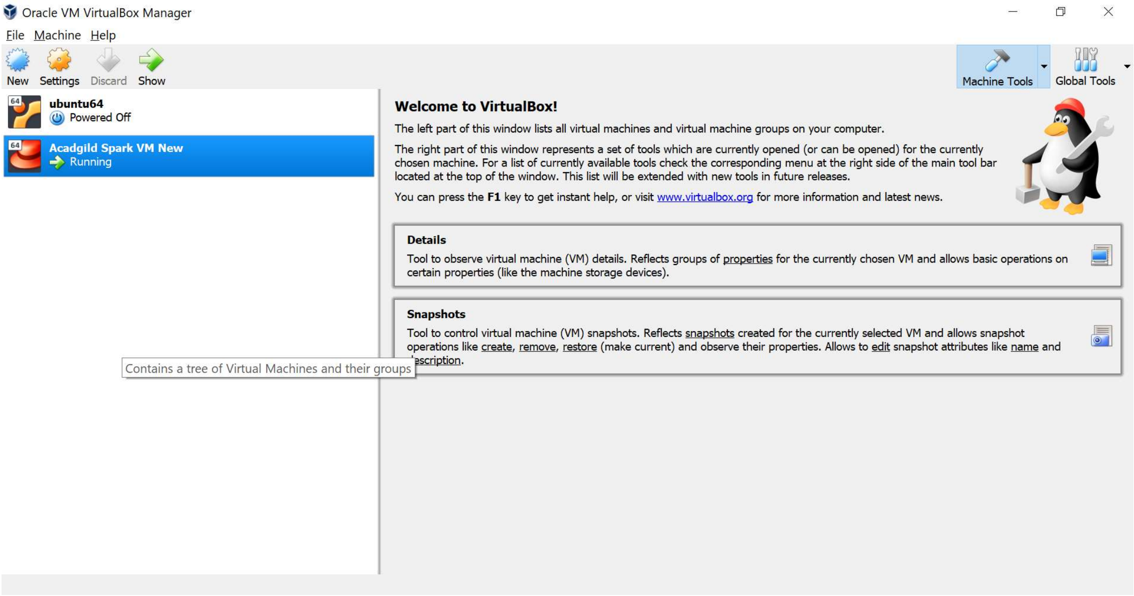
Task 1

Follow the below link document steps to download and import AcadgildSpark VM in the Oracle Virtual Box.

Screenshot of AcadgildSpark VM named “Acadgild Spark VM New” after import is as below:



Welcome to VirtualBox!

The left part of this window lists all virtual machines and virtual machine groups on your computer.

The right part of this window represents a set of tools which are currently opened (or can be opened) for the currently chosen machine. For a list of currently available tools check the corresponding menu at the right side of the main tool bar located at the top of the window. This list will be extended with new tools in future releases.

You can press the **F1** key to get instant help, or visit www.virtualbox.org for more information and latest news.

