Task 1

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

– Find count of all strings with length 4.

Command

```
object mainList {


  def main(args:Array[String]){

  val liststring: List[String] =  List("alpha","gamma","omega","zeta","beta") ;
  println("list of tuples = "+liststring );


  val lenght4 = liststring.count(item=>item.length==4);
 println("String with lenght 4 is " +lenght4);
```

Output

```
Console
<terminated> mainList$ [Scala Application] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (Aug 14, 2018, 1:49
list of tuples = List(alpha, gamma, omega, zeta, beta)
String with lenght 4 is 2
```

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

Command

```
val lengthofeachstring = liststring.map(item=>item.length);
 println( "the list of string to a list of integers = "+ lengthofeachstring);
```

Output

```
<terminated> mainList$ [Scala Application] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (Aug 14, 2018, 1:52
list of tuples = List(alpha, gamma, omega, zeta, beta)
the list of string to a list of integers = List(5, 5, 5, 4, 4)
```

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

Command

```
val containsm = liststring.count(item=>item.contains("m"));
println("the count of all strings which contain alphabet ‘m’ = "+ containsm);
```

Output

```
Console
<terminated> mainList$ [Scala Application] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (Aug 14, 2018
list of tuples = List(alpha, gamma, omega, zeta, beta)
the count of all strings which contain alphabet ‘m’ = 2
```

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

Command

```
val startswitha = liststring.count(item=>item.startsWith("a"));
println( "the count of all strings which start with the alphabet ‘a’ = "+startswitha);
```

Output

```
Console
<terminated> mainList$ [Scala Application] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (Aug 14, 2018, 1:58:16 AM)
list of tuples = List(alpha, gamma, omega, zeta, beta)
the count of all strings which start with the alphabet ‘a’ = 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'))

Command

list.scala | task2.scala

```
package task5_acadgild

import scala.collection.immutable._

object mainTask2 {


  def main(args:Array[String]){
  val tupledata = ((1,"alpha"),(2,"gamma"),(3,"beta"),(4,"zeta"),(5,"omega"))
  println("List of Tuple " + tupledata)
  }
}
```

Output

Console

```
<terminated> mainTask2$ [Scala Application] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (Aug 14, 201
List of Tuple ((1,alpha),(2,gamma),(3,beta),(4,zeta),(5,omega))
```

Command Prompt

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

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

```
scala> list.filter(x=>(x._2.length==4))
res2: List[(Int, String)] = List((3,beta), (4,zeta))

scala> list.filter(x=>(x._2.length==4)).foreach(x=>println("the corresponding number  string is
= ",x._1))
(the corresponding number  string is = ,3)
(the corresponding number  string is = ,4)
```

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

Step1 :

```
scala> val contains_m_z=list.filter{x=>(x._2.contains("m"))||(x._2.contains("z"))}
contains_m_z: List[(Int, String)] = List((2,gamma), (4,zeta), (5,omega))
```

Step2

```
scala> val tuple_sum=contains_m_z.map(x=>(x._1)).sum
tuple_sum: Int = 11
```

Step 3

```
scala> val length=contains_m_z.length
length: Int = 3
```

Step 4

```
scala> val avg = tuple_sum/length
avg: Int = 3
```

Task 3

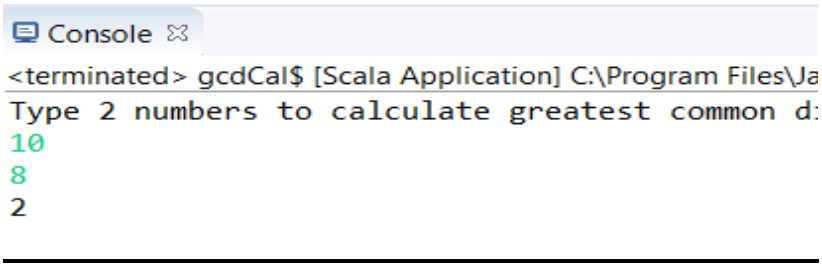
Create a Scala application to find the GCD of two numbers

Code:

list.scala | task2.scala | calculatortest.scala | gcd.scala

```
package task5_acadgild

object gcdCal {

  def gcd(a: Int,b: Int): Int = {

       if(b ==0) a else gcd(b, a%b)

    }
    def main(args: Array[String]) {

      println("Type 2 numbers to calculate greatest common divisor")
      var x =readInt()
      var y =readInt()

        println(gcd(x,y))

    }
}
```

Output



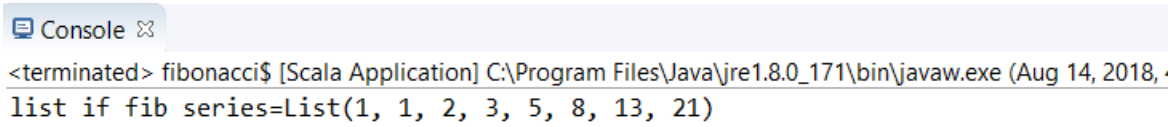
Task 4

Fibonacci series (starting from 1) written in order without any spaces in between, thus producing a sequence of digits.

Code

```
def fibSequence(n: Int): List[Int] = {
    var ret = scala.collection.mutable.ListBuffer[Int](1, 1)
    while (ret(ret.length - 1) < n) {
      val temp = ret(ret.length - 1) + ret(ret.length - 2)
      if (temp >= n) {
        return ret.toList
      }
      ret += temp
    }
    ret.toList
  }
```

Output



Write a Scala application to find the Nth digit in the sequence.

➢ Write the function using standard for loop

Code to find nth digit in the sequence using for loop

```
def nthFib(n: Int): Int = {
  var x = 0
  var y = 1
  for (_ <- 1 until n) {
    val temp = x + y
    x = y
    y = temp
  }
  y
}
```

Output

```
Console
<terminated> fibonacci$ [Scala Application] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (Aug 14, 2018, 4
Type the number to find the digit in sequence:
5
5 fibonacci value using for loop = 5
```

➢ **Write the function using recursion**

Code to find nth digit in the sequence using recursion

```
def fib_rec(n:Long):Long = {
  def fib_recursion(n:Long, a:Long, b:Long):Long = {
    if(n == 0) a
    else fib_recursion(n - 1, b, a + b)
  }
  return fib_recursion(n, 0, 1)
}
```

Output

```
Console
<terminated> fibonacci$ [Scala Application] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (Aug 14, 2018, 5:06
Type the number to find the digit in sequence:
7
7 fibonacci value using recursion function =13
```

Task 5

Find square root of number using Babylonian method.
1. Start with an arbitrary positive start value x (the closer to the root, the better).
2.Initialize y = 1.
3. Do following until desired approximation is achieved. a) Get the next approximation for root using average of x and y b) Set y = n/x

list.scala | task2.scala | calculatortest.scala | square.scala

```scala
package task5_acadgild

object square {
 def squareRoot(n : Double) : Double =
    {
        var x = n : Double;
        var y = 1 : Double;
        val e = 0.000001 : Double;    // e decides the accuracy level

        while(x - y > e)
        {
            x = (x + y)/2;
            y = n/x;
        }
        return x;
    }

      def main(args : Array[String])
    {
     println("Type the number to get Square root value : ")
     val n = readDouble()
     println("Square root of " + n + " is " + squareRoot(n));
    }
}
```

Output

Console

<terminated> square$ [Scala Application] C:\Program Files\Java\jre1.8.0_171\bin\javaw.e

```
Type the number to get Square root value :
36
Square root of 36.0 is 6.000000005333189
```