Scala Assignment 13.3

For this assignment, I've used Intellij Idea.

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.

   - Get

   - Set y = n/x

Steps followed:

1) Created a method squareroot, which takes a Float value as input as returns a Float value as output.

```
def squareRoot(n: Float):Float= {
}
```

2) Then initialized the x with value n(in the problem statement, its asked to take something closer to n.

```
  var x = n           //We are using n itself as initial approximation
```

3) Initialized y to 1F.

```
  var y = 1F          //Initializing y as 1
```

4) Then setting an accurancy level and condition, upto which, the steps 5 and 6 should be repeated.

```
  val e = 0.000001F   /* e decides the accuracy level*/
  while ( {/*We are using n itself as initial approximation */
    x - y > e
  })
```

5) Getting the next approximation for root using average of x and y

```
{
    x = (x + y) / 2   //Getting the next approximation for root using average
of x and y, then setting x
```

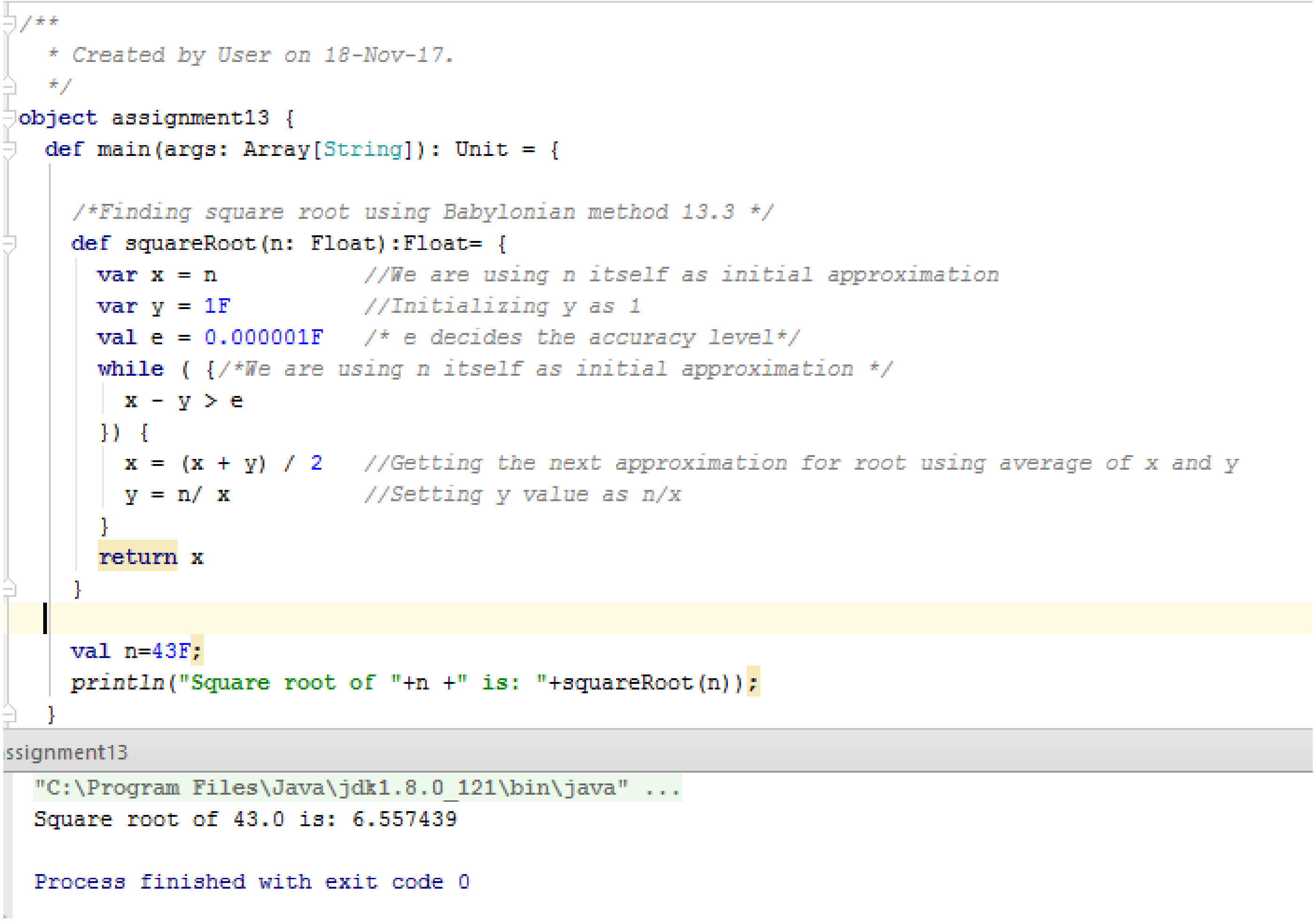
6) Setting y value as y = n/x

```
   y = n/ x          //Setting y value as n/x
}
```

7) Once the loop completes returning the value of x

```
return x
```

Screenshot of IntelliJ and output:



```
/**
  * Created by User on 18-Nov-17.
  */
object assignment13 {
  def main(args: Array[String]): Unit = {

    /*Finding square root using Babylonian method 13.3 */
    def squareRoot(n: Float):Float= {
      var x = n           //We are using n itself as initial approximation
      var y = 1F          //Initializing y as 1
      val e = 0.000001F   /* e decides the accuracy level*/
      while ( {/*We are using n itself as initial approximation */
        x - y > e
      }) {
        x = (x + y) / 2   //Getting the next approximation for root using average of x and y
        y = n/ x          //Setting y value as n/x
      }
      return x
    }

    val n=43F;
    println("Square root of "+n +" is: "+squareRoot(n));
  }
```

```
"C:\Program Files\Java\jdk1.8.0_121\bin\java" ...
Square root of 43.0 is: 6.557439

Process finished with exit code 0
```

Source code:

```
/*Finding square root using Babylonian method 13.3 */
def squareRoot(n: Float):Float= {
  var x = n           //We are using n itself as initial approximation
  var y = 1F          //Initializing y as 1
  val e = 0.000001F   /* e decides the accuracy level*/
  while ( {/*We are using n itself as initial approximation */
    x - y > e
  }) {
    x = (x + y) / 2   //Getting the next approximation for root using average of x and
y
    y = n/ x          //Setting y value as n/x
  }
  return x
}
```