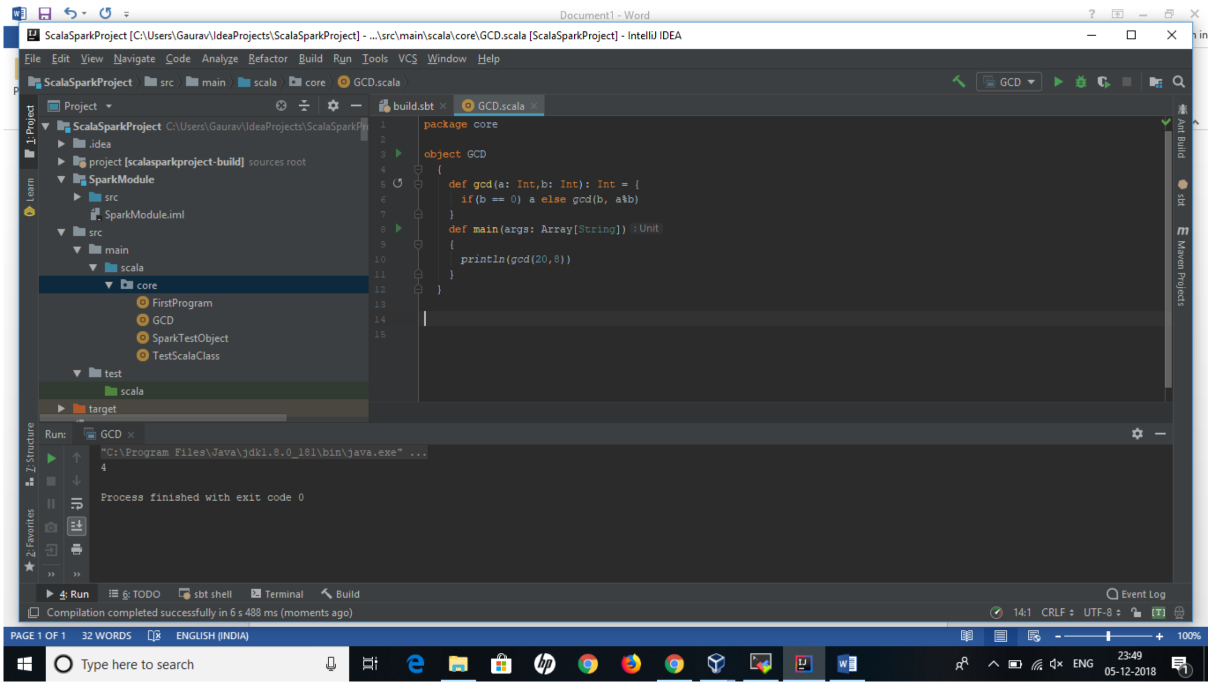
Assignment 15.1 Scala 2

Task 1: Create a Scala application to find the GCD of two numbers.

Solution: We have used the Intellij to do this program. The Program is mentioned below:

```scala
package core

object GCD
  {
    def gcd(a: Int,b: Int): Int = {
      if(b == 0) a else gcd(b, a%b)
    }
    def main(args: Array[String])
    {
      println(gcd(20,8))
    }
  }
```

Output:



ScalaSparkProject [C:\Users\Gaurav\IdeaProjects\ScalaSparkProject] - ...\src\main\scala\core\GCD.scala [ScalaSparkProject] - IntelliJ IDEA

```scala
package core

object GCD
  {
    def gcd(a: Int,b: Int): Int = {
      if(b == 0) a else gcd(b, a%b)
    }
    def main(args: Array[String])
    {
      println(gcd(20,8))
    }
  }
```

```
"C:\Program Files\Java\jdk1.8.0_181\bin\java.exe" ...
4

Process finished with exit code 0
```

Compilation completed successfully in 6 s 488 ms (moments ago)

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.

