**General Instructions**:

1. **Submission date: 8.9.2016**. The exercise is quite short, and there is plenty of time.
2. Write the solutions to all the questions in a single \*.py file (no zip files necessary).
3. Write your name and ID in a comment at the top of the file.
4. **You can do the exercise in singles, pairs, or triplets**. If you submit in pairs or triplets, since we use the Moodle system, then each student should submit the exercise by himself. All students submitting together should include all their names and IDs in a comment at the top of the file.
5. Make sure your code runs smoothly with Python version 3.4.x or higher.
6. Use the exact function names defined in the exercise.
7. Check that your functions return what is requested in each question (if there is a return value at all).
8. You should put print statements in the global scope demonstrating your functions implementations. It is advised to use (at least) the example inputs for each question.
9. **Write readable code**: use meaningful names, don't overcomplicate things. Working code that is overcomplicated will lose (some) points.

**Question 1:** (33 Pts)

Write a function ParseEmails() that gets a list of emails in the form [name@company.com](mailto:name@company.com), where *name* and *company* can **only contain alphanumeric characters and underscore**.

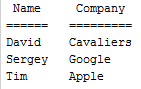
The function will return a dictionary mapping the names to their corresponding companies, only if the emails are valid according to the above rule.

**Question 2:** (33 Pts)

Now write a function PrintProfessionalContacts() that gets the dictionary from the previous question and will print a table of the names and companies, nicely aligned to the left.

Keep 3 spaces between the columns, make the first letters of names and companies capitalized, and add a title aligned to the center containing a title line. Here is a sample output:

emails = [**"sergey@google.com"**, **"tim@apple.com"**, **"david@cavaliers.com"**]  
mapping = ParseEmails(emails)  
PrintProfessionalContacts(mapping)



**Tip: look for the longest name and longest company and align accordingly.**

**Question 3:** (33 Pts)

Gorge, George and Gregory noticed that their names start with a 'g' and contain 5-7 letters. They made a bet about the number of English words that correspond to those **2 conditions**. Write a function that accesses <http://www.puzzlers.org/pub/wordlists/unixdict.txt>, finds all those words and save them to a file, each word in a line of its own. Make sure the words themselves are 5-7 letters long, not including line endings ('\n').