

Introduction to e-Technologies

Test 1 - October 10, 2001

Name: _____

1 Refer to the following code to answer the following questions:

```
1 #!/usr/bin/perl
2
3 use IO::Socket;
4 unless (@ARGV > 1) {die "usage: $0 host document ..."}
5 $host = shift(@ARGV);
6 foreach $document ( @ARGV ) {
7     $remote = IO::Socket::INET->new(
8         Proto => "tcp",
9         PeerAddr => $host,
10        PeerPort => "(80)",
11    );
12    unless ($remote) {die "cannot connect to http daemon on $host" }
13    $remote->autoflush(1);
14    print $remote "GET $document HTTP/1.0\n\n";
15    while (<$remote>) {
16        $page .= $_;
17    }
18    print "${page}\n";
19    -close $remote;
20 }
```

1.1 Explain line 4:**1.2 Explain lines 6-11:****1.3 Explain line 13:**

1.4 Explain line 16:

2 Using the program on the previous page, you request the Notre Dame home page and redirect output to a file. Opening that file in vi, and setting line numbers on, you see the following:

```
1 HTTP/1.1 200 OK^M
2 Date: Tue, 09 Oct 2001 21:41:17 GMT^M
3 Server: Apache/1.3.3 (Unix)^M
4 Connection: close^M
5 Content-Type: text/html^M
6 ^M
7 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">^M
8 ^M
9 <HTML>^M
10 <HEAD>^M
11     <TITLE>University of Notre Dame</TITLE>^M
12 ^M
13 <!--JavaScripts as server-side includes-->^M
```

2.1 What are those ^M at the end of each line? Explain what's going on?

2.2 Ignoring those ^M, explain the rest of lines 1-6:

2.3 You wish to write a simple Perl web browser. Lines such as 11 look interesting. You think you may want to grab the string inside the <TITLE> ... </TITLE> tags and hold it in a scalar variable for later use in your program. Provide the Perl code that would find, extract and save that string. Reminder: HTML is mostly case insensitive.

- 3 You are developing a simple Perl web browser, and your first prototypes struggle with the graphic and table loaded Notre Dame home page. You remember that the your teacher said, it is OK to link from such pages to "text only" versions of such pages. You observe the following downloaded by your prototype program:**

```

49 <BODY TEXT="#000099" bgcolor="#ffffff" LEFTMARGIN=0 TOPMARGIN=0 MARGINH EIGHT=0
MARGINWIDTH=0 LINK="#22064B">^M
50 <table WIDTH="100%" height="16" BORDER="0" CELSPACING="0" CELLPADDING="0">^M
51 <tr><td align="center">^M
52 ^M
53     <TABLE WIDTH="770" height="16" BORDER="0" CELSPACING="0" CELLPADDING="0">^M
54     <TR BGCOLOR="#F7E79C" align="left">^M
55     <!-- row 1 -->^M
56     <TD height="20" width="10">&nbsp;</td><td> <font color="#000066"
face="Arial, Helvetica, sans-serif" size="1"><A HREF=http://www.nd.edu/textonly/
index.shtml>TEXT VERSION</a></font>^M
57 </td>^M

```

- 3.1 Provide below, the Perl code that will find, extract and save in a scalar variable, the URL for the "TEXT VERSION". For this exam, assume that string will always mark the location of the required URL.**

- 3.2 Recall that a full URL may optionally include a port number. Write a Perl function, `parse_URL()`, below that accepts as an argument a URL, parses the URL into the four elements needed by the `IO::Socket::INET` object to link to the URL, i.e., something like: (`$scheme`, `$hostname`, `$port`, `$path`), and returns those four elements as an array.**

```
sub parse_URL {
```

4 Using pseudo Perl, sketch out below, the design of a text-only web browser. Design your program using Perl subroutines for maximum modularity.

Considerations: How will your program handle tags, both those you choose to render and those that you choose to ignore? Will your program analyze the response line and headers? Will your program handle URL redirection? What will your program do when the server does not respond? How will you extract links? How will your program handle relative links? What will your program do about image links?

5 The following are based on end-of-chapter exercises:

5.1 Write a Perl program to change directory to a location specified as input, then list the names of the files in alphabetical order after changing there.

5.2 Write a Perl subroutine to take a numeric value from 1 to 9 as an argument and return the English name (such as one, two, or nine). If the value is out of range, return the original number as the name instead. Write a Perl program that uses that subroutine to take two numbers and add them together, displaying the result as, for example, Two plus two equals four. (Don't forget to capitalize the initial word.)

6 Define and/or explain the following:

6.1 URL encoding

6.2 Request methods

6.3 XHTML

6.4 CSS

6.5 Perl code as a munition

6.6 MIME

6.7 OSI reference model