(s=>s.startsWith("a")==true)
c: List[String] = List(alpha)

scala> val c = givenlist.filter(s=>s.startsWith("a")==true).length
c: Int = 1

scala> █
```

A screenshot of a Scala REPL window. The window has a dark background with light green text. It shows two lines of code: the first filters a list 'givenlist' for elements starting with 'a', resulting in 'List(alpha)'; the second applies the 'length' function to the filtered list, resulting in '1'. The window title bar shows 'acadgild@localhost: ~'.


Task 2:

Creating a list which contains tuples containing an integer and string

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

scala> println(listoftuples)
List((1,alpha), (2,beta), (3,gamma), (4,zeta), (5,omega))

scala> █
```

A screenshot of a Scala REPL window. The window has a dark background with light green text. It shows three lines of code: the first creates a list of tuples 'listoftuples'; the second prints the list, showing 'List((1,alpha), (2,beta), (3,gamma), (4,zeta), (5,omega))'; the third is a blank line. The window title bar shows 'acadgild@localhost: ~'.

Here we have to print the integers corresponding to the string whose length is 4 so first using filter to remove the tuples according to condition and then using map function to map only the integer values to a list 'a'

```
scala> val a = listoftuples.filter(s=>s._2.length==4).map(s=>s._1)
a: List[Int] = List(2, 4)

scala> println(a)
List(2, 4)

scala> █
```

Again using filter function according to condition and using map to store the list in variable newList

the storing in 'avg' variable the output of sum function divided by length function to get average

```
scala> val newList=listoftuples.filter(s=>(s._2.count(_=='m')!=0 || s._2.count(_=='z')!=0)).map(s=>s._1)
newList: List[Int] = List(3, 4, 5)

scala> val avg = newList.sum/newList.length
avg: Int = 4

scala> println(avg)
4

scala> █
```