Session 15

SCALA BASICS 2

Assignment 1

PROBLEM STATEMENT –

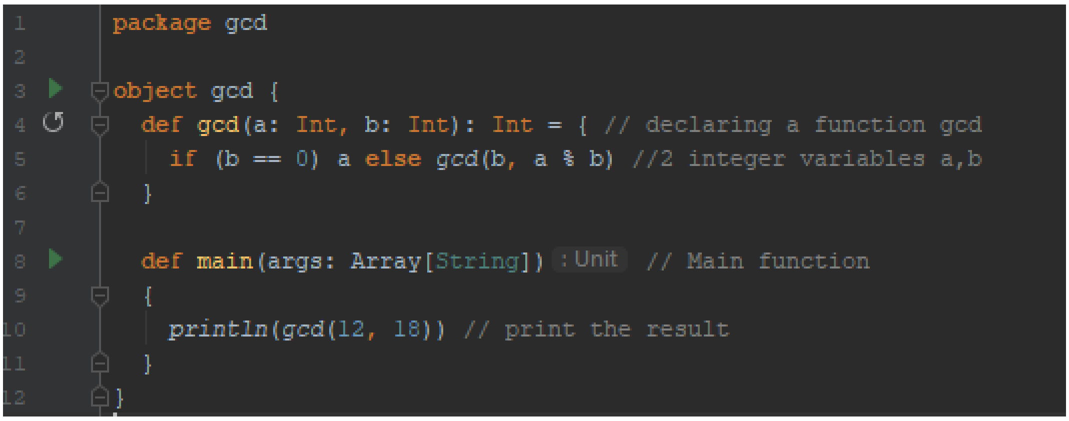
Task 1 -

Create a Scala application to find the GCD of two numbers

SOLUTION –

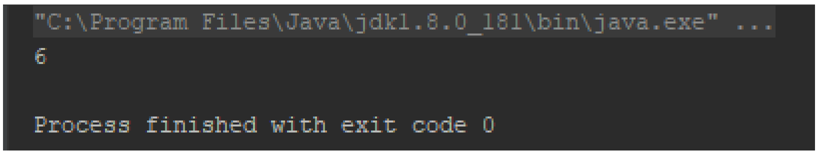
In the below scala code, we are going to find the gcd of the two numbers 12 and 18.

```
def gcd(a: Int, b: Int): Int = { // declaring a function gcd
  if (b == 0) a else gcd(b, a % b) //2 integer variables a,b
}

def main(args: Array[String]) // Main function
{
  println(gcd(12, 18)) // print the result
}
```



```
package gcd

object gcd {
  def gcd(a: Int, b: Int): Int = { // declaring a function gcd
    if (b == 0) a else gcd(b, a % b) //2 integer variables a,b
  }

  def main(args: Array[String]) : Unit // Main function
  {
    println(gcd(12, 18)) // print the result
  }
}
```

OUTPUT –



```
"C:\Program Files\Java\jdk1.8.0_181\bin\java.exe" ...
6

Process finished with exit code 0
```

Task 2 –

Fibonacci series (starting from 1) written in order without any spaces in between, thus

producing a sequence of digits.

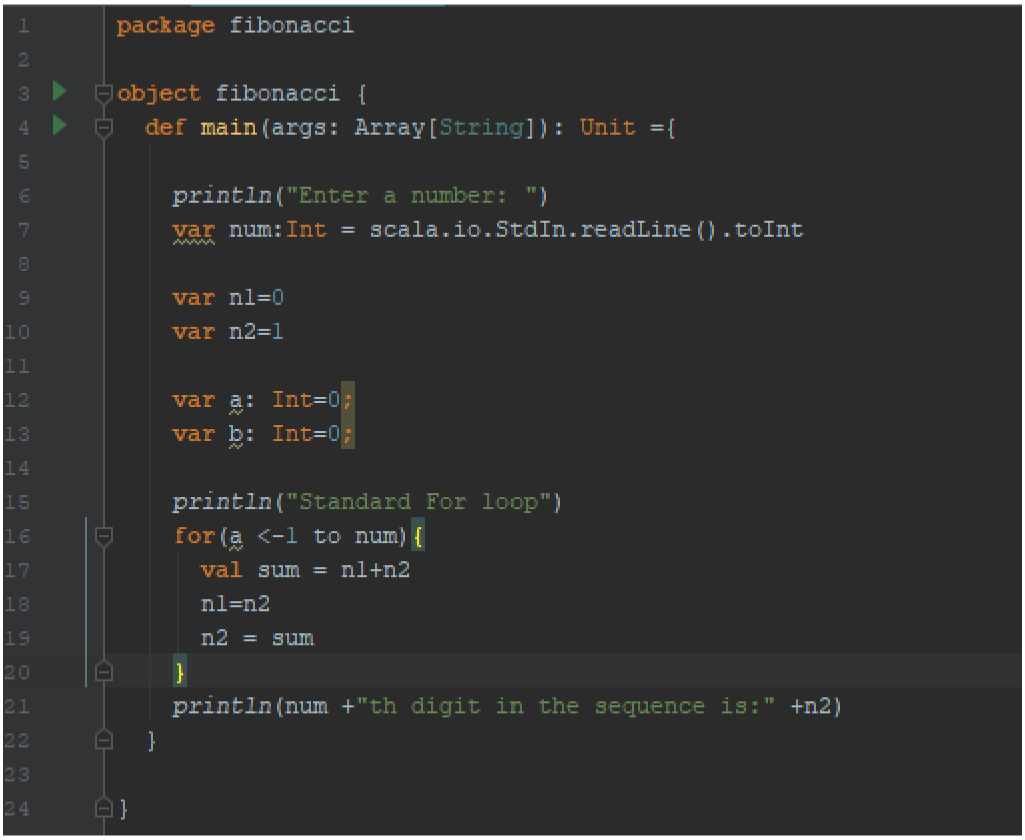
Write a Scala application to find the Nth digit in the sequence.

