ASSIGNMENT13.2

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

Fib.scala

```
 object FIB{

  def fibseq(n: Int):Seq[Int] = {
       var num1 = 0
       var num2 = 1
       for( i <- 1 to n) yield {
         var num3 = num1 + num2
         num1 = num2
         num2 = num3
         num1

       }

  }
  def main(args:Array[String])
  {
      println(fibseq(9))
  }
 }
```

Output

```
Vector(1, 1, 2, 3, 5, 8, 13, 21, 34)
```

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

Write the function using standard for loop
Digit.scala

```
object Digit extends App {
  val myseq = Seq(1,2,3,4,5,6,7,8,9,10)
      for (j <- 0 to myseq.length - 1)
      {
         println(myseq(j))
      }
}
```

Output

```
1
2
3
4
5
6
7
8
9
10
```

Write the function using recursion

```
object Digit extends App {
  val myseq = Seq(1,2,3,4,5,6,7,8,9,10)


def nthRecursive[A](n: Int, ls: Seq[A]): A = (n, ls) match {
        case (0, h :: _    ) => h
        case (n, _ :: tail) => nthRecursive(n - 1, tail)
        case (_, Nil       ) => throw new NoSuchElementException
  }
      println(nthRecursive(6,myseq))
}
```

Output

```
7
```