Assignment 15

Task 1

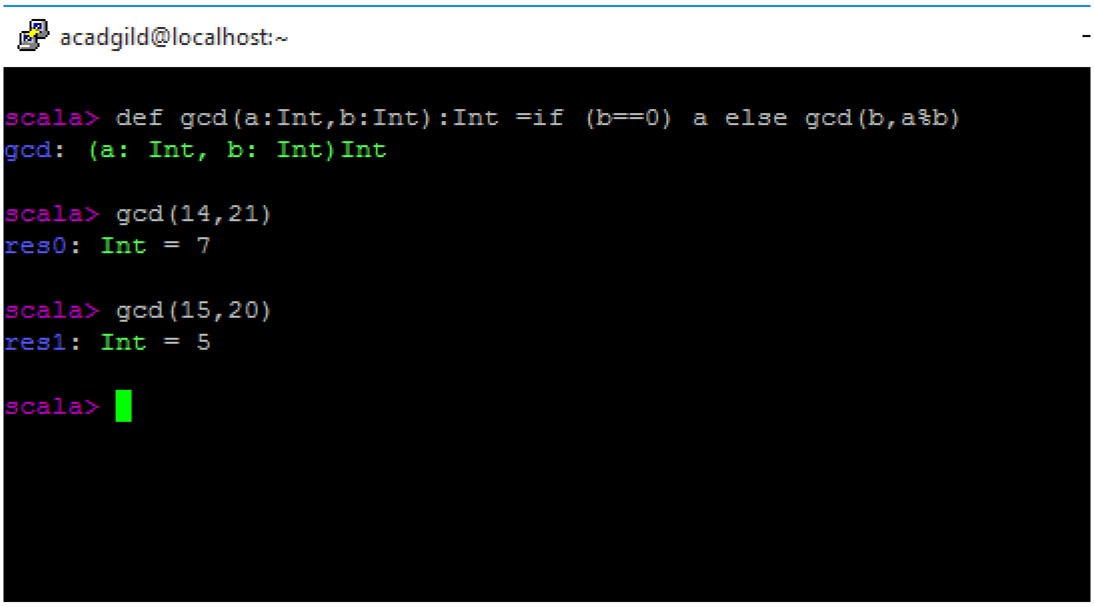
Create a Scala application to find the GCD of two numbers

Scala Code :

def gcd(a:Int , b:Int) :Int = if(b==0) a else gcd(b,a%b)

gcd(14,21)

gcd(15,20)



```
scala> def gcd(a:Int,b:Int):Int =if (b==0) a else gcd(b,a%b)
gcd: (a: Int, b: Int)Int

scala> gcd(14,21)
res0: Int = 7

scala> gcd(15,20)
res1: Int = 5

scala>
```

Task 2

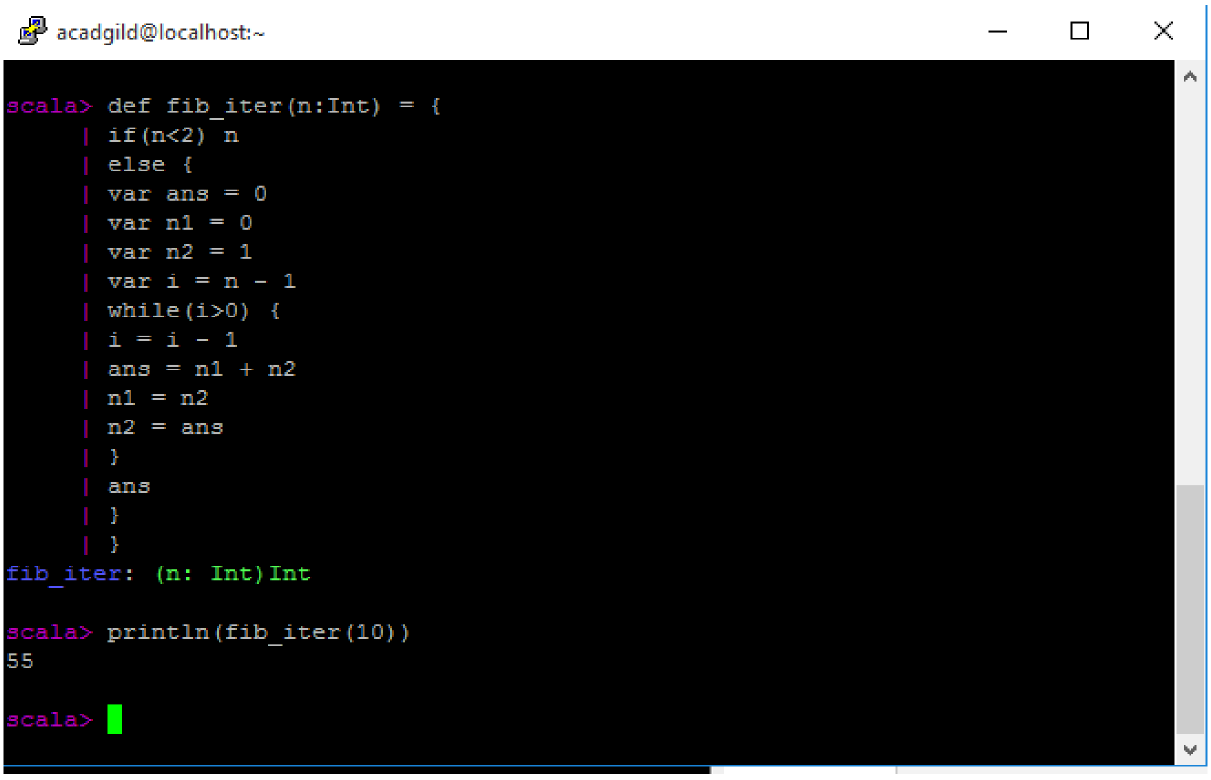
Fibonacci series (starting from 1) written in order without any spaces in between, thus producing a sequence of digits.

