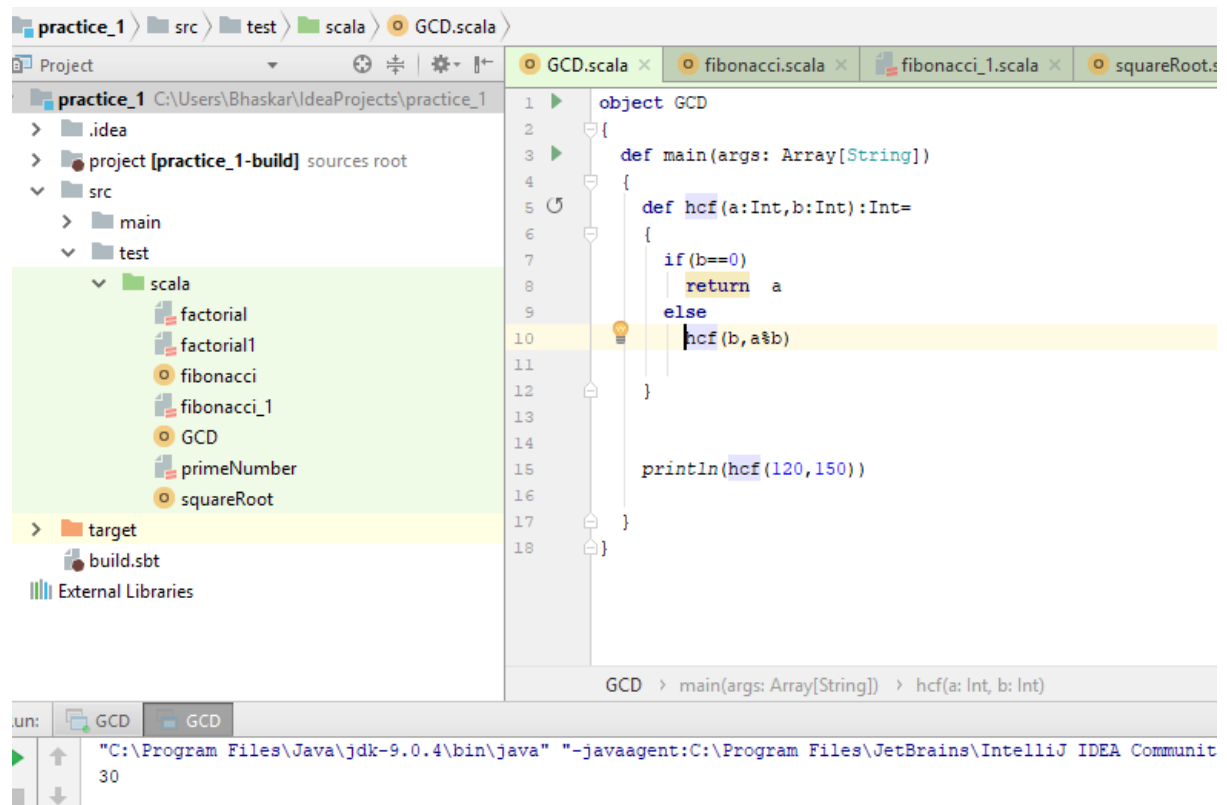


## ASSIGNMENT 14.1

### Task 1

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



```
scala> def hcf(a: Int, b: Int): Int =
|   {
|     if (b == 0)
|       return a
|     else hcf(b, a % b)
|   }
hcf: (a: Int, b: Int) Int

scala> hcf(24, 36)
res0: Int = 12

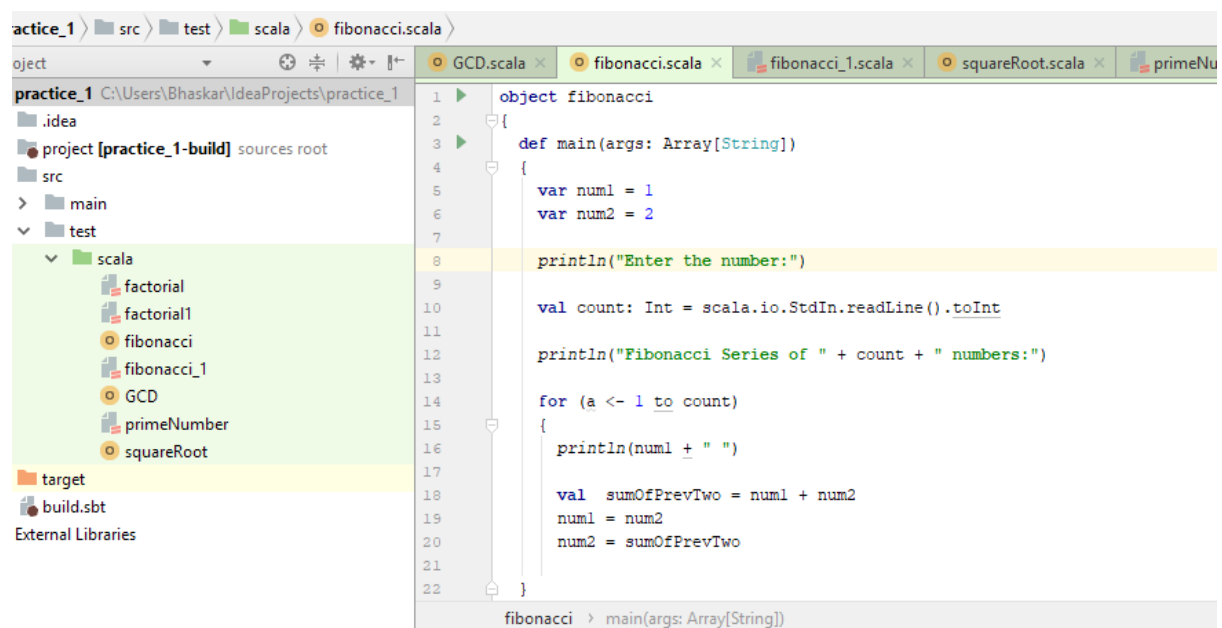
scala> hcf(27, 63)
res1: Int = 9
```

## 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.

1. Write the function using standard for loop



