Unit 3

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 prompts for input of a username and password and then displays an appropriate message, depending on whether these match. Use a dictionary to hold the usernames and associated passwords.

    0.2. Write a program that builds a word-line concordance for *Pride and Prejudice* and tests it using one or more look-ups included in the program code. Use `split` as a rough method for breaking lines into word.

    0.3. Revise the concordance program to allow a user to interactively look up as many words as desired.

    0.4. Revise the concordance program to use a more sophisticated method of breaking lines into words: convert uppercase letters to lowercase and replace non-alphabetic characters with spaces before using `split`.

    0.5. Revise the concordance program by moving the text clean-up code into a function and then calling that function before the application of `split`.

1. Reuse the `cleanedup` function in a program that finds the longest word used in *Pride and Prejudice*. Note that if you use `split` without `cleanedup`, your program will find 'inconveniences--cheerfulness', which is long, but not a single word.

2. Write a program that learns vocabulary in a language other than English. It asks the user for words in English, gives the translation if it has seen the word before and, if not, asks the user to enter it. Here is a sample run.

```
Enter English word: cat
Enter translation: gato

Enter English word: dog
Enter translation: perro

Enter English word: cat
cat = gato

Enter English word: cow
Enter translation: vaca
Enter English word: dog
dog = perro

Enter English word:
```

This program would be much more useful if it *saved* learned words between runs. We'll learn how to do that later in the course.

3. Write a program that compiles information on the number of occurrences of each word used in *Pride and Prejudice*. After the information is compiled, the user should be able to quickly find out how many times any particular words are used.

4. Write a program that prints the name Fred 100 times, one time per line.

5. Write a program that repeatedly gets a group of numbers from the user and displays the average. Define and use a function called `average` that takes in a list of numbers and returns the average. Here is a sample run:

```
Enter numbers: 1 4 3 2
Average: 2.5
Enter numbers: 1 2 3 999
Average: 251.25
Enter numbers: 5 2 -7
Average: 0.0
Enter numbers:
```

   Note that when you have a string like `'12'` you can't use it in an arithmetic expression: `'12'+'34'` is `'1234'` because `+` means concatenation for strings. The trick is to use the built-in function `int` which converts strings of digit characters into integers: `int('12')` is `12` and `int('12')+int('34')` is `46`.

6. Write a function called `lengths` that takes in a list of strings and returns a list of the lengths of the strings. If we pass it `['Ed', 'Ted', 'Fred', 'Jennifer']`, it will return `[2, 3, 4, 8]`. Use this function, along with the `average` function from the previous program and the `cleanedup` function, to write a program that accepts sentences from the user and reports the average length of the words in the sentence.

   Other than the function definitions, your program should contain only the following code:

```
while True:
   line = input('Enter a sentence: ')
   words = cleanedup(line).split()
```

```
print('Average word length:', average(lengths(words)))
```