CSE/ISE 337 Assignment 3 (Spring 2014)
Due date: Wednesday, April 9, at 11:55pm

Important! Must read: (a) When doing assignments, you **must** use the techniques that are described in the lecture notes. You may **not** use methods, modules, packages that were not covered in lectures. (b) Your assignment submission must be entirely your own. You **must** first read the lecture slides "0-Course-Overview.pdf" available in Blackboard – Documents – Lecture Slides, especially Slides 0-9 to 0-13, and follow them. (c) Start working on this assignment right away; you will **not** be able to finish it if you wait until the last day. (d) You must have "`use strict;`" and "`use warnings;`" at the start of your Perl programs.

1. **Simple matches** (16pts)

   (1) Does the pattern `foobar` match any of the following strings: `foobarshop, Foobar, barfoobar, linkfoobar`? If so, highlight the part of string that is matched (2pts)

   (2) Write a Perl script to verify your answer in (1). If no match, an error message "no match" should be printed. Otherwise, the matched part should be enclosed with symbols < and > and printed. You may input the pattern and the strings from the command line, from user input, or from a file. Document on how to use your script. Hint: read Module 4c slides for the coding part (5pts)

   (3) Write a regular expression that matches any string that contains at least one `m` followed by any number of `n`'s. Note that *followed* means ***immediately*** *followed*, and *any number* might be zero. Highlight the part of string that is matched, for the following strings: `dome, foobar, mmm, min, column, common`. Now write another regular expression to achieve the same. Use the script in (2) to verify your answers (5pts)

   (4) Find a regular expression that matches any string that contains any number of backslashes followed by any number of asterisks. Highlight the part of string that is matched, for the following five strings: `foo, \\**, bar  \\\***, *foobar\, /**/`. Use the same script to verify your answers (4pts)

2. **Given patterns** (18pts)

   What would the following regular expressions match or do? Be precise, complete, and give reasons. For each pattern, give two good example strings to illustrate your answer. Note that *good* means it is as general and different from other examples as possible.

   (1) `/"([^"]*)"/` (6pts) Note that in the pattern, `"` is a double-quote.

   (2) `/^\b[\w.]{2,9}\b$/` (6pts)

   (3) `s/(\w+)(\s)+(\w+)/\3\2\1/` (6pts)

3. **Finding patterns** (20pts)

   (1) Write a Perl script that prints any input line that matches both `foo` and `bar`. It prints "No match" otherwise. Your script processes one line at a time from user input (not from files or the command line), until the user terminates the input with Ctrl-d. Note that the order in which the two patterns appear does not matter. Give two good examples to verify your script (6pts)

(2) Find a regular expression that matches three consecutive copies of the regular expression `foo|bar`. Your pattern should match `foofoofoo`, `foobarbar`, `barbarbar`. Use the program in 1(2) to test your pattern. Give two good examples to verify your script. Hint: think of quantifiers (4pts)

(3) Find a regular expression that matches a word. A word is a sequence of letters (either lower or upper case), digits, and underscores. Now find a different regular expression that does the same thing. Give two good examples to verify your answers. (4pts)

(4) Find a regular expression that matches any line of input that has the same word repeated two or more times consecutively in a row. Assume there is one space between consecutive words. Use the script in 1(2) to test your pattern. Also pay attention to word boundaries. Give two good examples to verify your answer. Hint: use group and back references (6pts)

4. **Application to Perl Programming** (6pts)

A palindrome is a string that reads the same backwards as forwards. For example, each of the following strings is a palindrome: 12321, abba, 66666, and xy9yx. Write a Perl program that removes all non-digits from input first, prints the number left and checks to see if the number left is a palindrome. Hint: use the reverse() function

Deliverables

Your assignment submission should include two files: (a) Answers to all written questions, the good examples that you come up with, and a printout of all programs that you write. Concatenate them into one **plaintext** file called "a3-printout.txt". Each answer and program must be clearly labeled with its corresponding question/part numbers. (b) A **zip** file that includes all individual programs that you write. Name it "a3-source.zip". Be sure to name each program using its question and part number, e.g., "q2p1", "q2p3", and so on. You should include certain amount of program documentation, i.e., in-line comments, in your programs for important steps used. Do not repeat what the line of code says; rather write comments to help readers to understand your code.

Total: 60 points

Submission instructions

The handing-in will be through Blackboard Assignment. The submission instructions are at: http://it.stonybrook.edu/help/kb/creating-and-managing-assignments-in-blackboard.
You **must** read the submission instructions very carefully, and check to make sure your assignment has been submitted correctly **before** the deadline.

You can only submit once! However you can save your work by clicking "Save" as many times as you like. Only click "Submit" after you have checked and are certain that all requirements are followed.

Late submissions will not be accepted. The due date is **11:55pm on Wednesday, April 9.**