```
object fibonacci
{
  def main(args: Array[String])
  {
    var num1 = 1
    var num2 = 2

    println("Enter the number:")

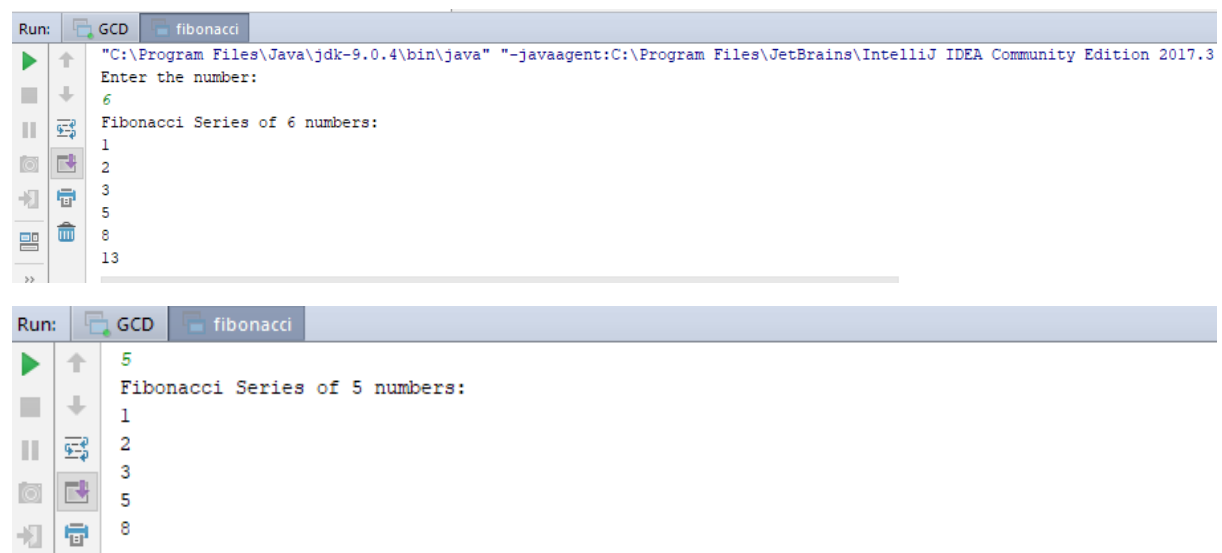
    val count: Int = scala.io.StdIn.readLine().toInt

    println("Fibonacci Series of " + count + " numbers:")

    for (a <- 1 to count)
    {
      println(num1 + " ")

      val sumOfPrevTwo = num1 + num2
      num1 = num2
      num2 = sumOfPrevTwo
    }
  }
}
```

