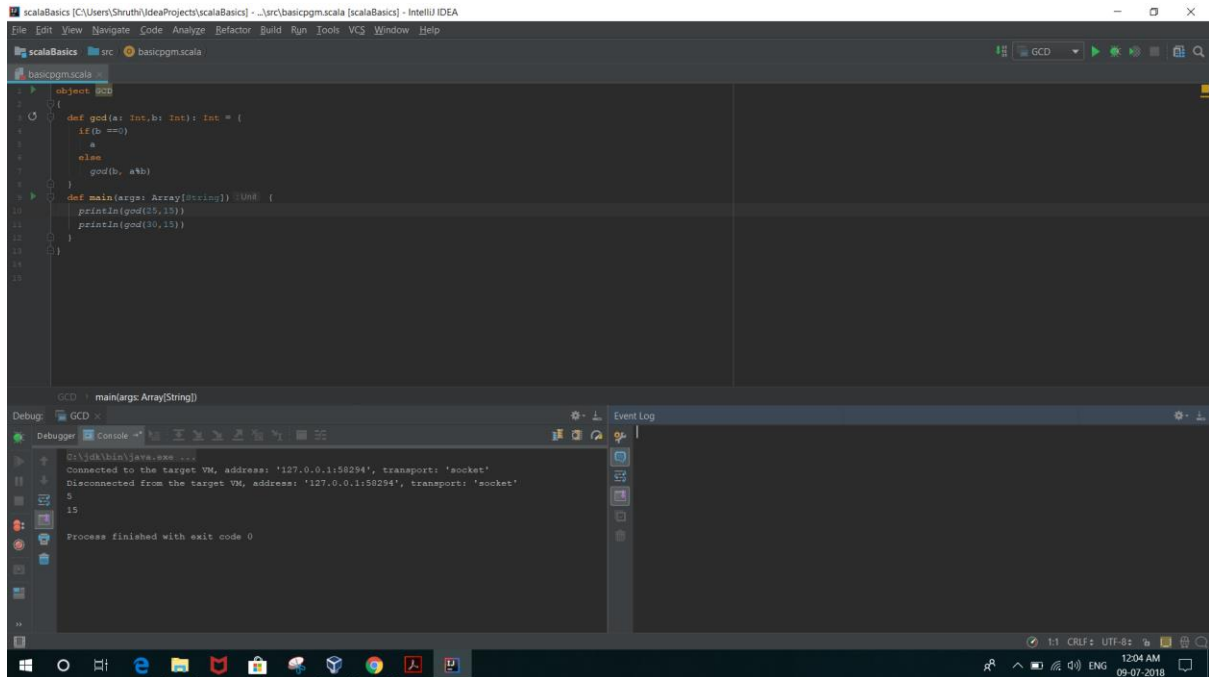


Assignment 15 : Scala Basics 2

Task 1

Create a Scala application to find the GCD of two numbers

Please find the below code for GCD of two numbers.