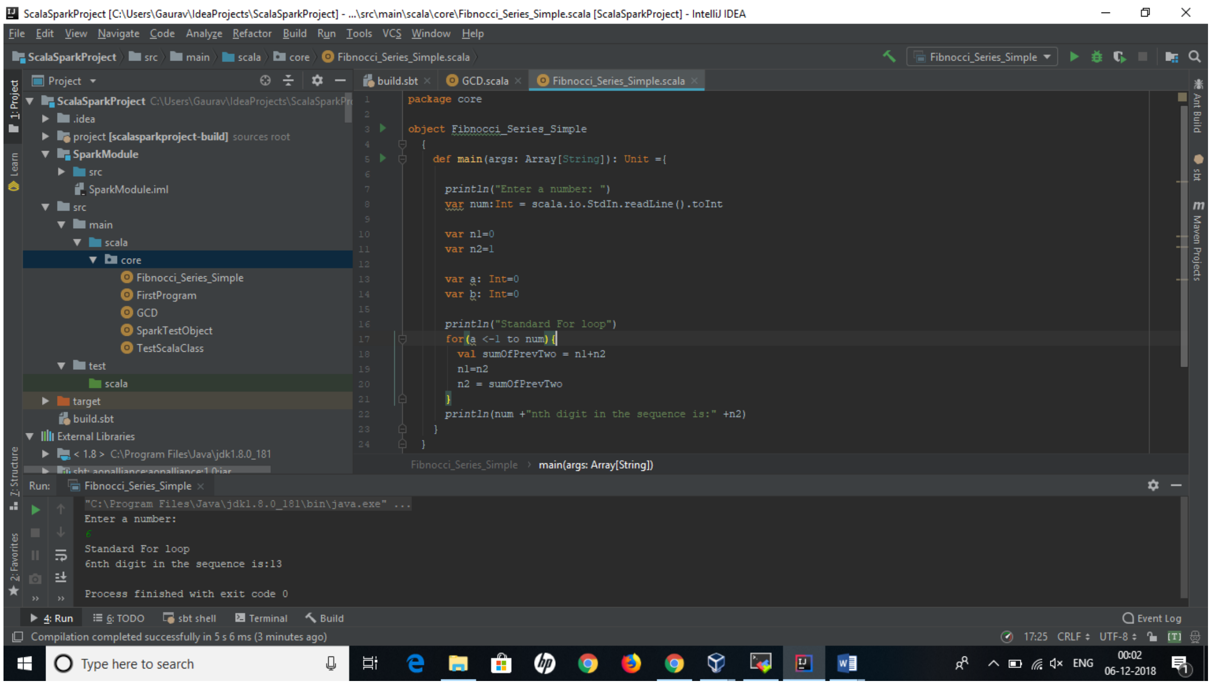
- Write the function using standard for loop

Program:

```scala
package core

object Fibnocci_Series_Simple
  {
    def main(args: Array[String]): Unit ={
```

```
    println("Enter a number: ")
    var num:Int = scala.io.StdIn.readLine().toInt

    var n1=0
    var n2=1

    var a: Int=0
    var b: Int=0

    println("Standard For loop")
    for(a <-1 to num){
      val sumOfPrevTwo = n1+n2
      n1=n2
      n2 = sumOfPrevTwo
    }
    println(num +"nth digit in the sequence is:" +n2)
  }
}
```

Output:


ScalaSparkProject [C:\Users\Gaurav\IdeaProjects\ScalaSparkProject] - ...\src\main\scala\core\Fibnocci_Series_Simple.scala [ScalaSparkProject] - IntelliJ IDEA

```
package core

object Fibnocci_Series_Simple
  {
    def main(args: Array[String]): Unit ={

      println("Enter a number: ")
      var num:Int = scala.io.StdIn.readLine().toInt

      var n1=0
      var n2=1

      var a: Int=0
      var b: Int=0

      println("Standard For loop")
      for(a <-1 to num){
        val sumOfPrevTwo = n1+n2
        n1=n2
        n2 = sumOfPrevTwo
      }
      println(num +"nth digit in the sequence is:" +n2)
    }
  }
```

```
"C:\Program Files\Java\jdk1.8.0_181\bin\java.exe" ...
Enter a number:
6
Standard For loop
6nth digit in the sequence is:13

Process finished with exit code 0
```

Compilation completed successfully in 5 s 6 ms (3 minutes ago)

