Task 1

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

Code:

```
package scala2

object Get_gcd {


  def getgc(a :Int,b:Int): Int ={
    if(b == 0 || a ==0)   a else  getgc(b,a%b)

  }
  def main(args:Array[String]): Unit ={
    println("Hello India !!!!")
    println(getgc(2,5))
  }

}
```

Output:

```
/usr/java/jdk1.8.0_151/bin/java ...
Hello India !!!!
1

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 the function using standard for loop

```
object Get_fact {

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

    def fib(n: Int): Int = {
      var a = 0
      var b = 1
      var g=0
      var f =0
      for (g<- 2 to n ) {
        val c = a + b
        f=c
        a = b
        b = c
        print(a+",")

      }
      return f
    }
      println("\t\tthe value is " +fib(9))
  }
```

Output

```
/usr/java/jdk1.8.0_151/bin/java ...
1,1,2,3,5,8,13,21,      the value is 34

Process finished with exit code 0
```

2. Write the function using recursion

Code:

```
object fib_recursion {
def main(args:Array[String]): Unit ={
  def getfib(n :Int):Int={
    if(n<=1)
      return n
    return getfib(n-1)+getfib(n-2)


  }
  println("The value of the "+ getfib(9))
}
}
```

Output:

```
/usr/java/jdk1.8.0_151/bin/java ...
The value of the 34

Process finished with exit code 0
```

```
object babylonian {
  def main(args: Array[String]): Unit = {
   def babylon(n :Int):Int={

     var sqroot = math.sqrt(n)
     if (sqroot % 1 == 0)
       sqroot.toInt
     else
       ((sqroot + (n / sqroot)) / 2).toInt
   }
    println("it is babylonian method with integer is \t\t" + babylon(93))

  }
  def babylondocuble(n :Int):Double={

    var sqroot = math.sqrt(n)
    if (sqroot % 1 == 0)
      sqroot
    else
      ((sqroot + (n / sqroot)) / 2)
  }
  println("it is babylonian method with double return  \t\t" + babylondocuble(93))


}
```

Output is with different return types int,double and the input is 93

```
/usr/java/jdk1.8.0_151/bin/java ...
it is babylonian method with double return          9.643650760992955
it is babylonian method with integer is         9

Process finished with exit code 0
```