2nd Midterm for CS 201

This exam tests your ability to think and write programs and/or subroutines in PERL, by hand, without books or notes. The 80% emphasis in grading is on whether you seem to know what you are doing. The 20% emphasis is on whether, for instance, you remembered to put semicolons in all the right places, or whether your grammar is perfect. I often deduct points (typically up to 10%, sometimes more) for failure to follow instructions.

You have until the end of the class period to complete the exam. Use the paper that I provide. During the test, keep all backpacks, papers, notes, and books on the floor. Only pens, pencils, erasers, and paper should be at table level. Avoid sitting directly next to another student.

For fairness and uniformity I typically grade the problems separately in random order. **Do NOT write your NAME anywhere on your answer sheets. A Test ID number will be given you to use instead.** Use one sheet of paper for each problem. If possible, write the complete answer on one side of the sheet and leave the other side blank. If necessary use both sides of one sheet. If you break a problem across two or more sheets of paper, I will ignore all but the first sheet. If you write two or more answers on the same sheet, I will ignore all but the first. Do not staple, fold, or tear your sheets.

Leave the edges of your paper blank, at least as much as the edges of **this sheet** are blank (one inch, or 25 mm, on all four sides). (You may lightly draw a border if it helps you.) Put your id number, then a dash (-), then the problem number, in the upper left corner (↖) on the front of each sheet (but not in the part of the page that should be blank). Draw a box around it. For example, if your assigned test id number is 102 and the problem number is 4, write 102-4 in the upper left corner, one inch down from the top of the paper, and one inch in from the left edge of the paper. Remember, do NOT write your name anywhere on the sheet.

Bring your own pen or pencil. Do not use red ink (because I use it for grading). Use a good-contrast writing method, e.g., dark pencil or ink. Light (faint) pencil can be very hard for me to read. Light (faint) pencil on top of erasures can be almost impossible to read. If I have trouble reading what you wrote, I will probably mark it wrong. It is okay to cross out things clearly instead of erasing them.

Explanations

I do not allow books or notes, including dictionaries, especially electronic dictionaries. If you are unclear about the meaning of a question, or some part of the question, you can ask me about it. If I can give you an answer, I will.

Confidentiality

You are **not allowed** to take the problem sheet from the room. However, starting at 3 PM today you may discuss the contents of this test with anyone you wish. Before that, do not discuss it in *any* way with *any* person (even a non-student) who has not yet completed it and turned it in.

Proctor Instructions

Student computer screens should be turned off or otherwise blanked. Student backpacks, papers, notes, and books should remain on the floor. Collect the tests **and** the answer sheets as students leave.

Finding Out Your Grade

Graded tests will be placed in a box outside my door, and uncollected tests will be returned to you in class on Friday.

Perl Hints: These lines may be handy.

```
#!/usr/bin/perl -w
( $a, $b, $c, $d ) = @ARGV;
( $a, $b, $c, $d ) = @_;
$x = int ( rand(6) ) + 1;
```

CGI Hints: These lines may be handy.

```
<body>
<!doctype html public ...>
<h1>
<head>
<html>
<table>
<title>
<form method=post action="bar.cgi"
  enctype="multipart/form-data">
<input type=text name=x value="" size=20>
<input type=ttt name=xxx value="yyy">
  type=button|checkbox|file|hidden|image
  type=password|radio|reset|submit|text
<tr><td>asdf<td>qwer<tr><td>zxcv<td>uiop
chmod 755 public_html/foo.cgi
s/%([\dA-F][\dA-F])/pack("c",hex($1))/eig;
$x =˜ /nuts=(.*)&bolts=(.*)&submit=submit/;
```

DBI Hints: These lines may be handy.

```
$foo -> disconnect();
$foo -> execute();
$foo -> finish();
$foo = $bar -> fetchrow_hashref();
$foo = $bar -> prepare( $bletch );
$foo = DBI -> connect(
  "DBI:mysql:$d:$h",$u,$p );
@foo = $bar -> fetchrow_array();
alter table foo drop cost;
create database DDD;
create table scores ( ... );
delete from inventory where ID=99;
drop table TTT;
grant all on DDD.* to UUU
  identified by "PPP";
insert scores values ( "Bob", 70 );
mysql -p -h HHH -u UUU
select * from scores order by cost;
set password = password("whatever");
show databases;
show tables;
update inv set desc="yadda" where ID=37;
use DDD;
use DBI;
```

________________ Test ID Number

________________ Student ID Num

Use the Test ID Number shown above to identify each of your answer sheets.

Write your seven-digit BYUH Student ID number in the blank above, and turn in this sheet along with your answer sheets. It will be kept separate until grading is completed, and will then be used to assign your score to the proper person.

Remember to follow instructions carefully. Each problem is worth ten points, up to a maximum of 40 points for the whole test.

1 Starline

Prompt for and read in one number. Use a loop to print that many stars ("*") on one line.

2 Odd Total, Even Total

Read lines from STDIN until you get a blank line. On each line is a whole number (e.g., 13 or 98). Add all the odd numbers and even numbers separately. (Even means divisible by two, such as 2, 4, 6, 8, and so on.) Report (a) how many numbers were read, (b) what is the odd total, (c) what is the even total. Do not use any kind of array. Use a small, constant amount of storage.

3 Lucky 3

Ask for two numbers, A and B. Print the numbers from A to B, smallest to largest, but print stars around any number that has a "3" in it. Example: A is 15. B is 10. Print 10, 11, 12, *13*, 14, 15. (Assume the numbers are integers. Don't worry about spacing, commas, or newlines.)

4 Dice

Use something like `$x=int(rand(6))+1` to "roll" two six-sided dice by generating two random integers between 1 and 6. Announce the result: "You rolled 5 and 3." Repeat the roll/print process ten times.

5 CGI: Which is Less

Write a perl CGI program that displays two blanks and invites the user to type in a number into each blank. When the user submits the form, the same CGI program should run, compare the two numbers, tell which one is smaller (say "$little is less than $big" or "the numbers are the same"), and display two empty blanks as before. Do not "use CGI;".

Leave the space below blank