**Problem Two**

I think you’re really going to like this one! Although it may prove to be trickier than the previous. I’m not sure if you’ve ever played any of the Fallout games (or if you’ve even heard of them), but if you have, this exercise is based on one of the minigames from Fallout!

I looked around a little bit online and I found the following:

<http://mitchellthompson.net/demos/terminal/>

Go ahead, try it out. I’ll wait.

Alright, so now hopefully you’ll have a basic grasp of how the game works. Just as a recap, here’s basically the rundown on what happens when you play.

1. Words are displayed to the screen, along with a bunch of garbled text (but we’re going to ignore that for the purposes of our program):



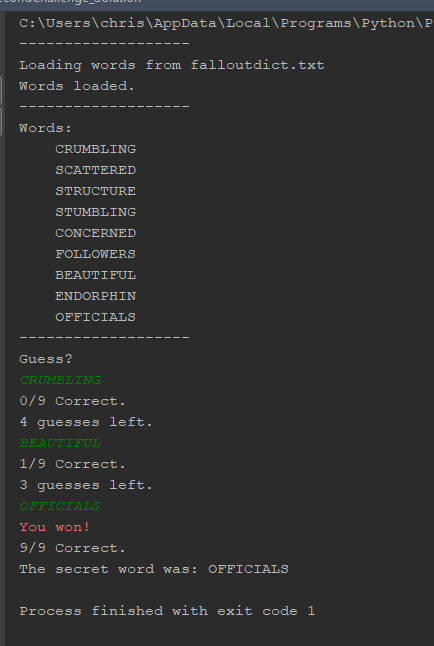
1. When you correctly select the secret word from the words displayed, you win and access is granted!



1. Otherwise, if you guess incorrectly, the program tells you how many characters from the word you selected match with that of the password (or secret word). For example, selecting ‘LATCHES’ displays 2/7 correct, since the two characters ‘ES’ match with two characters from the secret word ‘TOUZLES’. They both end in ‘ES’.



Now let’s try to model this game in Python! By the end of this challenge your result should look something like this:



Now let’s get started!

1. **Picking the secret word**
2. **Getting the additional words**
3. **Displaying the words**
4. **Checking if the user’s guess is valid**
5. **Comparing the user’s guess with the secret word**
6. **Picking the secret word**

If you read through the code already written, you’ll notice that the line:

C:\Users\chris\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_5.png

Gets a list of words from the text file ‘falloutdict.txt’ which is stored in the same folder as this Python script.