

Task 2:

I. Create the List

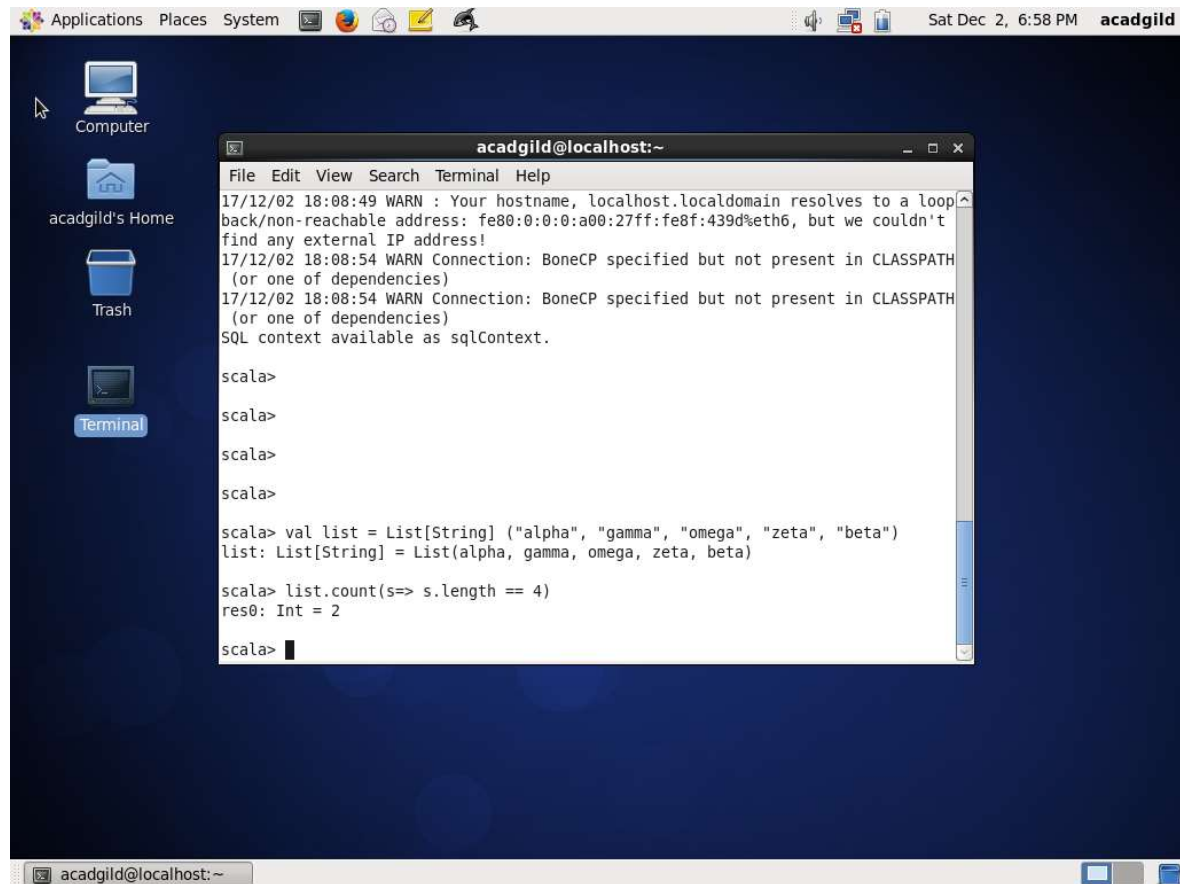
```
val list = List[String] ("alpha", "gamma", "omega", "zeta", "beta")
```

II. Find count of all strings with length 4

Use count function as below

```
list.count(s=> s.length == 4)
```

Screenshot is as below:

