**ASSIGNMENT-13.3**

1. Find square root of number using Babylonian method in Scala:

**PROGRAM:**

object Assignment\_13\_3\_Babylonian {

def squareRoot(n:Float):Float={

var x= n

var y = 1.0f

val e = 0.000001

while((x-y) > e) {

x = (x + y) / 2

y = n / x

}

x

}

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

println("Square root of 50 in Babylonian method: " +squareRoot(50))

}

}

**EXPLANATION:**

* Function is defined in scala using **def** keyword.
* First Method **squareRoot** takes one argument as parameter, that returns a Float value.
* In the method **squareRoot**, the logic to calculate the squareRoot of a number passed via **main** method is written.

**LOGIC:** var x= n

var y = 1.0f

val e = 0.000001

while((x-y) > e) {

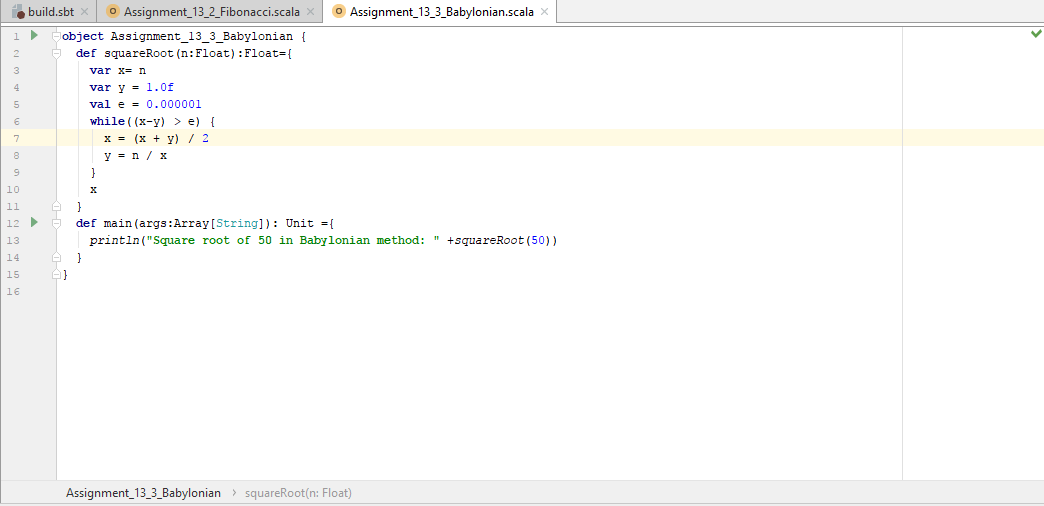
x = (x + y) / 2

y = n / x

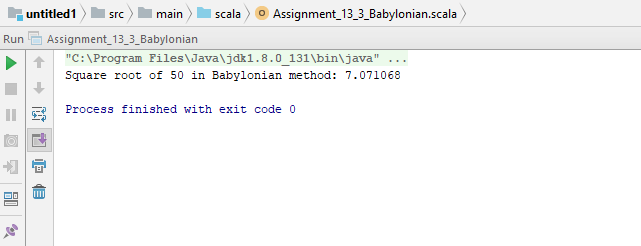
}

* Second method is **main**, which takes Array of string as arguments.
* Number for which square Root to be calculated is passed via main method and the method **squareRoot** is invoked with the integer as parameter.
* The parameter passed here is : 50

**PROGRAM SCREENSHOT:**

****

**RESULT:**

****

*File below is attached for reference:*

**