Need dynamic web content - you want Perl

1. The ass2 starting point contains this function. Discuss how it works.

```
sub user_page {
    my $n = param('n') || 0;
    my @users = sort(glob("$users_dir/*"));
    my $user_to_show  = $users[$n % @users];
    my $details_filename = "$user_to_show/details.txt";
    open my $p, "$details_filename" or die "can not open $details_filename: $!";
    $details = join '', <$p>;
    close $p;
    my $next_user = $n + 1;
    return <<eof
<div class="matelook_user_details">
$details
</div>
<p>
<form method="POST" action="">
    <input type="hidden" name="n" value="$next_user">
    <input type="submit" value="Next user" class="matelook_button">
</form>
eof
}
```

2. The starting point code for assignment 2 starts by prints out these lines?

```
Content-Type: text/html

<!DOCTYPE html>
<html lang="en">
<head>
<title>matelook</title>
<link href="matelook.css" rel="stylesheet">
</head>
<body>
<div class="matelook_heading">
matelook
</div>
```

   Quickly discuss why these lines are being printed, in particular

   1. Why is line 2 blank?
   2. What else might appear in the first 2 lines?
   3. What does `class="matelook_heading"` mean?

      1. The blank line marks the finish of the header.
      2. Cookies!
      3. The attribute `class="matelook_heading"` results in the style `.matelook_heading` from `matelook.css` being applied to the div (see next question) .

3. The file `matelook.css` supplied for the assignment contains this:

```
.matelook_heading {
    padding-top: 1em;
    padding-bottom: 1em;
    text-align: center;
    font-size: x-large;
    font-weight: bold;
    text-decoration: underline;
}

.matelook_user_details {
    background-color: #ABCDEF;
    color: #204142;
    white-space: pre;
    border:thin solid #204142;
    border-radius: 1em;
    padding-left: 0.42em;
}

.matelook_button {
    background-color: #FEDBCA;
    border:thin solid #904142;
    border-radius: 0.42em;
    color: #904142;
}
```

   CSS isn't really covered in COMP[29]041 - but can you quickly guess what the CSS does?

4. A good way to test out code for assignment 2 is to write small programs which can be run from the command line.
   A COMP[29]041 student wrote this small program to test Perl that wanted to use in `matebook.cgi` .

   It takes a zid as a command-line argument and is meant to print the filenames of the user's posts in reverse chronological order (most recent first).

```
#!/usr/bin/perl -w

$dataset ="dataset-medium";
foreach $user (@ARGV) {
    my $posts_dir = "$dataset/$user/posts";
    foreach $post_filename (reverse (glob "$dataset/$user/posts/*/post.txt")) {
        print "$post_filename\n";
    }
}
```

Unfortunately it prints them in the wrong order. Note `dataset-medium/z5023159/posts/17/post.txt` is the most recent post so should be printed first.

```
$ print_posts.pl z5023159
dataset-medium/z5023159/posts/9/post.txt
dataset-medium/z5023159/posts/8/post.txt
dataset-medium/z5023159/posts/7/post.txt
dataset-medium/z5023159/posts/6/post.txt
dataset-medium/z5023159/posts/5/post.txt
dataset-medium/z5023159/posts/4/post.txt
dataset-medium/z5023159/posts/3/post.txt
dataset-medium/z5023159/posts/2/post.txt
dataset-medium/z5023159/posts/17/post.txt
dataset-medium/z5023159/posts/16/post.txt
dataset-medium/z5023159/posts/15/post.txt
dataset-medium/z5023159/posts/14/post.txt
dataset-medium/z5023159/posts/13/post.txt
dataset-medium/z5023159/posts/12/post.txt
dataset-medium/z5023159/posts/11/post.txt
dataset-medium/z5023159/posts/10/post.txt
dataset-medium/z5023159/posts/1/post.txt
dataset-medium/z5023159/posts/0/post.txt
```

Rewrite the code, fixing the bug.

5. Write a CGI script which pseudo-randomly chooses a number between 1 and 100 and allows a user to guess it, indicating higher, lower or correct for each guess it.

   Match the behaviour of this example implementation:

guess_number.cgi (tut/perlcgi/guess_number.cgi)

I've thought of number 0..99

```
<html>
<head>
   <title>Guess A Number</title>
</head>
<body>
I've  thought of number 0..99
   <form method="POST" action="">
      <input type="textfield" name="guess">
      <input type="hidden" name="number_to_guess" value=
   </form>
</body>
</html>
```

Sample solution

```
#!/usr/bin/perl -w

use CGI qw/:all/;
use CGI::Carp qw(fatalsToBrowser warningsToBrowser);

# Simple CGI script written by andrewt@cse.unsw.edu.au
# Outputs a form which will rerun the script
# An input field of type hidden is used to pass an integer
# to successive invocations

$max_number_to_guess = 99;

print <<eof;
Content-Type: text/html

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Guess A Number</title>
</head>
<body>
eof

warningsToBrowser(1);

$number_to_guess = param('number_to_guess');
$guess = param('guess');

$game_over = 0;

if (defined $number_to_guess and defined $guess) {
    $guess =~ s/\D//g;
    $number_to_guess =~ s/\D//g;
    if ($guess == $number_to_guess) {
        print "You guessed right, it was $number_to_guess.\n";
        $game_over = 1;
    } elsif ($guess < $number_to_guess) {
        print "Its higher than $guess.\n";
    } else {
        print "Its lower than $guess.\n";
    }
} else {
    $number_to_guess = 1 + int(rand $max_number_to_guess);
    print "I've  thought of number 0..$max_number_to_guess\n";
}

if ($game_over) {
print <<eof;
    <form method="POST" action="">
        <input type="submit" value="Play Again">
    </form>
eof
} else {
print <<eof;
    <form method="POST" action="">
        <input type="textfield" name="guess">
        <input type="hidden" name="number_to_guess" value="$number_to_guess">
    </form>
eof
}

print <<eof;
</body>
</html>
eof
```

6. Write a shell pipeline that reports any e-mail address which is being used by more than 1 matelook users in the large dataset

```
$ egrep '^email' dataset-large/*/user.txt|sed 's/.*email=//'|sort|uniq -d
```

The above pipeline break if the string 'email=' appears in an email address. One way to avoid it is this:

```
$ egrep '^email' dataset-large/*/user.txt|cut -d= -f2- |sort|uniq -d
```

egrep also as a -h flag which supresses printing of the filename:

```
$ egrep -h '^email' dataset-large/*/user.txt|sort|uniq -d
```