

1. Find an approximation to the square root of 13, correct to within ten to the -4 power, using the Bisection Algorithm.

```
% bisection(function f, a, b, tol, nMax)
>> bisection(@(x) x.^2 - 13, 1, 5, 10.^-4, 30)
iteration 1
c = 3
err = 2
a = 3

iteration 2
c = 4
err = 1
b = 4

iteration 3
c = 3.5000
err = 0.50000
a = 3.5000

iteration 4
c = 3.7500
err = 0.25000
b = 3.7500

iteration 5
c = 3.6250
err = 0.12500
b = 3.6250

iteration 6
c = 3.5625
err = 0.062500
a = 3.5625

iteration 7
```

```
c = 3.5938
err = 0.031250
a = 3.5938
```

```
iteration 8
c = 3.6094
err = 0.015625
b = 3.6094
```

```
iteration 9
c = 3.6016
err = 0.0078125
a = 3.6016
```

```
iteration 10
c = 3.6055
err = 0.0039062
a = 3.6055
```

```
iteration 11
c = 3.6074
err = 0.0019531
b = 3.6074
```

```
iteration 12
c = 3.6064
err = 9.7656e-04
b = 3.6064
```

```
iteration 13
c = 3.6060
err = 4.8828e-04
b = 3.6060
```

```
iteration 14
c = 3.6057
err = 2.4414e-04
b = 3.6057
```

```
iteration 15
c = 3.6056
err = 1.2207e-04
b = 3.6056
```

```
iteration 16
c = 3.6055
err = 6.1035e-05
ans = 3.6055
```

The answer is 3.6055

Verification by direct calculation:

```
>> sqrt(13)
```

```
ans = 3.605551275
```

So the result of the bisection algorithm agrees with the “calculator” to the specified precision.

2. Find an approximation to the cube root of 36, correct to within ten to the -4 power, using the Bisection Algorithm.

```
% bisection(function f, a, b, tol, nMax)
>> bisection(@(x) x.^3 - 36, 1, 5, 10.^-4, 30)
iteration 1
c = 3
err = 2
a = 3

iteration 2
c = 4
err = 1
b = 4

iteration 3
c = 3.5000
err = 0.50000
b = 3.5000

iteration 4
c = 3.2500
err = 0.25000
a = 3.2500

iteration 5
```

```
c = 3.3750
err = 0.12500
b = 3.3750
```

```
iteration 6
c = 3.3125
err = 0.062500
b = 3.3125
```

```
iteration 7
c = 3.2812
err = 0.031250
a = 3.2812
```

```
iteration 8
c = 3.2969
err = 0.015625
a = 3.2969
```

```
iteration 9
c = 3.3047
err = 0.0078125
b = 3.3047
```

```
iteration 10
c = 3.3008
err = 0.0039062
a = 3.3008
```

```
iteration 11
c = 3.3027
err = 0.0019531
b = 3.3027
```

```
iteration 12
c = 3.3018
err = 9.7656e-04
a = 3.3018
```

```
iteration 13
c = 3.3022
err = 4.8828e-04
b = 3.3022
```

```
iteration 14
c = 3.3020
err = 2.4414e-04
b = 3.3020

iteration 15
c = 3.3019
err = 1.2207e-04
a = 3.3019

iteration 16
c = 3.3019
err = 6.1035e-05
ans = 3.3019
```

The answer is 3.3019

```
>> cbrt(36)
```

```
ans = 3.30192725
```

So the result of the bisection algorithm agrees with the “calculator” to the specified precision.