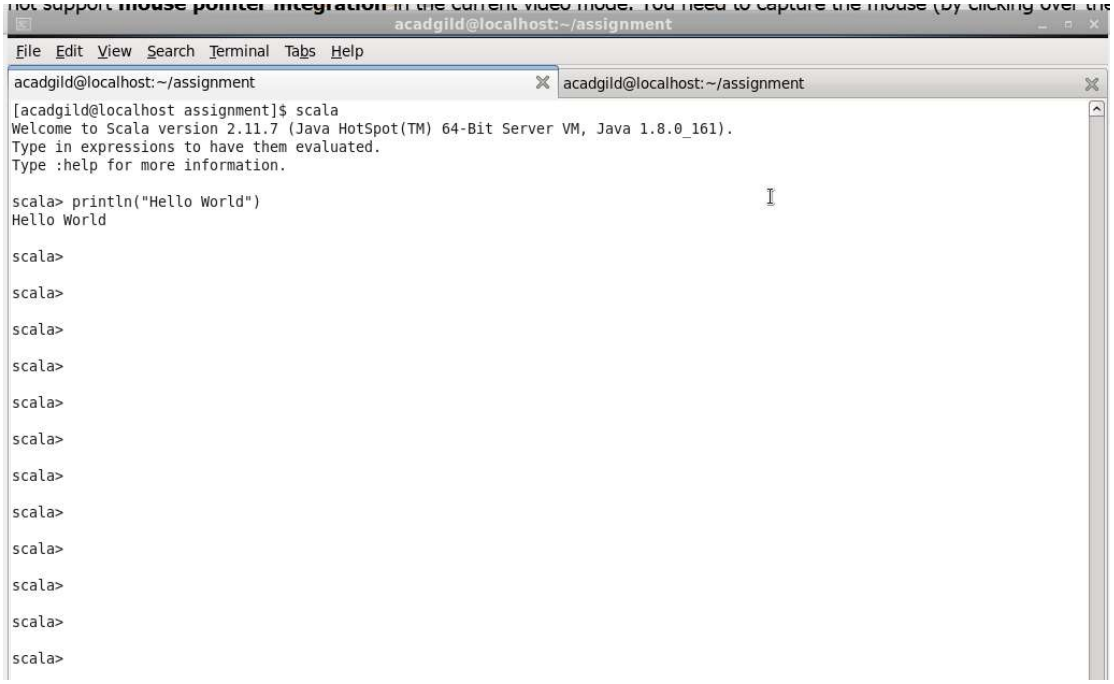
Details

Tool to observe virtual machine (VM) details. Reflects groups of properties for the currently chosen VM and allows basic operations on certain properties (like the machine storage devices).

Snapshots

Tool to control virtual machine (VM) snapshots. Reflects snapshots created for the currently selected VM and allows snapshot operations like create, remove, restore (make current) and observe their properties. Allows to edit snapshot attributes like name and description.

Screenshot of scala shell in Acadgild Spark VM New” with printing “Hello World” is as below:


not support mouse pointer integration in the current video mode. You need to capture the mouse (by clicking over the
acadgild@localhost:~/assignment
File Edit View Search Terminal Tabs Help
acadgild@localhost:~/assignment
acadgild@localhost:~/assignment

```
[acadgild@localhost assignment]$ scala
Welcome to Scala version 2.11.7 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_161).
Type in expressions to have them evaluated.
Type :help for more information.

scala> println("Hello World")
Hello World

scala>

scala>

scala>

scala>

scala>

scala>

scala>

scala>

scala>

scala>

scala>

scala>
```

Task 2:

A. Create a List named x

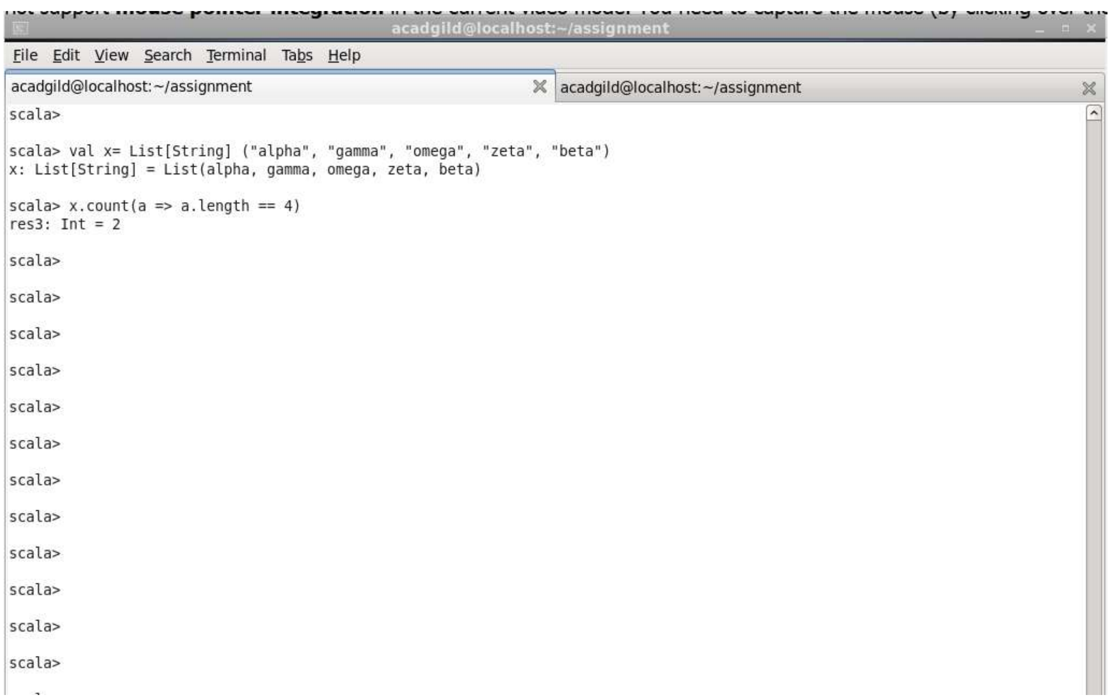
val x = List[String} (“alpha”, “gamma”, “omega”, “zeta”, “beta”)