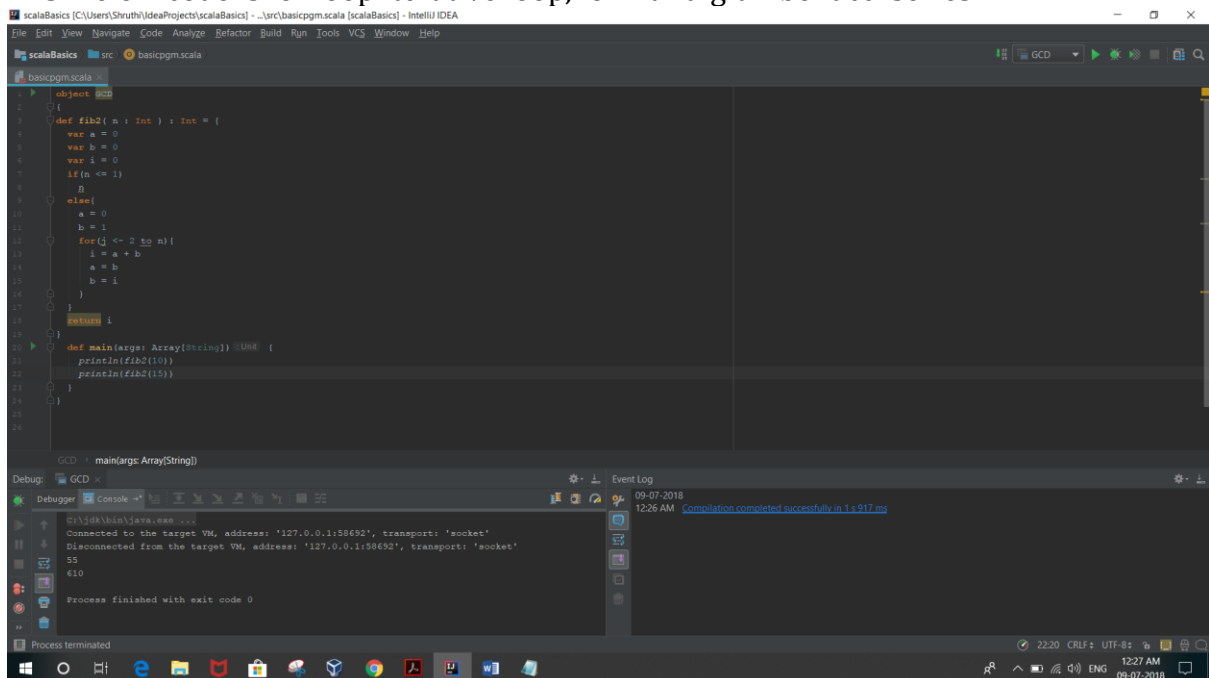
Task 2

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

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

➤ Write the function using standard for loop

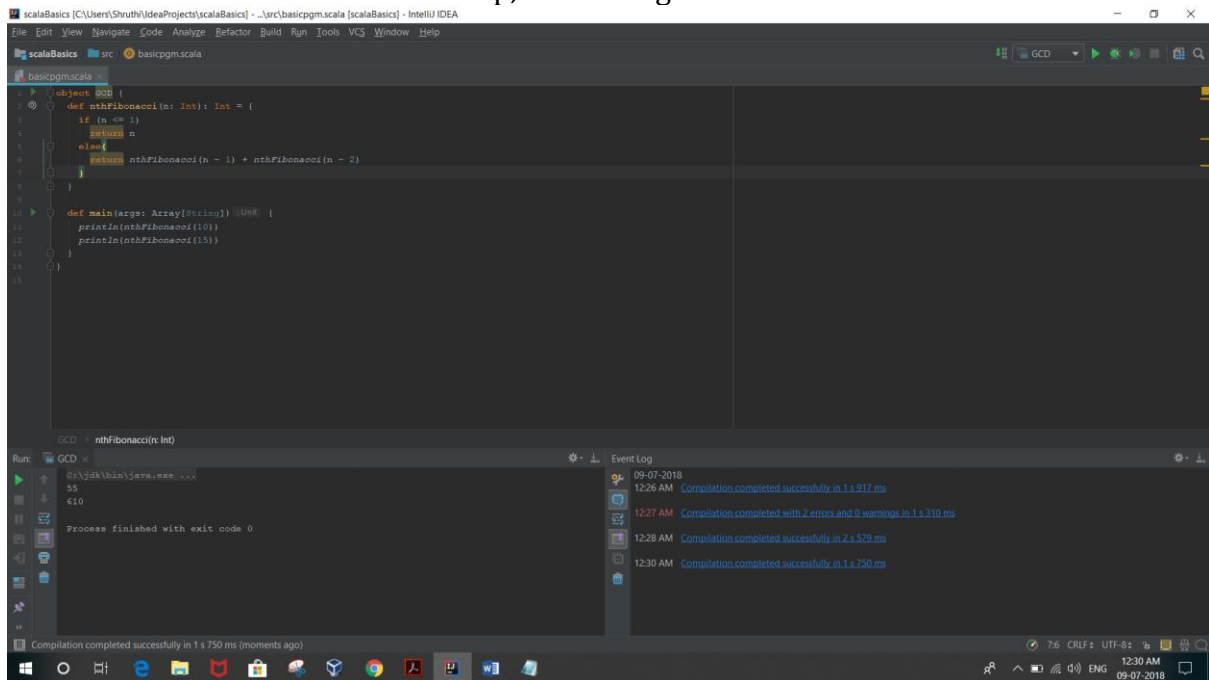
Ans: Below code is for for loop iterative loop, for Nth digit Fibonacci series