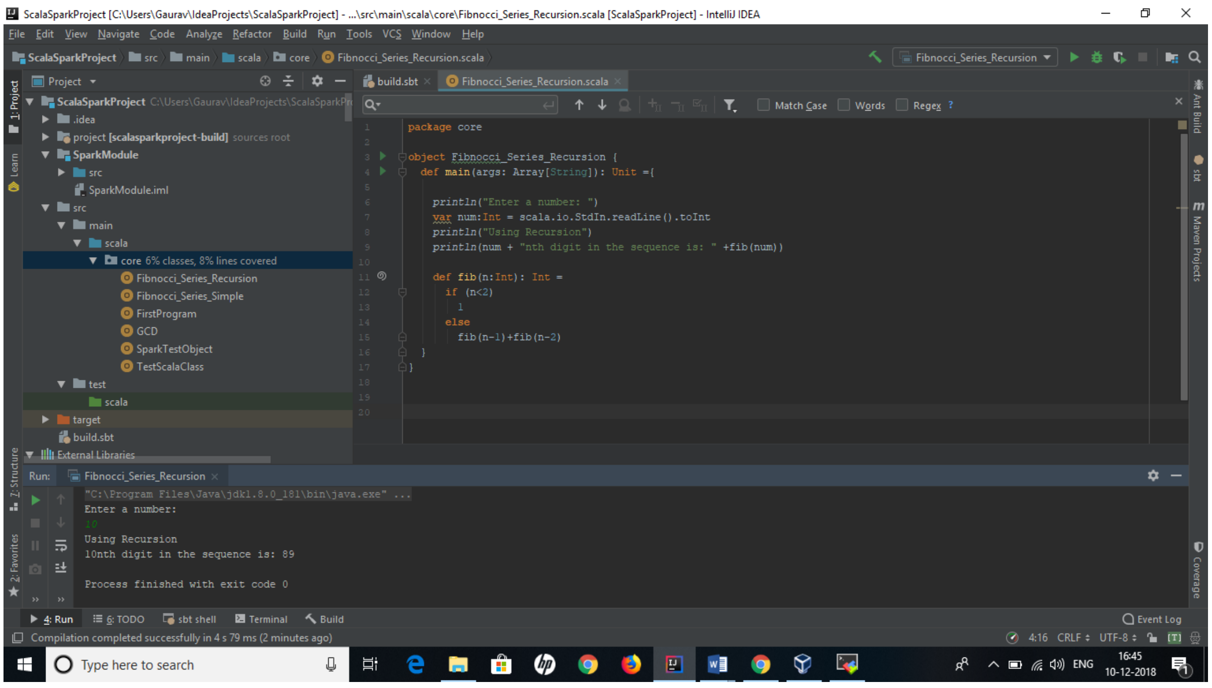
- Write the function using recursion

Program:

```
package core

object Fibnocci_Series_Recursion {
  def main(args: Array[String]): Unit ={

    println("Enter a number: ")
    var num:Int = scala.io.StdIn.readLine().toInt
    println("Using Recursion")
    println(num + "nth digit in the sequence is: " +fib(num))

    def fib(n:Int): Int =
      if (n<2)
```

```
          1
        else
          fib(n-1)+fib(n-2)
    }
}
```

Output:



ScalaSparkProject [C:\Users\Gaurav\IdeaProjects\ScalaSparkProject] - ...\src\main\scala\core\Fibnocci_Series_Recursion.scala [ScalaSparkProject] - IntelliJ IDEA

```
package core

object Fibnocci_Series_Recursion {
  def main(args: Array[String]): Unit ={

    println("Enter a number: ")
    var num:Int = scala.io.StdIn.readLine().toInt
    println("Using Recursion")
    println(num + "nth digit in the sequence is: " +fib(num))

    def fib(n:Int): Int =
      if (n<2)
        1
      else
        fib(n-1)+fib(n-2)
  }
}
```

```
"C:\Program Files\Java\jdk1.8.0_181\bin\java.exe" ...
Enter a number:
10
Using Recursion
10nth digit in the sequence is: 89

Process finished with exit code 0
```

Compilation completed successfully in 4 s 79 ms (2 minutes ago)