1. Write the function using standard for loop
2. Write the function using recursion

SOLUTION –

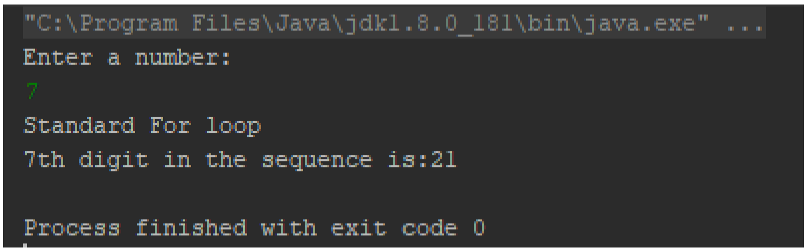
Scala code –

```
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 sum = n1+n2
    n1=n2
    n2 = sum
  }
  println(num +"th digit in the sequence is:" +n2)
}
```



```scala
package fibonacci

object fibonacci {
  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 sum = n1+n2
      n1=n2
      n2 = sum
    }
    println(num +"th digit in the sequence is:" +n2)
  }

}
```

OUTPUT –



```
"C:\Program Files\Java\jdk1.8.0_181\bin\java.exe" ...
Enter a number:
7
Standard For loop
7th digit in the sequence is:21

Process finished with exit code 0
```

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.

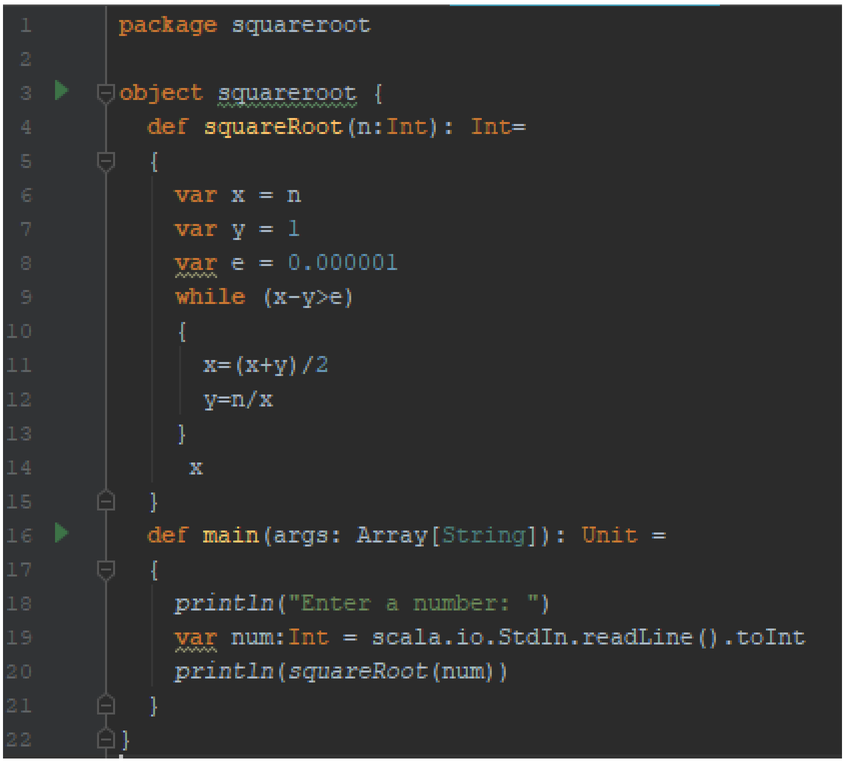
a) Get the next approximation for root using average of x and y

b) Set y = n/x

SOLUTION –

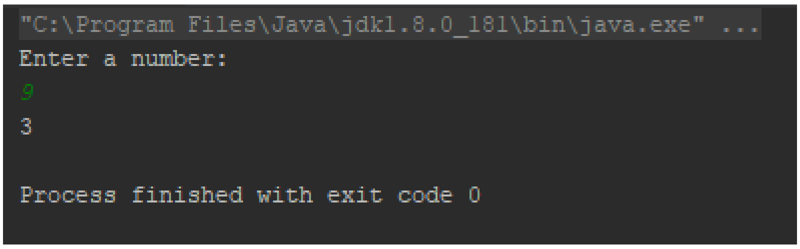
Scala code –

```
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
  }
   x
}
def main(args: Array[String]): Unit =
{
  println("Enter a number: ")
  var num:Int = scala.io.StdIn.readLine().toInt
  println(squareRoot(num))
}
```



```
package squareroot

object squareroot {
  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
    }
     x
  }
  def main(args: Array[String]): Unit =
  {
    println("Enter a number: ")
    var num:Int = scala.io.StdIn.readLine().toInt
    println(squareRoot(num))
  }
}
```

OUTPUT –



```
"C:\Program Files\Java\jdk1.8.0_181\bin\java.exe" ...
Enter a number:
9
3

Process finished with exit code 0
```