

Assignment 15

Task1

Create a Scala application to find the GCD of two numbers

```
def gcd(a: Int,b: Int): Int = {  
    if(b ==0) a else gcd(b, a%b)    //if one of the numbers is 0 then GCD is  
the other number else recursive approach is followed to find the GCD of two  
numbers  
}  
val result = gcd (25,15)    //I/Ps: 25 and 15  
println(result);
```

O/P: 5

Task2

Write a Scala application to find the Nth digit in the Fibonacci sequence (starting from 1)

a. Write the function using standard for loop

```
var a =1;    //initialising the first number in the series  
var b=2;    //initialising the second number in the series  
for (i<-1 until 10)    //iterating the loop for 10 times  
{  
    var c= a+b;    //initialising a variable to store the sum of  
previous two numbers  
    a=b;    //assigning second value to the first value  
    b=c;    //assigning third value to the second value  
}  
println(a);
```

O/P: 89

b. Write the function using recursion

```
def fib2( n : Int) : Int = n match {    //defining a method, fib2, for the  
calculation of nth value of the series  
    case 0 | 1=>n    //if the number of terms in the series is 0, then  
return 1
```

```

case 1 | 2 => n //if the number of terms in the series is
1, then return 2
case _ => fib2( n-1 ) + fib2( n-2 ) //else if it is any other number then
recursion is done with the previous number and the previous to previous
number
}
val res=fib2(10); // passing the parameter as 10, therefore,
there are 10 terms in the series
println(res);

```

O/P: 89

Task3

Find square root of number using Babylonian method

```

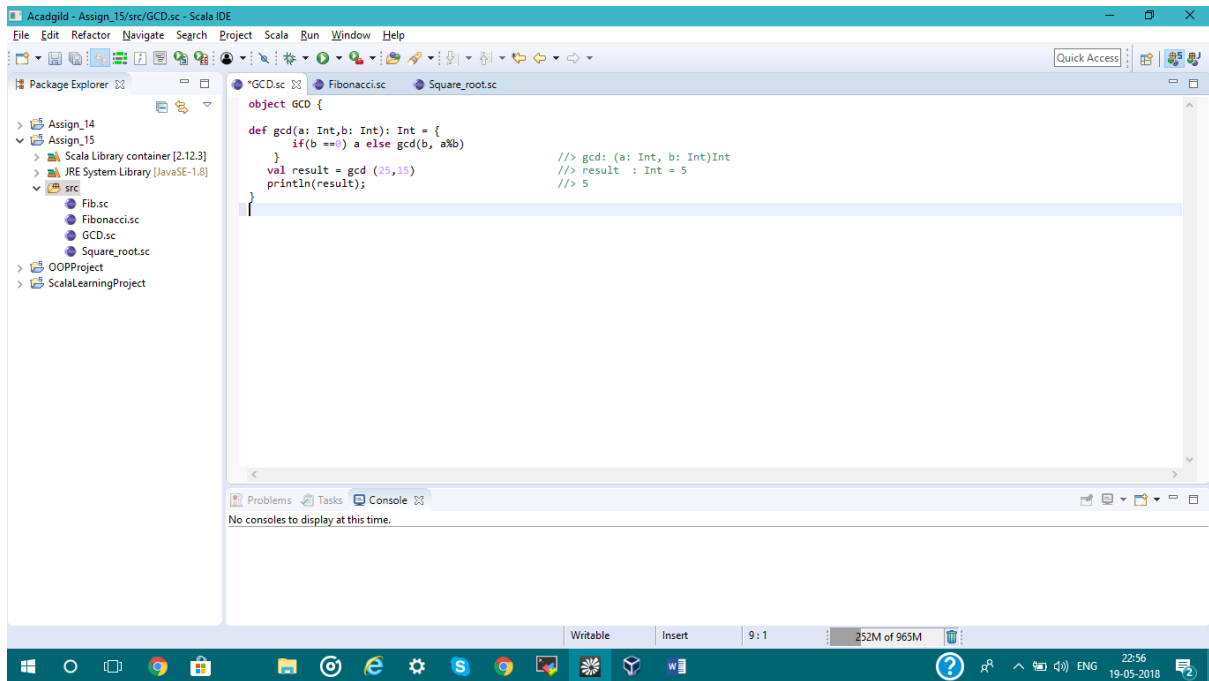
def sq(n : Int) : Float = { //defining a method 'sq' to find the square root
var y =1; //initialising a variable y
var x = 5; //initialising a variable x, with the value, closer
to the root
var e = 0.000001; //e decides the accuracy level
while(x-y>e) //declaring a loop
{
x=(x+y)/2; // Get the next approximation for root using
average of x and y
y=n/x; //set y=n/x
}
return x;
}
val result = println(sq(9)); //passing 9 as an the argument to find the
square root

```

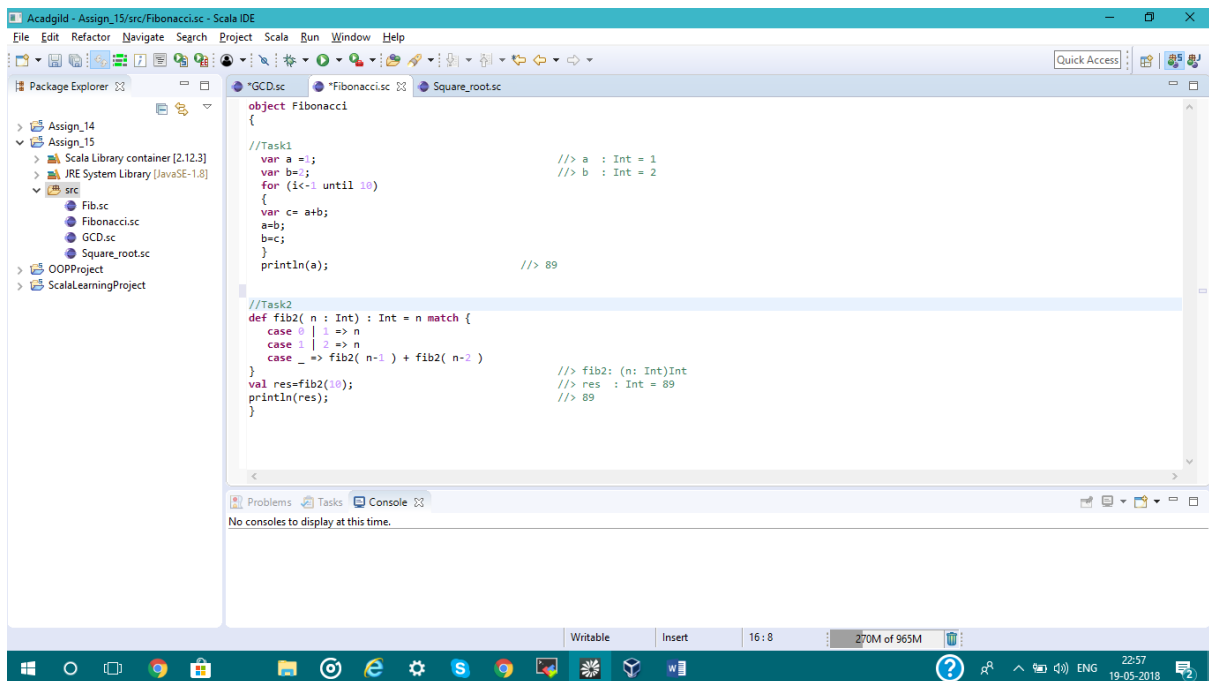
O/P: 3

Screenshots

Task1



Task2



Task3