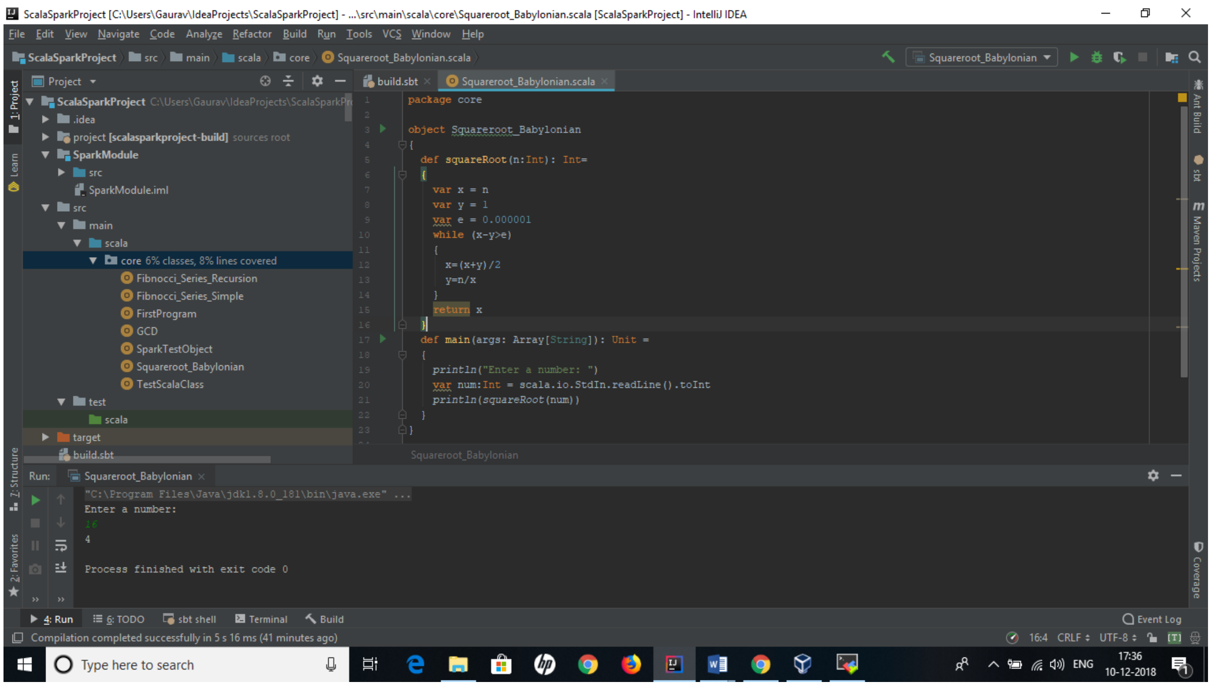
Task 3: Find square root of number using Babylonian method.

1. 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

Program:

```
package core

object Squareroot_Babylonian
{
  def squareRoot(n:Int): Int=
  {
    var x = n
    var y = 1
    var e = 0.000001
    while (x-y>e)
    {
      x=(x+y)/2
      y=n/x
    }
    return x
  }
  def main(args: Array[String]): Unit =
  {
    println("Enter a number: ")
    var num:Int = scala.io.StdIn.readLine().toInt
    println(squareRoot(num))
  }
}
```

Output:



ScalaSparkProject [C:\Users\Gaurav\IdeaProjects\ScalaSparkProject] - ...\src\main\scala\core\Squareroot_Babylonian.scala [ScalaSparkProject] - IntelliJ IDEA

```
package core

object Squareroot_Babylonian
{
  def squareRoot(n:Int): Int=
  {
    var x = n
    var y = 1
    var e = 0.000001
    while (x-y>e)
    {
      x=(x+y)/2
      y=n/x
    }
    return x
  }
  def main(args: Array[String]): Unit =
  {
    println("Enter a number: ")
    var num:Int = scala.io.StdIn.readLine().toInt
    println(squareRoot(num))
  }
}
```

```
"C:\Program Files\Java\jdk1.8.0_181\bin\java.exe" ...
Enter a number:
16
4

Process finished with exit code 0
```

Compilation completed successfully in 5 s 16 ms (41 minutes ago)