

Task 1 : GCD of two numbers

```
object gcd {  
  def gcd(a: Int,b: Int): Int = {  
    if(b ==0) a else gcd(b, a%b)  
  }  
  
  def main(args: Array[String]) {  
    println(gcd(25,15))  
  }  
}
```

```
object gcd {  
  
  def gcd(a: Int,b: Int): Int = {  
    if(b ==0) a else gcd(b, a%b)  
  }  
  
  def main(args: Array[String]) {  
    println(gcd(25,15))  
  }  
}
```

Problems Tasks Console

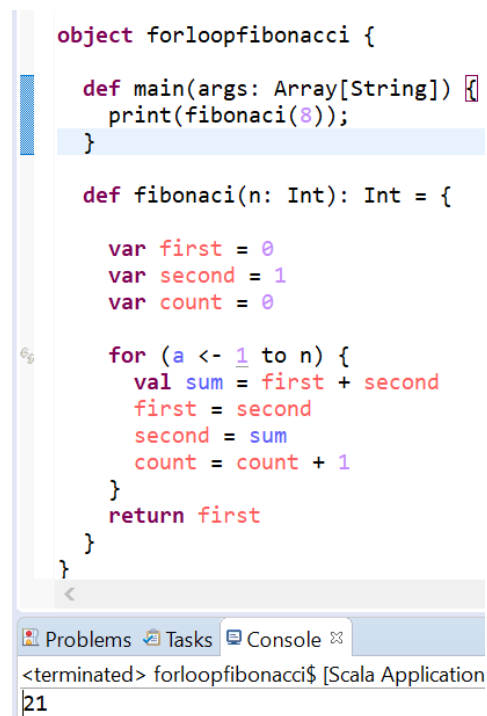
<terminated> gcd\$ [Scala Application] C:\Program File

5

Task 2 : Fibonacci

a)Using For loop

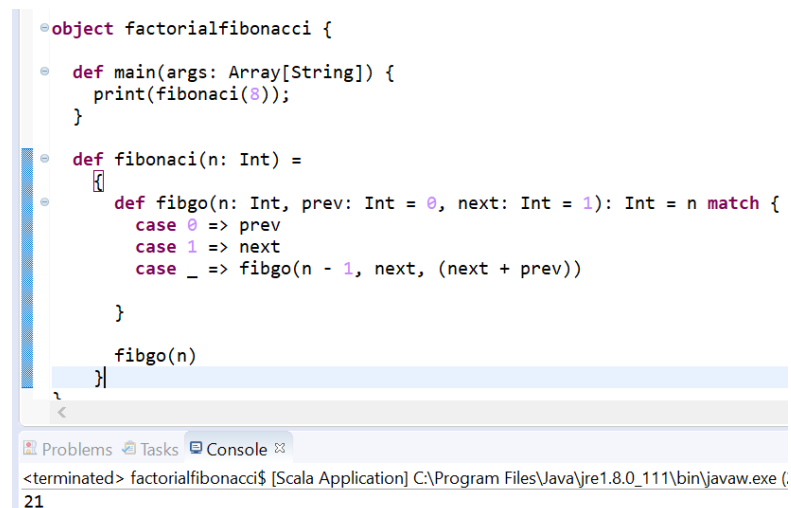
```
object forloopfibonacci {  
  
  def main(args: Array[String]) {  
    print(fibonacci(8));  
  }  
  
  def fibonacci(n: Int): Int = {  
  
    var first = 0  
    var second = 1  
    var count = 0  
  
    for (a <- 1 to n) {  
      val sum = first + second  
      first = second  
      second = sum  
      count = count + 1  
    }  
    return first  
  }  
}
```



The screenshot shows an IDE with a Scala file named `forloopfibonacci`. The code is identical to the one above. The `main` method is highlighted in blue. Below the code editor, there is a `Problems` tab, a `Tasks` tab, and a `Console` tab. The `Console` tab is active, showing the output: `<terminated> forloopfibonacci$ [Scala Application]`. The line number `21` is visible in the left margin.

b)Using recursion

```
object factorialfibonacci {  
  
  def main(args: Array[String]) {  
    print(fibonacci(8));  
  }  
  
  def fibonacci(n: Int) =  
  {  
    def fibgo(n: Int, prev: Int = 0, next: Int = 1): Int = n match {  
      case 0 => prev  
      case 1 => next  
      case _ => fibgo(n - 1, next, (next + prev))  
    }  
  
    fibgo(n)  
  }  
}
```



Task 3: Find Square root using Babylonian method

```
object squareroot {  
  def main(args: Array[String]) {  
    print(findsquareroot(12));  
  }  
  
  def findsquareroot(n: Int): Int = {  
    var x = n  
    var y = 1  
  
    while (x - y > 0) {  
      x = (x + y) / 2;  
      y = n / x;  
    }  
    return x;  
  }  
}
```

```
object squareroot {  
  def main(args: Array[String]) {  
    print(findsquareroot(144));  
  }  
  def findsquareroot(n: Int): Int = {  
    var x = n  
    var y = 1  
  
    while (x - y > 0) {  
      x = (x + y) / 2;  
      y = n / x;  
    }  
    return x;  
  }  
}
```

Problems Tasks Console

<terminated> squareroot\$ [Scala Application] C:\Progran

12