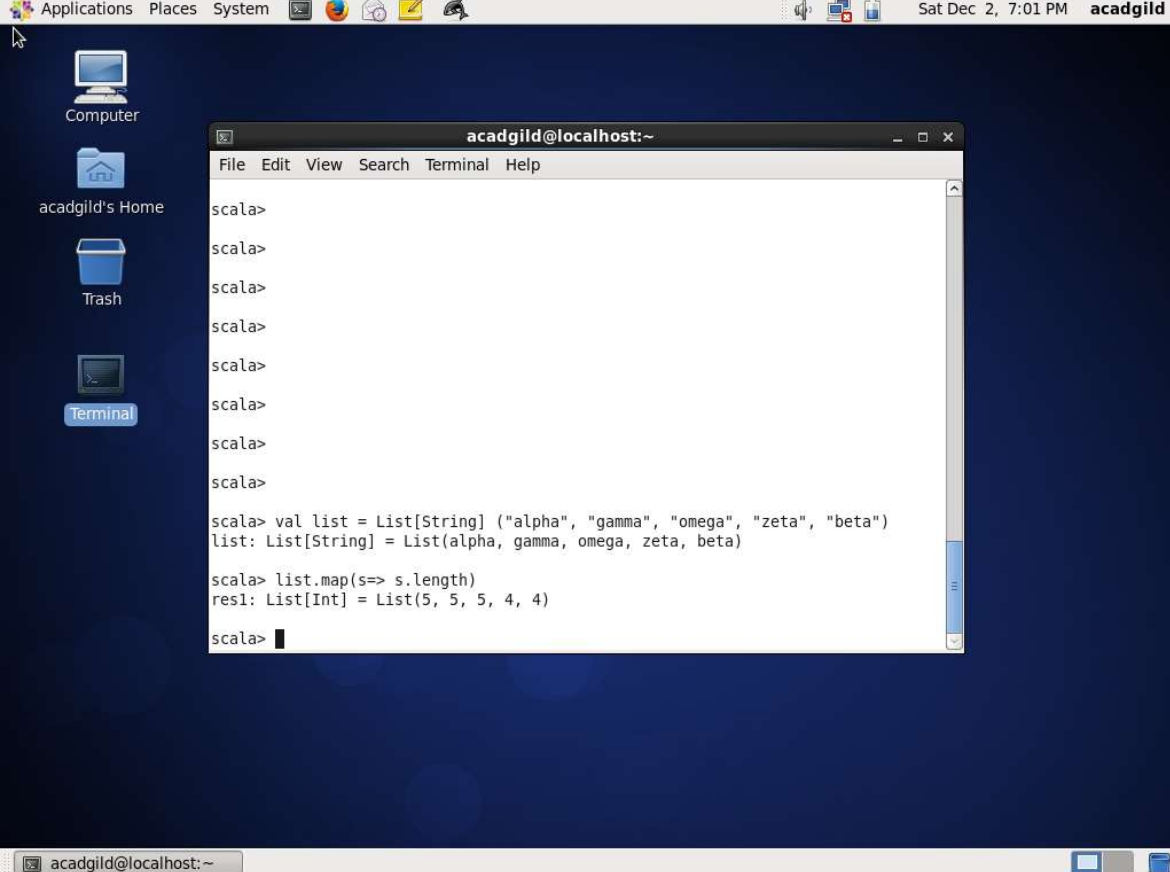


III. Convert the list of string to a list of integers, where each string is mapped to its corresponding length

Use map functions as below:

```
list.map(s=> s.length)
```

Screenshot is as below:

A screenshot of a Linux desktop environment. The desktop background is dark blue. On the left side, there is a sidebar with icons for 'Computer', 'acadgild's Home', 'Trash', and 'Terminal'. The 'Terminal' icon is highlighted. In the center, a terminal window titled 'acadgild@localhost:~' is open. The terminal shows the following Scala code and output:

```
scala>  
scala>  
scala>  
scala>  
scala>  
scala>  
scala>  
scala>  
scala>  
scala> val list = List[String] ("alpha", "gamma", "omega", "zeta", "beta")  
list: List[String] = List(alpha, gamma, omega, zeta, beta)  
  
scala> list.map(s=> s.length)  
res1: List[Int] = List(5, 5, 5, 4, 4)  
scala>
```