Assignment 15 : Scala Basics 2

➤ Write the function using recursion

Ans: Below code is for recursive loop, for Nth digit Fibonacci series



```
scalaBasics [C:\Users\Shruthi\IdeaProjects\scalaBasics] - \src\basicpgm.scala [scalaBasics] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

scalaBasics src basicpgm.scala
basicpgm.scala
object GCD {
  def nthFibonacci(n: Int): Int = {
    if (n <= 1)
      return n
    else
      nthFibonacci(n - 1) + nthFibonacci(n - 2)
  }

  def main(args: Array[String]) {
    printIn(nthFibonacci(10))
    printIn(nthFibonacci(15))
  }
}

Run: GCD x
C:\jdk\bin\java.exe ...
55
610
Process finished with exit code 0

Event Log
09-07-2018
12:26 AM Compilation completed successfully in 1 s 917 ms
12:27 AM Compilation completed with 2 errors and 0 warnings in 1 s 310 ms
12:28 AM Compilation completed successfully in 2 s 579 ms
12:30 AM Compilation completed successfully in 1 s 750 ms

Compilation completed successfully in 1 s 750 ms (moments ago)
7-6 CRLF UTF-8 12:30 AM 09-07-2018
```

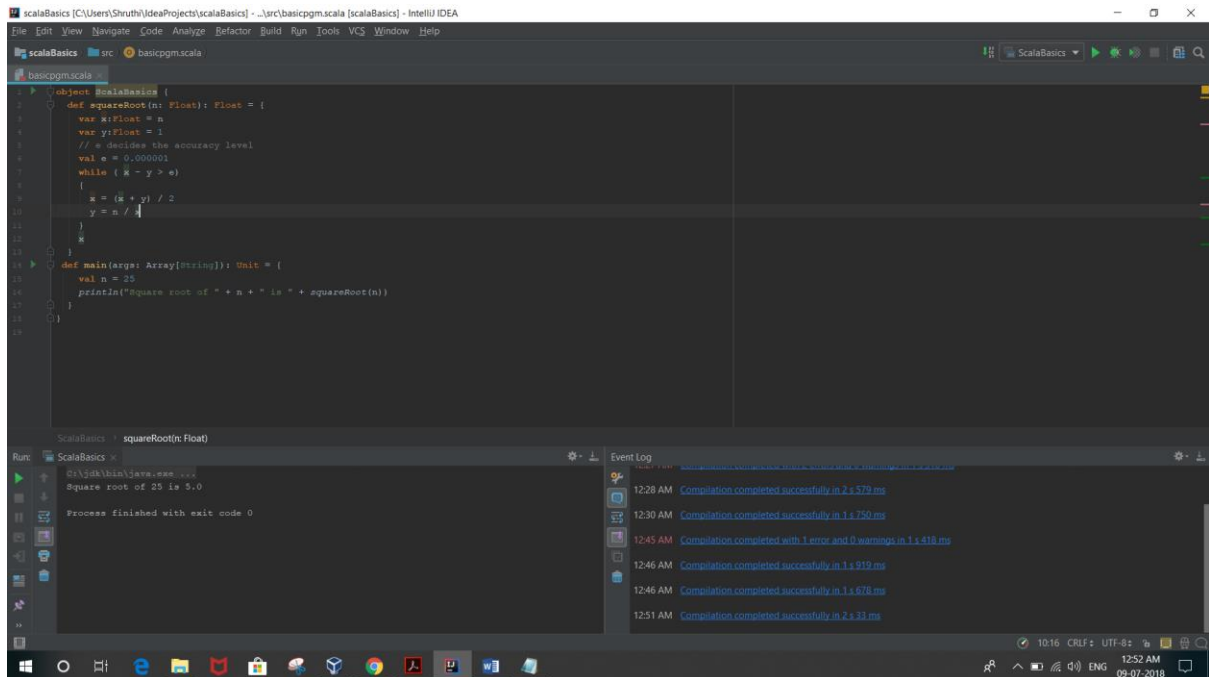
Assignment 15 : Scala Basics 2

Task 3

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 $y = n/x$

Ans: please find the code & answer for the given question



```
object ScalaBasics {  
  def squareRoot(n: Float): Float = {  
    var x: Float = n  
    var y: Float = 1  
    // e decides the accuracy level  
    val e = 0.000001  
    while ((x - y) > e) {  
      x = (x + y) / 2  
      y = n / x  
    }  
    x  
  }  
}  
  
def main(args: Array[String]): Unit = {  
  val n = 25  
  println("Square root of " + n + " is " + squareRoot(n))  
}
```

Run: ScalaBasics x
E:\Java\bin\idea.exe ...
Square root of 25 is 5.0
Process finished with exit code 0

Event Log
12:28 AM Compilation completed successfully in 1 s 579 ms
12:30 AM Compilation completed successfully in 1 s 750 ms
12:45 AM Compilation completed with 1 error and 0 warnings in 1 s 418 ms
12:46 AM Compilation completed successfully in 1 s 919 ms
12:46 AM Compilation completed successfully in 1 s 678 ms
12:51 AM Compilation completed successfully in 2 s 33 ms