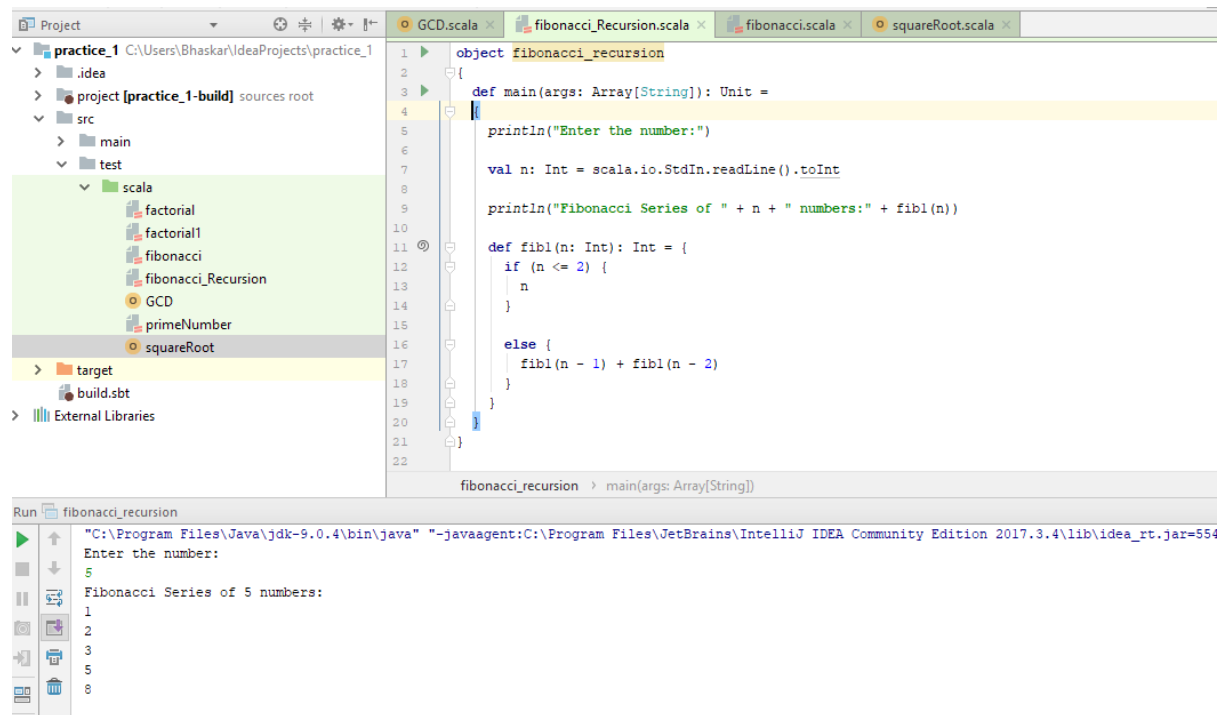
Sample output:



```
Run: GCD fibonacci
"C:\Program Files\Java\jdk-9.0.4\bin\java" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2017.3
Enter the number:
6
Fibonacci Series of 6 numbers:
1
2
3
5
8
13

Run: GCD fibonacci
5
Fibonacci Series of 5 numbers:
1
2
3
5
8
```

