CSE/ISE 337 Assignment 3

Due date: Friday, April 8, 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 0-14, 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 all your non-single-line Perl programs.

1. **Simple matches** (14pts)

   (a) Does the pattern `ord` match any of the following strings: `Ordinary, order, afford, cordford, 'ORD airport'`? If so, highlight the part of string matched (2pts)

   (b) Does the pattern `.p` match any of the following strings: `p, a.pl, <p>, developing, ppp`? If so, highlight the part of string matched (2pts)

   (c) Does the pattern `fo.*` match any of the following strings: `foo, foobar, barfoo, foobarfoo, portfollio`? If so, highlight the part of string matched (2pts)

   (d) Does the pattern `\Bwork\b` match any of the following strings: `work, back2work, to_do_work, networking, "coursework list'`? If so, highlight the part of string matched (2pts)

   (e) Write a Perl script to verify your answers above. If no match, print a "no match" message. Otherwise, print the input with the matched part enclosed between symbols < and >. Input the pattern from the stdin first, then the strings to be matched from stdin one at a time indefinitely, until ctrl-d or ctrl-c is entered.

   Hint: read the Module 4b slides on Regex Variables for the script coding. (6pts)

2. **Given patterns** (16pts)

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

   (a) `/[\w-]{5,10}/` (2pts)

   (b) `/^[+-]?\d+\.?\d*$/` (3pts)

   (c) `s/<(\/?)b>/<\1strong>/ig` (3pts)

   (d) `/<(.*?)>.*?<\/\1>/` (3pts)

   (e) `/<(.*?)>.*<\/\1>/` (2pts) Hint: How does this one differ from the one in (d)?

   (f) `/\s+[a-z\d_-]+@[\da-z.]+\.[a-z]{2,6}\s+/` (3pts)

3. **Finding patterns** (15pts)

   Use the script in 1e, <u>verify your answer with two good examples; include them in the handin.</u>

   (a) Find a regular expression that matches the course number of any <u>undergraduate</u> CSE or ISE courses. E.g., CSE101, ISE321, ISE487. See the following URLs for these courses: http://www.cs.stonybrook.edu/students/Undergraduate-Studies/csecourses for CSE and

http://www.cs.stonybrook.edu/students/Undergraduate-Studies/isecourses for ISE ones. Note that 'CSE' and 'ISE' can be written in lower cases. Course numbers such as 011 are not valid. (5pts)

(b) Write a regular expression that finds those file names that are *not* made of the word characters and the characters '.'(dot), '/'(slash), or '-'(hyphen). Then, write another different regular expression that does the same. (5pts)

(c) Find a regular expression that matches any line of input that has at least one word repeated 2 or more times. You may assume that all inputs are made of word characters, and there is one space between consecutive words. Don't forget the word boundaries. (5pts)

(d) (Bonus 3pts) Find a regular expression that matches those strings in which every pair of a's has a 'b' in between them. E.g., aba, ababa are matched, while abaa, abaca are not.

4. **Application to Perl Programming** (14pts)

(a) Write a Perl script that takes input from a file and replace all dates in the file in the format mm/dd/yy using the following scheme. If the two-digit year is in between 50 and 99 inclusive, add '19' in front of the two-digit year. Otherwise, add '20' instead. E.g., 05/06/75 becomes 05/06/1975, while 12/21/06 becomes 12/21/2006. You **must** use regular expressions to do the substitution. The file name is specified in the command line. Print the changed file content to the screen. Note that there may be multiple date strings in one line. (7pts)

(b) Write a Perl script that searches a dictionary of words and finds those words that satisfy a simple form of user specification: `L m c n r` that is entered at the command line. `L` denotes the length of the word in characters. `m` denotes a position in the word where `m=1` denotes the first character in the word, `m=2` denotes the second character in the word, and so on. `c` denotes an English letter. `n` denotes another position in the word, and `r` denotes another English letter.

As an example, the command `perl q4b.pl 9 4 s 6 m` issued at the OS command line would run your script called `q4b.pl` and find all those words in dictionary that are 9-character long whose 4th character is `s` and 6th character is `m`. The dictionary that we use is the standard Linux dictionary file `/usr/share/dict/words`. You must use regular expressions to search the dictionary for desired words. The words found are printed to the screen. (7pts)

Deliverables

Your assignment submission should include two files: (a) A **plaintext** file called "a3-printout.txt". It contains answers to all written questions and a printout of all programs that you write. It also contains instructions on how to use your programs and sample input/output. 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., "q1e.pl", "q4a.pl", and so on. You should include certain amount of in-line comments for important steps used. Do not repeat what the line of code says; rather write comments to help to understand your code. (1pt on format)

Important: we will use the allv Linux machines to mark your A3.

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 Friday, April 8.**