COMS 3101 - Fall 2013
Perl Homework 2

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

1. **MTA bus number extraction:**

   a) Write a program **mta_extract.pl** to list all mentions of MTA bus numbers in the input file. It should recognize bus numbers in all five boroughs of NYC - Manhattan, Bronx, Brooklyn, Queens and Staten Island.

   The bus number contains mainly two parts - a part denoting the borough that the bus services followed by a numeric part. Express buses maybe denoted by an M after the first part or by an X. Sometimes there could be characters such as a,b,A added at the end to specify alternate routes.

   The list of bus numbers can be found at the url:
   http://www.mta.info/nyct/service/bus/bussch.htm

   Note 1: look at all the bus numbers in all 5 boroughs before writing your regular expression.
   Note 2: the program should list all bus numbers mentioned in the input file, one bus number per line.
   Note 3: there could be more than one mention of bus numbers per line.

   b) Test your code by creating (or copying) a file **test1.txt** with some random text (one paragraph is sufficient) that includes various bus numbers. Include the file in your submission.

   Your code should run as follows:
   $ perl mta_extract.pl test1.txt

   Example output:
   M104

BxM2
Bx10
Q20a

2. Phone number normalization:

a) Write a program **pn_norm.pl** to normalize the phone numbers mentioned in an input file.

A valid phone number could be mentioned with area code in parentheses - (917) 333 4444, or 917 333 4444, or 917-333-4444 or 9173334444 or even (917)-333-4444. The program should recognize any of these formats and then substitute with the standard format (917) 333 4444 and generate an output file with all phone number mentions normalized to this standard format.

Note 1: there could be more than one phone number per line.
Note 2: phone numbers won't run across lines. In other words, all parts of the number will be in the same line.
Note 3: the program should print the number of phone number mentions in the input file.

b) Test your code by creating (or copying) a file **test2.txt** with some random text (one paragraph is sufficient) that includes various phone numbers. Include the file in your submission.

Your code should run as follows:
$ perl pn_norm.pl test2.txt

Example output:
The passage contains 34 phone number mentions.
output file with normalized numbers 'test2n.txt' created

3. Algebraic chess notation:

a) Write a program **chess_moves.pl** to identify all the chess moves mentioned in an input file.

Download the **game.txt** file. The file contains a complete chess game (in algebraic notation) with analysis (text), and is your sample input file.

You can familiarize yourself with algebraic notation by reading the **acn.pdf** document.

Your program should:
1. Read the game.txt file and use regular expressions to identify all the piece moves (you should ignore pawn moves).

2. Moves may be preceded by move number and followed by annotation symbols. Print the percentage of moves that are followed by (any) annotation symbol.

3. Each move is moving a piece to a square. List the unique squares, and their number of mentions (again ignore pawn moves).

Note: although the chess board has a very simple structure, you should NOT generate a hash with appropriate keys a-priori. Rather suppose that the squares are not known in advance and that you are updating your counts on the fly.

b) Your code should run as follows:
$ perl chess_moves.pl game.txt

Example output:
Percent of moves followed by symbols: 20%
Unique squares:
e4 15
f6 13