## 2. Write the function using recursion



```
scala> def fib1(n: Int): Int = {
|       if (n <= 2) {
|         n
|       }
|       else {
|         fib1(n - 1) + fib1(n - 2)
|       }
|     }
fib1: (n: Int)Int

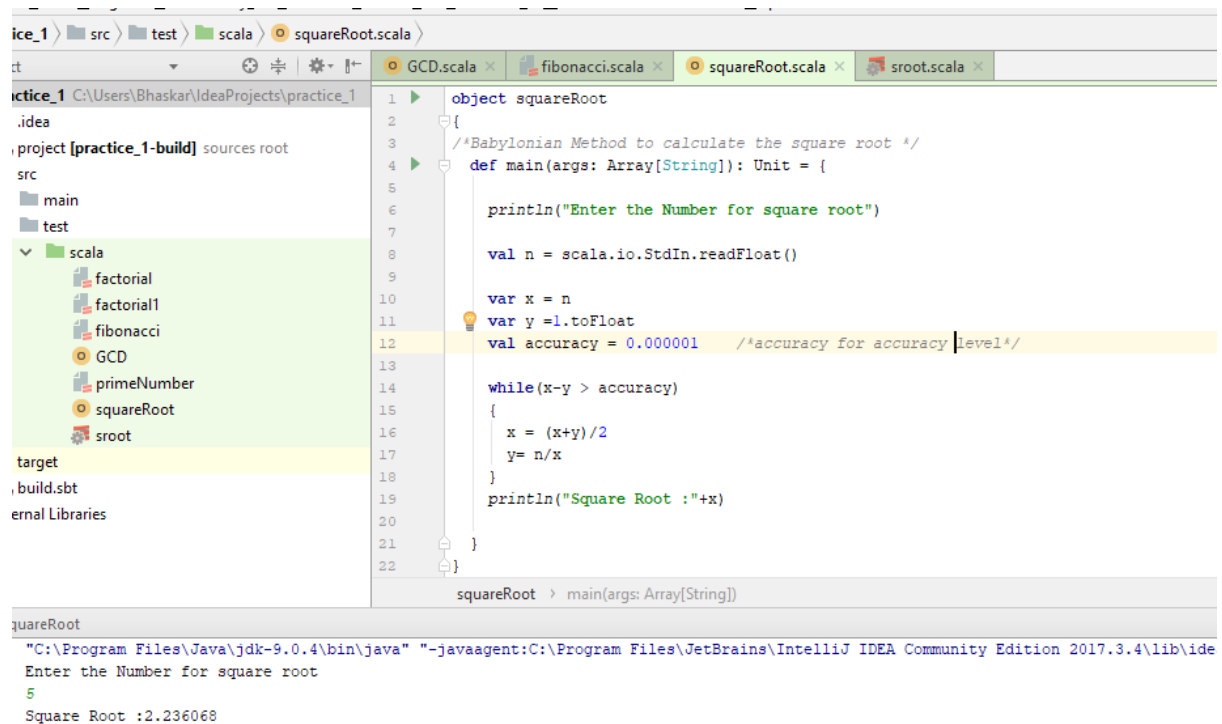
scala> println(fib1(6))
13

scala> println(fib1(5))
8
```

### 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  $y = n/x$



```
1  object squareRoot
2  {
3      /*Babylonian Method to calculate the square root */
4      def main(args: Array[String]): Unit = {
5
6          println("Enter the Number for square root")
7
8          val n = scala.io.StdIn.readFloat()
9
10         var x = n
11         var y = 1.toFloat
12         val accuracy = 0.000001 /*accuracy for accuracy level*/
13
14         while(x-y > accuracy)
15         {
16             x = (x+y)/2
17             y = n/x
18         }
19         println("Square Root :"+x)
20
21     }
22 }
```

squareRoot > main(args: Array[String])

```
"C:\Program Files\Java\jdk-9.0.4\bin\java" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2017.3.4\lib\ide
Enter the Number for square root
5
Square Root :2.236068
```