

# Final Exam CSCI 363 Fall 2001

## Open Book Section

13 December, 2001

Name: \_\_\_\_\_

This exam has a closed book and an open book section. Until you have turned in the closed book section, you are not allowed to use any resource materials: books, notes, or calculators.

Be sure to show your work in order to get full credit for the problem. When possible place your answers in the provided space. There are 7 questions for a total of 35 points on the open book section of the quiz.

This exam is to be turned in by 2:45 pm.

### **Problem 1 (4 points)**

Suppose an IP datagram of 1500 bytes, not including the IP header, is to be sent across a network with an MTU of 1000 bytes. How is the datagram fragmented for the trip? What fields are set in the fragments to aid its eventual reassembly?

### **Problem 2 (2 points)**

Suppose the 48-bit, or six-byte, pattern

00000000 01111100 01111101 01111110 01111111 11111111

is to be transmitted within the data section of a PPP packet. How are these six bytes transmitted?

### **Problem 3 (4 points)**

If 10111001 is being used as the generator  $G$  for a Cyclic Redundancy Check, what is the remainder  $R$  left after dividing  $G$  into the 12-bit message 101010101011?

#### Problem 4 (10 points)

Write a CGI script that is “called” from the browser by the following URL:

```
http://www.cs.unca.edu/~USERID/FinExam.cgi?COURSENUM
```

where *USERID* is your user id on the CSCI workstations and *COURSENUM* is a UNCA course number. Your CGI program should respond with a page that says:

```
You got an A in COURSENUM
```

Where *COURSENUM* has been replaced with the value provided in the URL request. For example, the request

```
http://www.cs.unca.edu/~USERID/FinExam.cgi?CSCI201
```

should result in the reply:

```
You got an A in CSCI201
```

You may use the programming language of your choice in this question. Do not try to do any error checking on the *COURSENUM* field. Just assume it is valid. Also, don't bother with showing details of the HTML tags sent with your reply. A simple programming comment like:

```
// Send the HTML tags
```

will do.

### **Problem 5 (7 points)**

Now, let's modify the CGI program written in the preceding problem so that it only replies with

```
You got an A in COURSENUM
```

on the first time you ask about any course. When you ask more than once, **about any course**, it always replies with

```
Didn't I already tell you about COURSENUM
```

Just to make things absolutely clear the following sequence of two URL requests:

```
http://www.cs.unca.edu/~USERID/FinExam.cgi?MATH191
```

```
http://www.cs.unca.edu/~USERID/FinExam.cgi?CSCI201
```

should result, in order, in the following two responses

```
You got an A in MATH191
```

```
Didn't I already tell you about MATH191
```

### **Subproblem 5a**

Suppose you are going to call your cookie `FECourse`. How would you install a cookie called `FECourse` in the remote browser?

### **Subproblem 5b**

How would you test if the remote browser already has an appropriate cookie called `FECourse`?

### **Subproblem 5c**

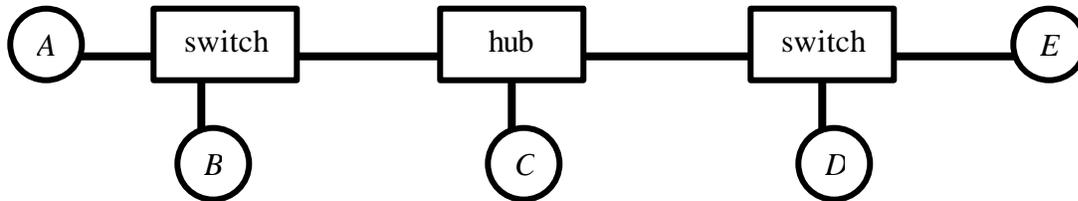
How would you retrieve the value of `FECourse`?

**Problem 6 (3 points)**

In which UNCA CSCI directory would you store your CGI script so it can be accessed as:  
`http://www.cs.unca.edu/~USERID/FinExam.cgi`

**Problem 7 (5 points)**

Consider the following LAN composed of five computers, two switches, and one hub:



Suppose that after a power failure, all computers, switches, and hubs are simultaneously powered-on in a state where both switches have “learned” nothing. If D transmits packet to C, this packet will be “seen” by all five computers.

The table below shows a sequence of packet transmissions that occur in order, one per second, on this LAN after the power failure. The first column is the source computer and the second column is the destination computer. The remaining five columns have places for checkmarks. You should check-off each computer that will “see” the transmitted packets. (The “na” means “not applicable”.)

source	destination	A	B	C	D	E
D	C	×	×	×	na	×
C	D			na		
A	D	na				
B	A		na			
C	B			na		
E	C					na