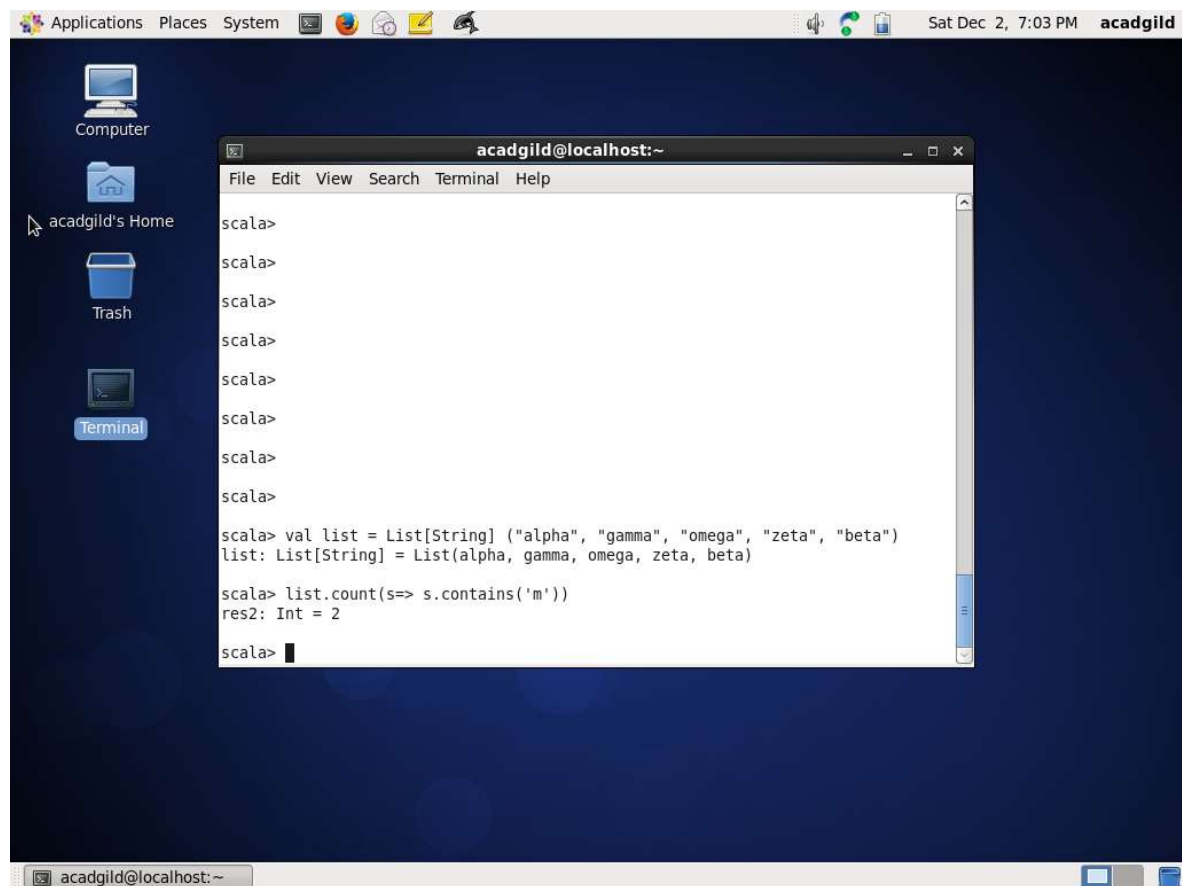
The terminal window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The desktop's top bar shows 'Applications', 'Places', 'System', and system status icons. The system status bar at the bottom shows 'Sat Dec 2, 7:01 PM' and the username 'acadgild'.