B. Find count of all strings with length 4 in list x

Use count function as below with filter

x.count(a=> a.length == 4)

Screenshot is as below:


not support mouse pointer integration in the current video mode. You need to capture the mouse (by clicking over the
acadgild@localhost:~/assignment
File Edit View Search Terminal Tabs Help
acadgild@localhost:~/assignment
acadgild@localhost:~/assignment

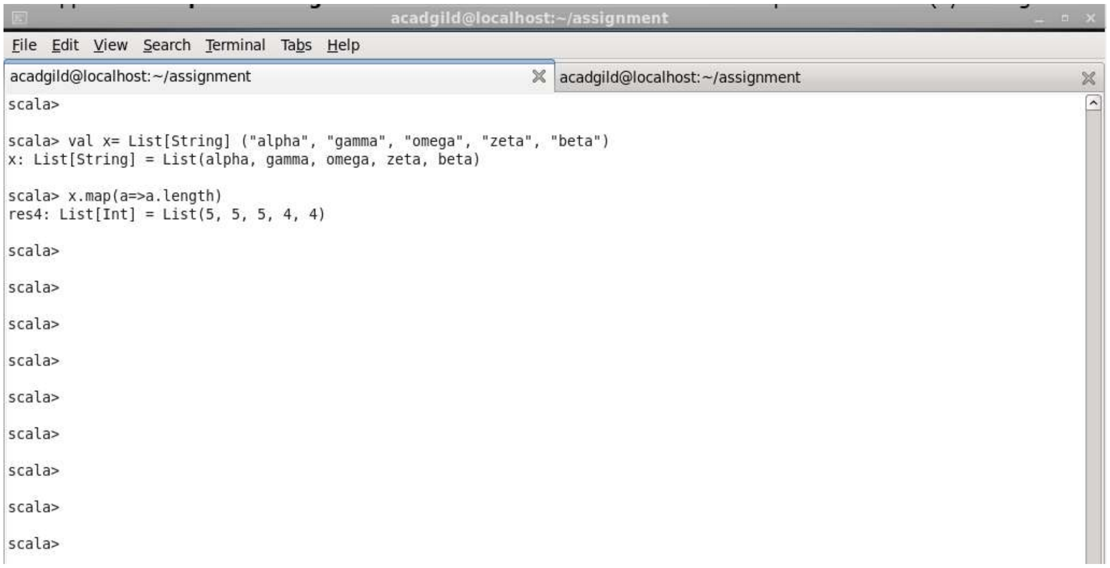
```
scala>

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

scala> x.count(a => a.length == 4)
res3: Int = 2

scala>

scala>

scala>

scala>

scala>

scala>

scala>

scala>

scala>

scala>

scala>

scala>
```

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

- Create the list
- Use map function is used to map to length of each element in string as below:

x.map(a=> a.length)

Screenshot is as below:



