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

object SquareRoot\_using\_BabyloinanMethod{

//funtion to return the square root of a number using Babylonian Method

def squareRoot (num:Int):Float ={

var x:Float =num

var y:Float =1

var e:Double =0.000001

while (x -y > e) {

x=(x+y)/2

y=num/x

}

x //returing the square root answer

}

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

var wish =""

do {

println("Find the squar root of number\n")

println("-----------------------------\n")

println("Enter the number")

var input =readLine.toInt

println("-----------------------------")

println(s"Squar root of $input is ${squareRoot(input)}")

println("-----------------------------\n")

println("\nDo you wish to continue ? (Y/N):")

wish =readLine.toUpperCase

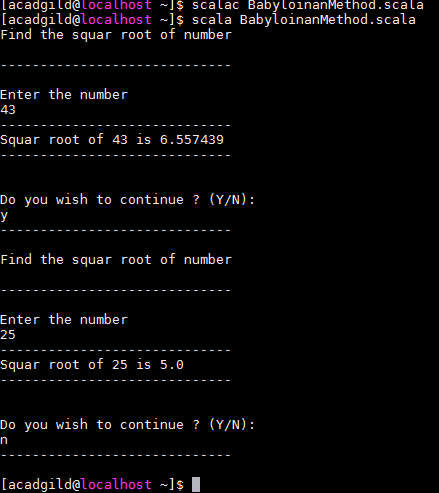
println("-----------------------------\n")

}while(wish.equals("Y"))

}

}

**OUTPUT**

****