Chapter7_Exercises

May 29, 2017

1 Chapter 7 Exercises

1.1 Exercise 7.1

Copy the loop from Section 7.5 and encapsulate it in a function called mysqrt that takes a as a parameter, chooses a reasonable value of x, and returns an estimate of the square root of a. To test it, write a function named test_square_root that prints a table

The first column is a number, a; the second column is the square root of a computed with mysqrt; the third column is the square root computed by math.sqrt; the fourth column is the absolute value of the difference between the two estimates.

In [19]:
```
import math

epsilon = 0.000001

def mysqrt(a):
    x = math.log2(a)

    if x == 0:
        x = 1

    while True:
        y = (x + a/x) / 2
        if abs(y-x) < epsilon:
            break
        x = y

    return y

def test_square_root():
    print('a\tmysqrt(a)\tmath.sqrt(a)\tdiff')
    print('-\t---------\t------------\t----')
    for i in range(1, 10):
        print('{:.1f}\t{:.4f}\t\t{:.4f}\t\t{:.4f}'.format(i, mysqrt(i), ma
    print()

test_square_root()
```

```
a         mysqrt(a)         math.sqrt(a)         diff
-         ---------         ------------         ----
1.0         1.0000                1.0000               0.0000
2.0         1.4142                1.4142               0.0000
3.0         1.7321                1.7321               0.0000
4.0         2.0000                2.0000               0.0000
5.0         2.2361                2.2361               0.0000
6.0         2.4495                2.4495               0.0000
7.0         2.6458                2.6458               0.0000
8.0         2.8284                2.8284               0.0000
9.0         3.0000                3.0000               0.0000
```

1.2 Exercise 7.2

The built-in function eval takes a string and evaluates it using the Python interpreter.

Write a function called eval_loop that iteratively prompts the user, takes the resulting input and evaluates it using eval, and prints the result.

It should continue until the user enters 'done', and then return the value of the last expression it evaluated.

```
In [28]: def eval_loop():
             temp = None
             while True:
                 userInput = input('Enter your prompt or type "done" to finish:')

                 if userInput == 'done':
                     return temp

                 print(eval(userInput))

                 temp = eval(userInput)

         returnVal = eval_loop()

         print('The return val is: {}'.format(returnVal))
```

```
Enter your prompt or type "done" to finish:math.pi
3.141592653589793
Enter your prompt or type "done" to finish:done
The return val is: 3.141592653589793
```

1.3 Exercise 7.3

The mathematician Srinivasa Ramanujan found an infinite series that can be used to generate a numerical approximation of $1/\pi$:

(Excluded here)

Write a function called estimate_pi that uses this formula to compute and return an estimate of π. It should use a while loop to compute terms of the summation until the last term is smaller than 1e-15 (which is Python notation for 10^{-15}). You can check the result by comparing it to math.pi.

```
In [30]: def estimate_pi():
             k = 0
             summation = 0
             while True:
                 term = ((math.factorial(4*k))*(1103 + 26390*k))/((math.factorial(k

                 summation += term

                 if term < 1e-15:
                     break
                 else:
                     k += 1

             pi = (summation * ((2*math.sqrt(2))/9801))**-1

             return pi

         print(estimate_pi())
```

```
3.141592653589793
```

Neat