**Assignment 4: Regular Expressions 2**

|  |  |
| --- | --- |
| **Date:** | Monday, June 8th at 1pm |
| **Due:** | Wednesday, June 10th at 1pm |
| **Assignment Type:** | Group (Up to 6 – may be randomly assigned) |
| **Assignment Title:** | Regular Expressions |
| **Style:** | One problem at a time on Discord – WORK TOGETHER |
| **External Sources:** | You are allowed notes, books, and searches |
| **Description:** | This lab covers Regex topics including Backreferences, Accessing Matches, and Substitution. Please review the page below for all user input asked for in the problem set given. |
| **Points** | 20 |
| **Starting Files** | **mkdir lab4**  **cd lab4**  **cp –r /afs/umbc.edu/users/j/d/jdixon/pub/cs433/lab4/\* .** |

Enter your group member names here:

Michael Wellen

Ilya Golubs

Rajat Shetty

William Jenkins

Elliot Ressler

**Backreferencing**

1. Create the Perl script “lab4\_1.pl” that will read in everything from the file “lab4\_data.txt” and finds all airports with City twice.

The output should look like this:

Kansas City     Kansas City Airport     MCI     Medium  11850825

Salt Lake City  Salt Lake City Airport  SLC     Large   25554244

Show both code and result view!!

Michael’s Answer

#!/bin/perl

use strict;

use warnings;

my $filename = 'lab4\_data.txt';

open(my $input, '<:encoding(UTF-8)', $filename)

        or die "Could not open file '$filename' $!";

while (my $row = <$input>)

{

        chomp $row;

        print "$row\n" if $row =~ /(City).\*\w.\*\1.\*/

}

close $input;

**[mwellen1@linux6 lab4] perl ex1.pl**

**Kansas City     Kansas City Airport     MCI     Medium  11850825**

**Salt Lake City  Salt Lake City Airport  SLC     Large   25554244**

**Matching**

1. Create the Perl script “lab4\_2.pl” that uses at least three groups to return any airport codes where all three of the letters are between A-K from “lab4\_data.txt”.

CAK

IAH

JFK

GEG

IAD

Show both code and result view!!

**Michael’s Solution**

#!/bin/perl

use strict;

use warnings;

my $filename = 'lab4\_data.txt';

open(my $input, '<:encoding(UTF-8)', $filename)

        or die "Could not open file '$filename' $!";

while (my $row = <$input>)

{

        chomp $row;

        if($row =~ /([A-K])([A-K])([A-K])/){

            print $1,$2,$3, "\n"

        }

}

close $input;

**[mwellen1@linux6 lab4] perl ex2.pl**

**CAK**

**IAH**

**JFK**

**GEG**

**IAD**

**Substitution**

1. Create the Perl script “lab4\_3.pl” that will use the results from matching above and adds Airport Code with the airport code from “lab4\_data.txt”.

Airport Code:CAK

Airport Code:IAH

Airport Code:JFK

Airport Code:GEG

Airport Code:IAD

Show both code and result view!!

Ilya’s Solution

#!/bin/perl

use strict;

use warnings;

my $filename = 'lab4\_data.txt';

open(my $input, '<:encoding(UTF-8)', $filename)

        or die "Could not open file '$filename' $!";

while (my $row = <$input>)

{

        chomp $row;

        if($row =~ /([A-K])([A-K])([A-K])/){

            print "Airport Code:".$1,$2,$3, "\n"

        }

}

close $input;

**Backreferencing**

1. Create the Perl script “lab4\_4.pl” replace the relative link in this to an absolute link.

**<a href="happy">A link</a><a href="img/happy.jpg">An image</a>**

**<a href="happy2">A link</a><a href="happy2.jpg">An image</a>**

**<a href="excited">A link</a><a href="img/current/excited.jpg">An image</a>**

The output should look like this:

<a href="happy">A link</a><a href="http://csee.umbc.edu/jdixon/img/happy.jpg">An image</a>

<a href="happy2">A link</a><a href=" http://csee.umbc.edu/jdixon/happy2.jpg">An image</a>

<a href="excited">A link</a><a href=" http://csee.umbc.edu/jdixon/img/current/excited.jpg">An image</a>

Show both code and result view!!

#!/bin/perl

use strict;

use warnings;

my @href = ('<a href="happy">A link</a><a href="img/happy.jpg">An image</a>',

'<a href="happy2">A link</a><a href="happy2.jpg">An image</a>',

'<a href="excited">A link</a><a href="img/current/excited.jpg">An image</a>');

my $length\_href = @href;

for(my $i = 0; $i < $length\_href; $i++){

  my $add\_part = "href=\"http://csee.umbc.edu/jdixon/";

  $href[$i] =~ s/href="([^"]\*\.jpg")/$add\_part$1/gm;

  print($href[$i]."\n");

}

**Lookahead**

1. Create the Perl script “lab4\_5.pl” that replaces breaking and entering or breaking and theft with theft. The only tricky part is that you must match cases!

Here is the starting paragraph:

Breaking and entering, sometimes called housebreaking, is an unlawful entry into a building or other location for the purposes of committing an offence. When breaking into a home, the goal is usually theft. In some cases, breaking and entering is considered breaking and theft. Interestingly, break and enter a home would be the same as to burgle a home.

Here is the finishing paragraph:

Theft, sometimes called housebreaking, is an unlawful entry into a building or other location for the purposes of committing an offence. When breaking into a home, the goal is usually theft. In some cases, theft is considered theft. Interestingly, break and enter a home would be the same as to burgle a home.

Show both code and result view!!

Rajat’s version:

#!/bin/perl

use strict;

use warnings;

my $filename = "lab4\_dataC.txt";

open(my $input, '<:encoding(UTF-8)', $filename)

        or die "Could not open file '$filename' $!";

while (my $row = <$input>) {

    chomp $row;

    $row =~ s/breaking\sand\s(entering|theft)/theft/g;

    $row =~ s/Breaking\sand\s(entering|theft)/Theft/g;

    print "$row\n";

}

close $input;

**OR:**

Prof. Dixon said “Must use lookahead”, so did this implementation. Seems to work fine

#!/bin/perl

use strict;

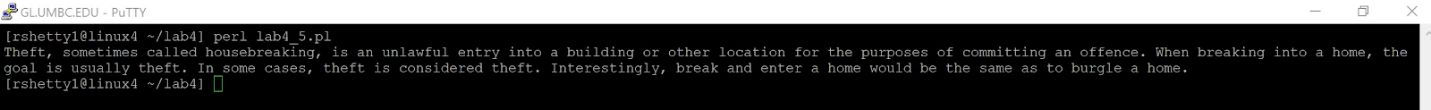
use warnings;

my $info = "Breaking and entering, sometimes called housebreaking, is an unlawful entry into a building or other location for the purposes of committing an offence. When breaking into a home, the goal is usually theft. In some cases, breaking and entering is considered breaking and theft. Interestingly, break and enter a home would be the same as to burgle a home.";

$info =~ s/breaking\sand\s(entering|theft)/theft/g;

$info =~ s/Breaking\sand\s(entering|theft)/Theft/g;

print "$info\n";



**Negative Lookahead and Behind**

1. Create the Perl script “lab4\_6.pl” that will replace Hillary (of Hillary Clinton) to Former Secretary Clinton. You must use negative lookahead or negative lookbehinds in this answer.

Start with this:

Once Hillary Clinton was talking about Sir Edmund Hillary to Hillary Makasa and then Hillary Clinton had to run off on important business.

Show both code and result view!!