IV. Find count of all strings which contain alphabet 'm'

Use count function as below:

```
list.count(s=> s.contains('m'))
```

Screenshot is as below

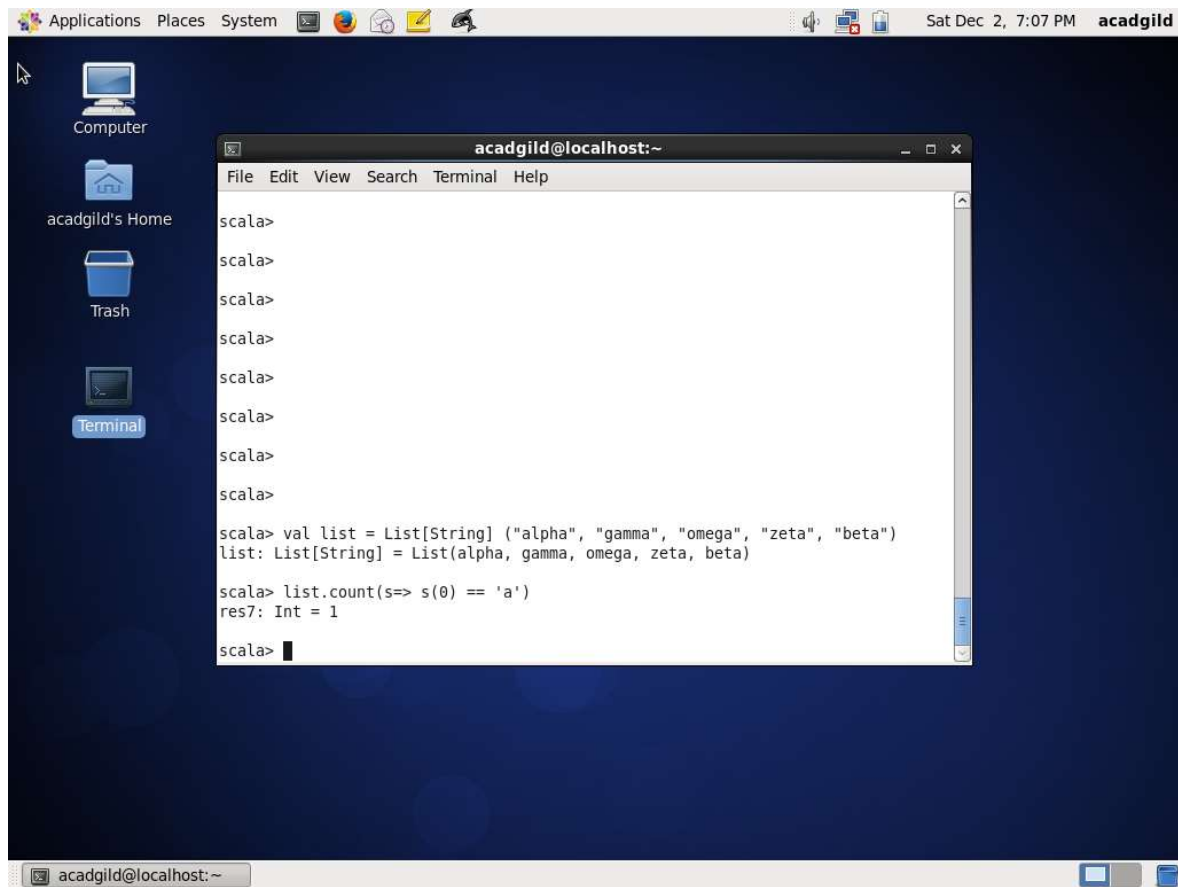


V. Find the count of all strings which start with the alphabet 'a'

Use count function as below:

```
list.count(s=> s(0) == 'a')
```

Screenshot as below:



Task 3

Create a Scala application to find the GCD of two numbers.

I have written a scala method gcd which takes two integer arguments and return an integer. This method first finds greater of two numbers and assign to first_number and second_number respectively. Remainder is calculated by using modulus operator (%) between first_number and second_number. This steps is done repeated in while loop till remainder is 0. When remainder becomes 0, first_number is returned as the gcd. Code is as below:

```
def gcd(a:Int, b:Int) : Int = {  
    var first_number = 0  
    var second_number = 0  
    if (a>b) {  
        first_number = a  
        second_number = b  
    } else {
```

```

    first_number = b
    second_number = a
  }
  var remainder = 1
  while (remainder != 0) {
    remainder = first_number % second_number
    first_number = second_number
    second_number = remainder
  }
  return first_number
}

```

Screenshot is as below:

The screenshot shows a Scala REPL window titled 'acadgild@localhost:~'. The window contains the following code and output:

```

scala>
scala>
scala> def gcd(a:Int, b:Int) : Int = {
|   var first_number = 0
|   var second_number = 0
|   if (a>b) {
|     first_number = a
|     second_number = b
|   } else {
|     first_number = b
|     second_number = a
|   }
|   var remainder = 1
|   while (remainder != 0) {
|     remainder = first_number % second_number
|     first_number = second_number
|     second_number = remainder
|   }
|   return first_number
| }
gcd: (a: Int, b: Int)Int
scala> x = gcd(6,8)
x: Int = 2
scala> x = gcd(8,6)
x: Int = 2
scala> x = gcd(45,30)
x: Int = 15
scala> x = gcd(18,24)
x: Int = 6
scala>

```