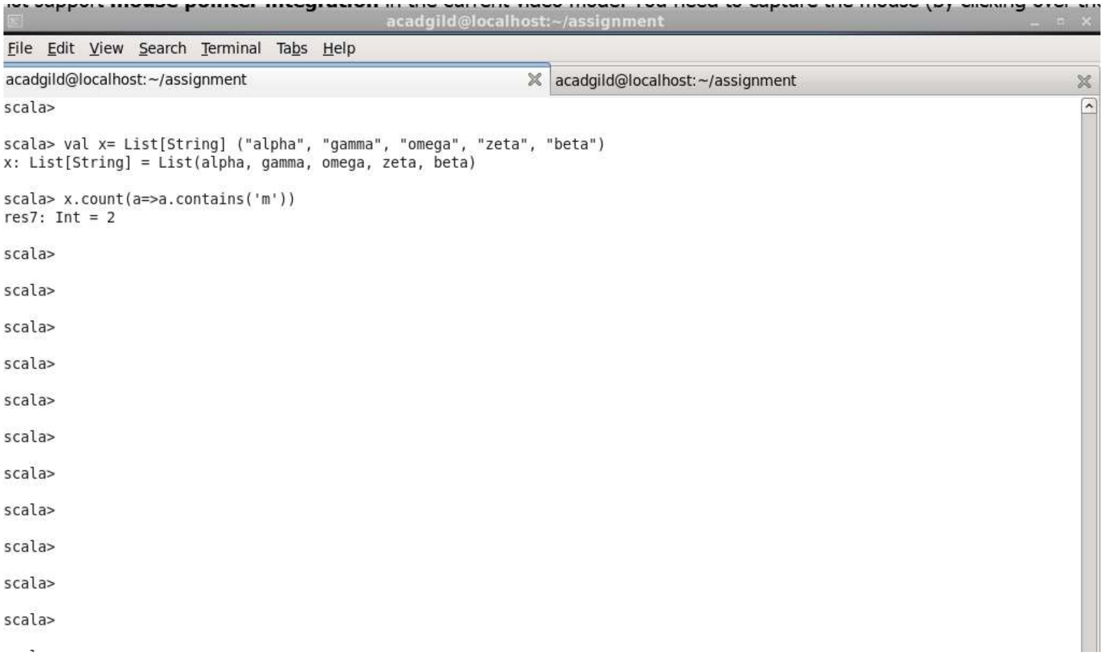
acadgild@localhost:~/assignment
File Edit View Search Terminal Tabs Help
acadgild@localhost:~/assignment
acadgild@localhost:~/assignment

```
scala>

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

scala> x.map(a=>a.length)
res4: List[Int] = List(5, 5, 5, 4, 4)

scala>

scala>

scala>

scala>

scala>

scala>

scala>

scala>

scala>
```

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

- Use count function to get only the string that contain ‘m’ as below:

x.count(a=> a.contains(‘m’))


acadgild@localhost:~/assignment
File Edit View Search Terminal Tabs Help
acadgild@localhost:~/assignment
acadgild@localhost:~/assignment

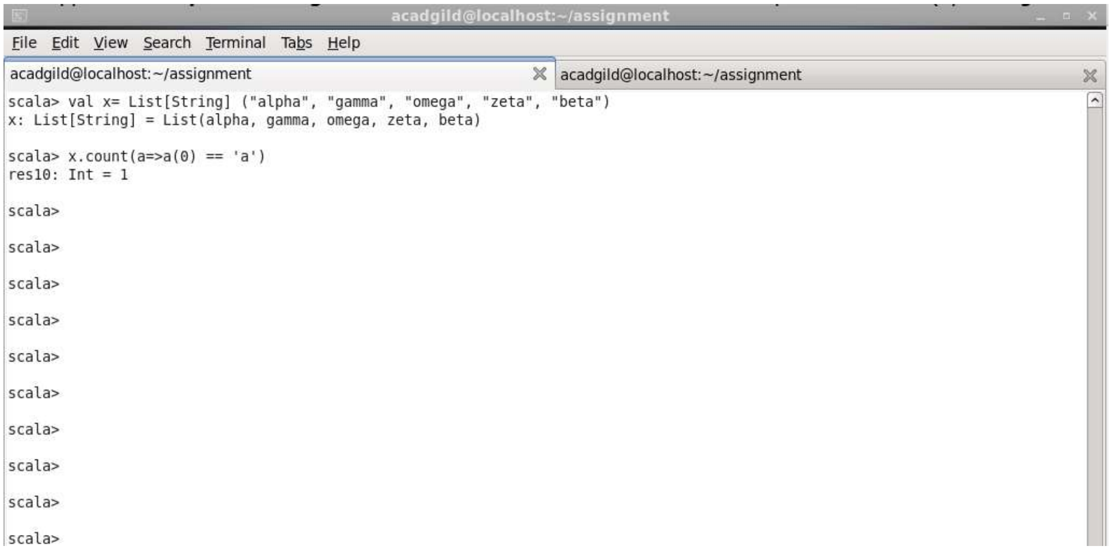
```
scala>

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

scala> x.count(a=>a.contains('m'))
res7: Int = 2

scala>

scala>

scala>

scala>

scala>

scala>

scala>

scala>

scala>

scala>

scala>
```

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

- Use count function to get only the strings whose first character is ‘a’ as below:

x.count(a=> a(0) == ‘a’)

Screenshot as below:


acadgild@localhost:~/assignment
File Edit View Search Terminal Tabs Help
acadgild@localhost:~/assignment
acadgild@localhost:~/assignment

