COMS 3101 - Fall 2013
Perl Homework 3

- Due by start of class (Monday 4pm).
- See submission instructions.

1. **File listing:**

   a) Write a program **file_list.pl** to build an array of arrays - where each row corresponds to a file in the current directory and each column corresponds to a particular attribute of that file.

   The Unix (works in mac OS X too) command 'ls -l' lists the files in current directory in the Long Format form. The Long Format displays the following information for each file: permissions, number of links, owner name, group name, number of bytes in the file (size), date of last modification, time of last modification, file name (pathname). The information displayed can vary a bit from system to system.

   Execute the unix shell command 'ls -l' and obtain the results into an array. If you are using a Windows machine, you can use the corresponding windows shell command or use the Cunix machine to write this program.

   b) After building the array of arrays, print the 3rd field (which is the owner name) of the 2nd row (assuming you have more than one file in current directory).

   c) Print the { 3rd,4th,5th } fields of every row from the 2nd, row on. In other words, print multiple rows (assuming you have more than one file in current directory).

   Note 1: see lecture slides for how to execute a shell command and obtain the results into an array.
   Note 2: in addition to the file fields, the ls command returns a row which can be ignored, hence 2nd row really refers to the 1st file/directory

2. Team Results in European Cup Matches:

a) Download the file **uefa.txt** which contains the results of over 500 European cup futbol matches in recent years. The file is in tabular form, where each line lists data for a particular match (or pair of matches), and the fields are separated by a single tab (tab-separated-value format).

The fields are (in this order): team 1 name, team 1 country (abbr), team 2 name, team 2 country (abbr), game 1 result, game 2 result. Your task is to write a tool that allows users to search for particular teams/countries/matches.

b) Write a program **cup_match.pl** that first reads in the data file (uefa.txt) once (see part c below on how this should be implemented) and then asks the user repeatedly (in a loop until the user enters q) to enter a country (abbreviated), and team name separated by comma (e.g. Eng, Chelsea). For each request, the program lists all matches involving the requested team.

A match should be printed in one line: — rival team name, rival team flag (country), game 1 result, game 2 result (separated by commas). Each match should be followed by an empty line.

c) Write a subroutine that, given a filename, opens the file in the format described, reads in the data, and returns a hash named %games, which is a hash of hashes of arrays. The idea is to index the matches first by country and then by team. There can be many teams from the same country, hence, the two levels of hashes. A particular team may be involved in many matches (hence hash of hashes of arrays).

Each match should be represented as a hash with keys-value pairs as follows {team => “rival team name”, flag => “rival team country”, game1 => “result of game 1”, game2 => “result of game 2”}.

d) Keep the following notes in mind when writing the code:

   Note 1: abbreviated country names are three letters long (“Eng” for England, “Fin” for Finland)
   Note 2: A user can supply a country-team pair in small or capital letters (e.g. ENG or Eng or eng, Chelsea or chelsea). If user request is incorrect you should return an error message rather than a list of matches.
   Note 3: some of the matches have only one result, in that case game 2 result would be denoted NA (not available).
   Note 4: when populating the hash %games we do not distinguish between team 1 and team 2. In other words, for each match you’d have to make TWO entries into the hash.