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

1. Write the function using standard for loop
   Scala Code :

```
def fib_iter(n:Int) = {
if(n<2) n
else {
var ans = 0
var n1 = 0
var n2 = 0
var i = n - 1
while(i>0) {
i = i - 1
ans = n1 + n2
n1 = n2
n2 = ans
}
ans
}
}

println(fib_iter(10))
```


acadgild@localhost:~

```
scala> def fib_iter(n:Int) = {
     | if(n<2) n
     | else {
     | var ans = 0
     | var n1 = 0
     | var n2 = 1
     | var i = n - 1
     | while(i>0) {
     | i = i - 1
     | ans = n1 + n2
     | n1 = n2
     | n2 = ans
     | }
     | ans
     | }
     | }
fib_iter: (n: Int)Int

scala> println(fib_iter(10))
55

scala>
```

2 . Write the function using recursion

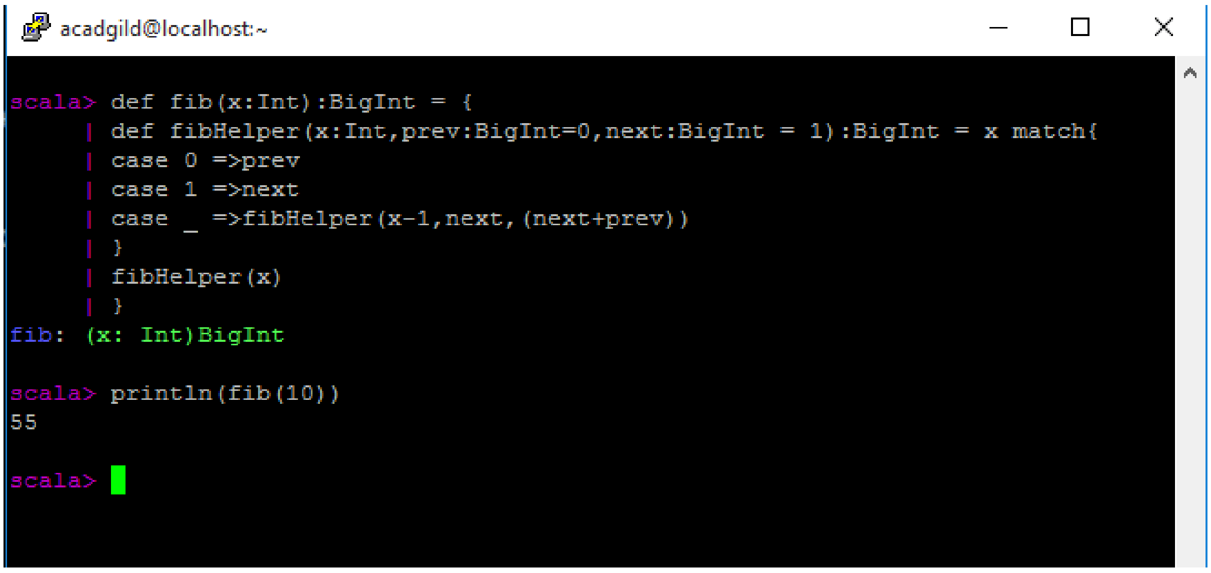
Scala Code :

```
def fib(x:Int):BigInt = {

def fibHelper(x:Int,prev:BigInt=0,next:BigInt = 1):BigInt = x match {

case 0 => prev
case 1 => next
case_ =>fibHelper(x-1,next,(next+prev))
}
fibHelper(x)
}

println(fib(10))
```



```
scala> def fib(x:Int):BigInt = {
     | def fibHelper(x:Int,prev:BigInt=0,next:BigInt = 1):BigInt = x match{
     | case 0 =>prev
     | case 1 =>next
     | case _ =>fibHelper(x-1,next,(next+prev))
     | }
     | fibHelper(x)
     | }
fib: (x: Int)BigInt

scala> println(fib(10))
55

scala>
```