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

scala> x.count(a=>a(0) == 'a')
res10: Int = 1

scala>

scala>

scala>

scala>

scala>

scala>

scala>

scala>

scala>

scala>
```

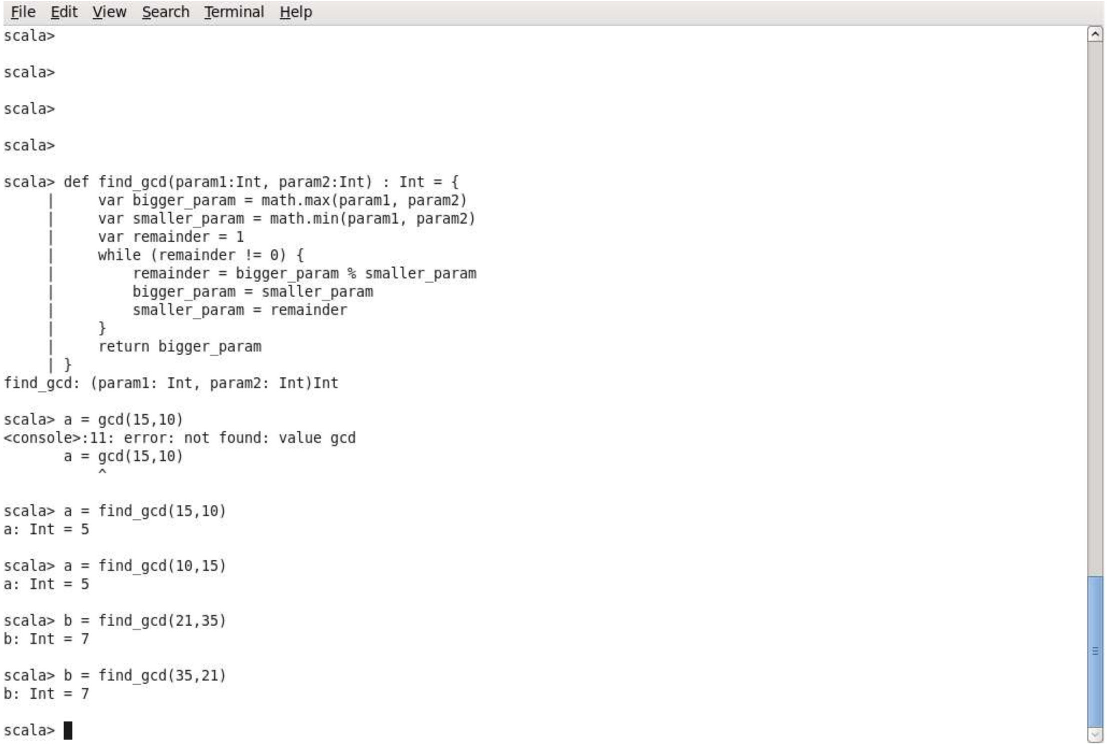
Task 3

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

```
-   Take two parameters whose gcd to be determined
-   Get the bigger and smaller parameter from the two numbers
-   Initialize remainder to 1
-   Repeat the next three steps below until remainder is 0
-       Get modulus value of bigger number and smaller number and assign to remainder
-       Assign smaller number to bigger number
-       Assign remainder to smaller number
-
-   Return bigger number
```

```
def find_gcd(param1:Int, param2:Int) : Int = {

    var bigger_param = math.max(param1, param2)

   var smaller_param = math.min(param1, param2)

    var remainder = 1

    while (remainder != 0) {

      remainder = bigger_param % smaller_param

      bigger_param = smaller_param

      smaller_param = remainder

    }

    return bigger_param

}
```

Screenshot is as below:



```
scala> 

scala> 

scala> 

scala> 

scala> def find_gcd(param1:Int, param2:Int) : Int = {
     |     var bigger_param = math.max(param1, param2)
     |     var smaller_param = math.min(param1, param2)
     |     var remainder = 1
     |     while (remainder != 0) {
     |         remainder = bigger_param % smaller_param
     |         bigger_param = smaller_param
     |         smaller_param = remainder
     |     }
     |     return bigger_param
     | }
find_gcd: (param1: Int, param2: Int)Int

scala> a = gcd(15,10)
<console>:11: error: not found: value gcd
       a = gcd(15,10)
           ^

scala> a = find_gcd(15,10)
a: Int = 5

scala> a = find_gcd(10,15)
a: Int = 5

scala> b = find_gcd(21,35)
b: Int = 7

scala> b = find_gcd(35,21)
b: Int = 7

scala> 
```