**CS 3421 Week 8 Problems: Perl**

**1.** For an arbitrary number of arguments, print any arguments corresponding to a valid file in alphabetical order followed by all other arguments in reverse alphabetical order. Separate each output with a newline character.

#!/usr/bin/perl –w  
use strict;  
  
my @validList;  
my @invalidList;  
  
my $arg;

foreach $arg (@ARGV)  
{  
 if( -f $arg ) {  
 push @validList, $arg;  
 }  
 else {  
 push @invalidList, $arg;  
 }  
}  
print “Valid: \n”;  
print join(“\n”), sort(@validList));  
print “\nInvalid: \n”;  
print join(“\n”, reverse(sort(@invalidList)));

**2.** For an arbitrary file where the first column is an item ID and the second column is a quantity (see input8-2.txt below for an example), use a hash to calculate the total inventory for each item and print the accumulated values in the same format.

**input8-2.txt:**

AB123 1

BD122 2

AB123 2

XD984 2

BD122 7

QR899 3

#!/usr/bin/perl –w  
use strict;  
  
my %idHash;  
  
my $id;  
my $count;

open(IN, “<”, $ARGV[0]) or die “Couldnot open file”;  
  
while($line = <IN>) {  
 ($id, $count) = split(“ “, $line);  
 $idHash{$id} += $count;  
}  
foreach my $key (keys %idHash) {  
 print “$key $idHash{$key}\n”;  
}

**3.** Given text containing phone numbers in different formats (see below), modify the text so that all phone numbers are in the format of (XXX)-XXX-XXXX.

**input8-3.txt:**

My phone number is 2101234567.

123 123-1234 is your phone number.

The phone number (123) 123-1234 is different than 123-555-1234.

(123)-123-1234 is the format these numbers should be in.

#!/usr/bin/perl –w  
use strict;  
  
open (IN, “<”, $ARGV[0]) or die “Could not open file\n”;  
  
my $line;  
  
while($line = <IN>) {  
 $line =~ s/\(?(\d{3})[^\w]\*(\d{3})[^\w]\*(\d{4})/($1)-$2-$3/g;  
print “$line”;  
}

**Expected output**:

My phone number is (210)-123-4567.

(123)-123-1234 is your phone number.

The phone number (123)-123-1234 is different than (123)-555-1234.

(123)-123-1234 is the format these numbers should be in.

**4.** For the text file below, use Perl’s backreferencing ability to replace any instances of header tags (e.g., <h1></h1>, <h2></h2>, etc) that contain the word “Project” (capitalized or not) with the same text but with <strong></strong> tags instead. Assume all matches will be across single lines and will be the only thing on the line.

**input8-4.html:**

<h3>project 4</h3>

<h2>assignment 5</h2>

<h1>Old Project</h1>

<em>Project 7</em>

#!/usr/bin/perl –w  
use strict;  
  
while(<>) {  
 if($\_ =~ m/<h\d>(.\*([pP]roject.\*)<\/h\d>/) {  
 print “<strong>$1</strong>\n”;  
 }  
 else {  
 print $\_;  
 }  
}

**Expected output**:

<strong>project 4</strong>

<h2>assignment 5</h2>

<strong>Old Project</strong>

<em>Project 7</em>

**NOTE:** Unless it is on short, well-known snippets, it is generally not a good idea to use regular expressions to parse arbitrary html since html is not a regular language.