Unit 7

Apply what you know

0. Study the programs in the *Learn something new* section until you can write them yourself from scratch without relying on this document or any other source of information. Here are the programs:

    0.1. Write a program that uses shelved CPI data to produce a neatly formatted table to show, for each year, the total percentage increase in the index over the past one, five and ten years.

    0.2. Write a program that allows the user to use trial and error to find the annual rate of increase necessary over five years to increase a starting index value of 175.1 by a total of 13.2 percent.

    0.3. Modify the previous program so that the trial-and-error process is automated. Have the program make 20 guesses, each at the midpoint of the current possible range. The initial range should be from -100 percent to 100 percent.

    0.4. Modify the previous program so that the trial-and-error process is conducted within a function and so that it continues until the error has been reduced below a specified maximum. Pass in the error limit as an argument, along with the limits of the initial range and the target value.

    0.5. Modify the previous program so that the trial-and-error function takes an additional argument specifying the function to be explored. Test by finding the annual rate of increase necessary over ten years to increase a starting index value of 154.4 by a total of 28.4 percent, in addition to solving the five-year problem.

    0.6. Modify the previous program so that functions like the two it explores by trial and error can be produced automatically.

    0.7. Write a program that uses shelved CPI data to produce a neatly formatted table to show, for each year, the *annualized* increase in the index over the past one, five and ten years. By annualized, we mean the steady yearly rate that would produce the same total percentage increase over the specified number of years.

1. Use the final version of `goalSeek` to find the square root of two. Start by defining a function that squares numbers. Then see what argument value causes this function to return two.

2. Use the final version of `goalSeek` to find all the roots of two from the square root and the cube root up to the 10^{th} root. Instead of manually writing different functions to pass to `goalSeek`, use a function that creates them for you as they're needed.

3. Write a program to produce a neat multiplication table like this:

```
    1  2  3  4  5  6  7  8  9 10

 1  1  2  3  4  5  6  7  8  9 10
 2  2  4  6  8 10 12 14 16 18 20
 3  3  6  9 12 15 18 21 24 27 30
 4  4  8 12 16 20 24 28 32 36 40
 5  5 10 15 20 25 30 35 40 45 50
 6  6 12 18 24 30 36 42 48 54 60
 7  7 14 21 28 35 42 49 56 63 70
 8  8 16 24 32 40 48 56 64 72 80
 9  9 18 27 36 45 54 63 72 81 90
10 10 20 30 40 50 60 70 80 90 100
```

4. The following function returns 10 for some argument value between 0 and 5. We'd like to be able to find this value by making the call `goalSeek(mystery, 0, 5, 10, .001)`, but if we try, something goes wrong. Find and fix the problem.

```
def mystery(x):
  return x*x - 10*x + 25
```

5. Write a program to produce an *alphabetized* phone directory based on data in a file. The data file uses the following 'tab-delimited' format—there is a tab character between the family and given names and between the given name and the phone number.

```
Smith                   John            212-745-1234
Mayflower               Abigail Lou     718-255-6656
Brown-Appleby           Anthony         516-778-9813
```

The output should use the following format:

```
Brown-Appleby, Anthony  (516) 778-9813
Mayflower, Abigail Lou  (718) 255-6656
Smith, John             (212) 745-1234
```

It will help you to know that the string method `split` includes an optional argument that specifies the string to be considered as a separator. For example, `'abcdbcda'.split('b')` returns `['a', 'cd', 'cda']`.