__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

__Scala Code :__

```
def sqrt(a:Double) = {

val acc = 1e-10

def findRoot(x:Double):Double = {
```

val nextx = (a/x + x)/2

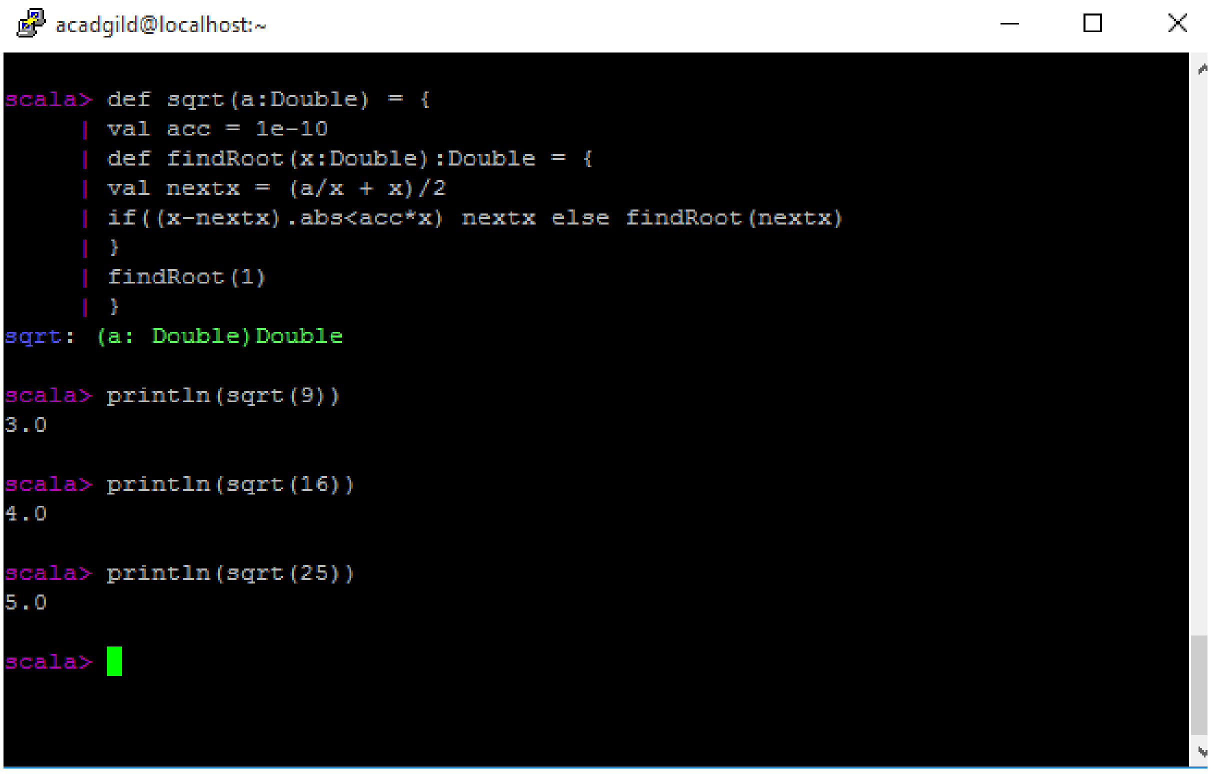
if((x-nextx).abs<acc*x) nextx else findRoot(nextx)

}

findRoot(1)

}

println(sqrt(9))



acadgild@localhost:~

```
scala> def sqrt(a:Double) = {
     | val acc = 1e-10
     | def findRoot(x:Double):Double = {
     | val nextx = (a/x + x)/2
     | if((x-nextx).abs<acc*x) nextx else findRoot(nextx)
     | }
     | findRoot(1)
     | }
sqrt: (a: Double)Double

scala> println(sqrt(9))
3.0

scala> println(sqrt(16))
4.0

scala> println(sqrt(25))
5.0

scala>
```