

Find square root of number using Babylonian method.

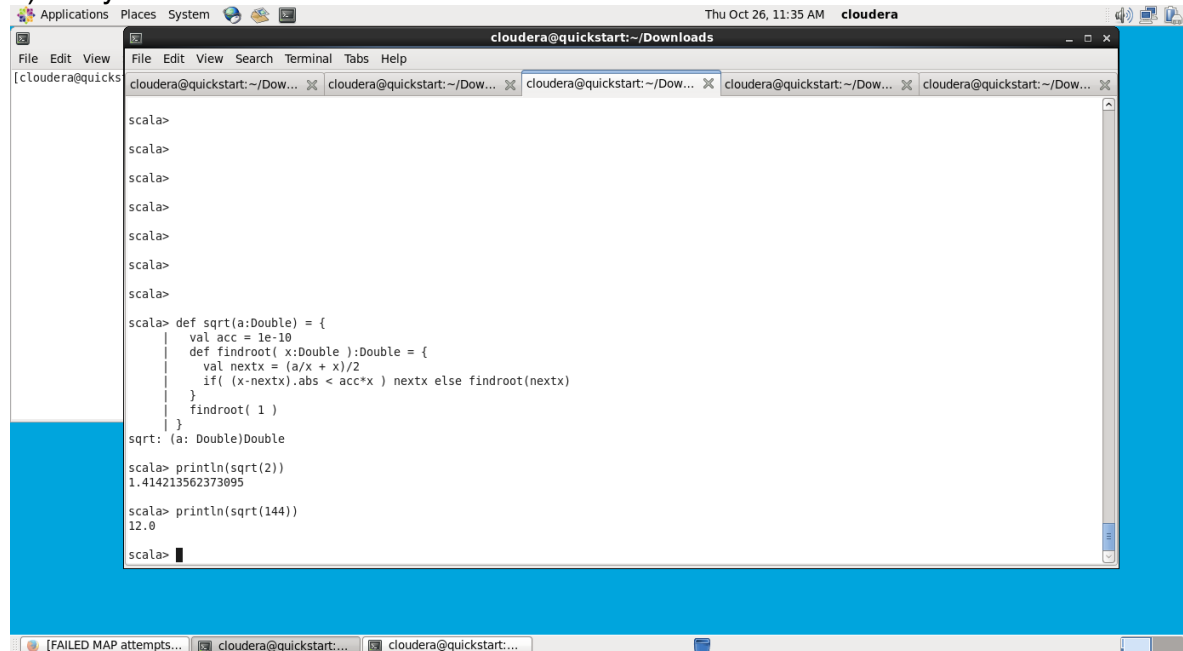
1 Start with an arbitrary positive start value x (the closer to the root, the better).

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 $y = n/x$



```
cloudera@quickstart:~/Downloads
File Edit View Search Terminal Tabs Help
cloudera@quickstart:~/Dow... cloudera@quickstart:~/Dow... cloudera@quickstart:~/Dow... cloudera@quickstart:~/Dow... cloudera@quickstart:~/Dow...

scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala> def sqrt(a:Double) = {
    |   val acc = 1e-10
    |   def findroot( x:Double ):Double = {
    |       val nextx = (a/x + x)/2
    |       if ( (x-nextx).abs < acc*x ) nextx else findroot(nextx)
    |   }
    |   findroot( 1 )
    | }
sqrt: (a: Double)Double
scala> println(sqrt(2))
1.414213562373095
scala> println(sqrt(144))
12.0
scala>
```