

## Programming Workshop – Homework Exercise 3

**Publication date:** *Thursday, November 13, 2014*

**Due date:** *Wednesday, November 26, 2014 @ 21:00*

### Problem 1:

1. Copy the files `phones.txt` from directory `/share/ex_data/ex3/`

**Note:** The purpose of 2-4 below is to create a file with all last names, such that the first letter is in upper case and the following letters are in lowercase. The three files you generate should each have exactly 365 lines.

2. Using a single series of piped commands, take `phones.txt` and generate a file called `lastnames-first.txt` which will contain the first letter of all last names (in upper case).
  - The first line of `lastnames-first.txt` should contain the letter A
3. Using a single series of piped commands, take `phones.txt` and generate a file called `lastnames-rest.txt` which will contain all the last names in lower case, **with the first letter removed**.
  - The first line of `lastnames-rest.txt` should contain the suffix `dams`
4. Using a single series of piped commands, take `lastnames-first.txt` and `lastnames-rest.txt` and generate a file called `lastnames.txt`, which will contain all the last names with the first letter is in upper case and the following letters are in lowercase.
  - The first line of `lastnames.txt` should contain the name `Adams`
5. Using a single series of piped commands, take `lastnames.txt` and `phones.txt` and generate a file called `phones-format1.txt`, which will contain all entries in `phones.txt`, but with the first name first, last name second (with only the first letter capitalized), and the telephone number last. Adjacent fields should be separated by a single `<tab>`.
  - The first line in `phones-format1.txt` should be  
`Andrew<tab>Adams<tab>7583`
6. Using a single series of piped commands, take `phones-format1.txt` and generate a file called `phones-format2.txt`, which contain full phone numbers using the following rule. The prefix of each number is 95X, where X is the last digit of the number.
  - The first two lines in `phones-format2.txt` should be  
`Andrew<tab>Adams<tab>953-7583`  
`Imelda<tab>Aguilar<tab>958-6518`

7. **Bonus (5pt.):** Using a single series of piped commands, take `phones.txt` commands (and none of the intermediate files above) and generate a file `phones-format2b.txt`, which is identical to `phones-format2.txt`.
8. Create a script (in TCSHELL) named `problem1sol` that will contain a line for each of the sections above (7 is optional). Create it such that if you are in a directory that contains only that file, and you type `problem1sol` on the command line, you will get the seven (or six) files you generated in the previous sections.

For example:

```
> ls
problem1sol
> problem1sol
> ls
lastnames-first.txt  lastnames-rest.txt  lastnames.txt
phones-format1.txt  phones-format2.txt  phones-format2b.txt
phones.txt          problem1sol
```

## Problem 2:

1. Copy the files `a-tale-of-two-cities.txt` from directory `/share/ex_data/ex3/`
2. Using a single series of piped commands, take the text file and generate a list of all words in the story and the number of occurrences. The list should be sorted alphabetically (dictionary order), but without ignoring case. Print the list out into the file `wordcount.txt`. The file should contain 11,044 lines. File `wordcount.txt` should be identical to `/share/ex_data/ex3/wordcount.txt`. (check using `diff`).  
**Note:** you need to remove punctuation marks, but note that some (like ``-``) bridge between two words, and you want those words to appear as two different words in your list.
3. Using a single series of piped commands, take the file `wordcount.txt` and generate a file `wordcount-filtered.txt`, which will contain only proper words (those containing only letters), which appear at least 100 times, including their counts. The file should contain 196 lines and the first three should be:
 

```
154 A
2864 a
163 about
```

4. Using a single series of piped commands, take the original text of the story, and output the **three most common** four letter proper words. This time, make sure your counts are **not case sensitive**. The most common word is 'that' with 1,956 appearances.
5. Create a script (in TCSHELL) named `problem2sol` that will contain a line for each of the sections above. Create it such that if you are in a directory that contains only that file, and you type `problem2sol` on the command line, you will get the three files you generated in (1-3), and the output of (4) will be printed on the screen.

### Submission Instructions:

In the directory `~/exercises/ex3/` you should have **only the two script files**:

- `problem1sol`
- `problem2sol`

**The directory should not contain any additional text files!**

All of the files should be placed on the server (in `~/exercises/ex3/`) at the time of deadline (21:00 @ Nov 26, 2014). Any changes made after that point will count as a late submission. For more information see the [Homework submission instructions](#) file on the course website.