COS 109 Final Exam, Fall 2016
January 22, 2017
3 hours   180 points total

Please PRINT your name here ________________________________________________

Honor Pledge: “I pledge my honor that I have not violated the Honor Code during this examination.”

Please write the pledge in full and sign it:

This examination is open-book and open-notes:
- you may use course materials including the text book, course PowerPoints, your own notes, corrected problem sets and solutions, lab instructions, etc.
- you may use a computer or tablet to access these course materials but not for other purposes.
- you may use a calculator though questions requiring calculation will accept solutions that are approximations as we have done in class.

There are 180 points for the questions; use the point values for each question to allocate your time (one point per minute). There are choices in the second and third sections of the exam. No extra credit will be given for answering extra questions.

If you’re writing or calculating a lot on any question, you may be off on the wrong track.

Write your answers directly on these pages; use the back if necessary. In general, be brief, but if you need more space, attach extra pages and make sure your name is on every extra page. Please write legibly — I can't grade it if I can't read it.

Good luck.
1. (60 points, 3 each) Short Answers. Write your answer in the space provided.

(a) What is a pixel? How is the color of a pixel represented? What is a typical resolution of a laptop screen?

(b) The Princeton University libraries contain 11 million books. If the books were numbered starting at 1, how many bytes would be required to give each book a unique number?

(c) What is 419 in binary? In hexadecimal?

(d) Show the truth table for a 3 input XOR gate
(e) If a program runs in quadratic time and the input size is doubled, what do you expect will happen to the running time?

(f) What is the purpose of functions in a programming language?

(g) What is middleware?

(h) Give an example of an API that you have used.

(i) What is open source software? Give an example of an open source program.

(j) What happens when your computer runs out of RAM?
(k) What happens when your computer runs out of disk space?

(l) How many bytes are used to represent a MAC address? How many bytes are used to represent an IPv4 address?

(m) What is the difference between a cookie and a super cookie?

(n) Alan Turing appeared in this course twice. In what contexts did he appear?

(o) Why are there two keys in a public key cryptosystem?
(p) What are the steps in doing a Google search? You need not give details of how each step is done.

(q) What is net neutrality?

(r) We said that Twitter could disable Donald Trump’s account. Why could they do this?

(s) What is the basic unit of storage in a file system? Typically how much is stored in this unit?

(t) About how many bananas are there in a pound of bananas?
2. (80 points) Do 4 of the following 6 problems; each problem counts for 20 points.

(a) The following is the source for a web page with an embedded Javascript program that reads a collection of numbers, tests each to see if it is even and adds together all of the even numbers it sees. This value is then reported by writing its value (eventotal here). The program stops accepting input when the number 0 is input. The % operator computes the remainder when its first operand is divided by its second operand (in this example, num%2 returns 0 if num is even and 1 if num is even.

Modify the program so that it creates another variable named oddtotal in which it stores the sum of the odd numbers that come as input. Thus, it will compute two values -- eventotal and oddtotal -- where eventotal is (as in the code below), the sum of even numbers input and oddtotal is the sum of odd numbers input. Make sure that your code also writes out both of these values.

```html
<html>
<body>
P evens.html: add all even numbers
<p>
<script>
    var eventotal = 0  
    var num
    num = prompt("Enter new value, or 0 to end")
    num = parseInt(num)
    while (num != 0)  {
        if (num%2 == 0)
            eventotal = parseInt(eventotal) + parseInt(num)
        num = prompt("Enter new value, or 0 to end");
    }
    document.write(eventotal)
</script>
```
(b) For the following program written in the Toy language; study the problem to answer the questions below

```toy
MORE GET get an input from the user
STORE SUM1 store the accumulator value as SUM1
IFZERO END if the accumulator value is 0, go to END
ADD SUM2 add the value SUM2 to the accumulator
STORE SUM store the accumulator value as SUM
LOAD SUM1 load the value SUM1 into the accumulator
STORE SUM2 store the accumulator value as SUM2
GOTO MORE go to the instruction labeled MORE
END LOAD SUM load the value SUM into the accumulator
PRINT print the value in the accumulator
STOP end program execution
SUM1 0 initialize the value of SUM1 to 0
SUM2 0 initialize the value of SUM2 to 0
SUM  0 initialize the value of SUM  to 0
```

What does the program do? For each of the input sequences

- 7, 4, 2, 0
- 6, 12, 1, 1, 0
- 1, 0

what is the output of the program?

(c) For this problem you will deal with integers modulo 3. The value of an integer modulo 3 is the remainder when the number is divided by 3. So, for example, 7 modulo 3 is 1 and 9 modulo 3 is 0. When we count numbers modulo 3, we count as 1, 2, 0, 1, 2, 0, etc. Using this definition of integers
modulo 3, we will now design a finite state machine to determine when I have won a certain game of
solitaire.

In the solitaire game, I have a deck of cards consisting of kings (K) and queens (Q). The way the
game works is that I deal cards from the deck. As I do this, I keep track of the number of kings
modulo 3 and the number of queens modulo 3. The first time these two numbers are the same, the
game ends and I am declared the winner. The game continues until I have won.

For example, if the cards drawn are KKQKQQ I have won because there were 3 (= 0 modulo 3) kings
and 3 queens. Similarly, if the cards drawn are KKQKQKK because 5 (the number of kings) is 2
modulo 3 which equals the number of queens. In each case, this is the first time I will have won.

Your task is to build a state machine that takes as input a string of K’s and Q’s keeping track of the
number of kings modulo 3 and the number of queens modulo 3. Before drawing the machine, specify
what the states of the machine are, what the input is, … You can assume that the machine has one
final state (which you can label as SUCCESS) and that the machine continues to compute until it
reaches this state.

(d) Suppose I put $1 into a savings account that pays 4% interest and $1 into a savings account that pays
6% interest. After 72 years, how much money will be in each account?
(e) The message below has been encoded by a Caesar cipher of some length. What is the unencoded version of the message?

FRPSXWHU VFLHQFH LV D GHOLJKWIXO VXEMHFW WKDW L HQMRB

(f) For each of these running times, give an example of an algorithm that runs in that time. Also, explain what happens when the input size grows. You might tell what changes in the running time of the algorithm if the input size doubles or grows by some amount of … or you might tell what has to happen to the input size to make the running time double or grow by a certain amount.

(1) log N
(2) N
(3) N log N
(4) N²
(5) 2⁰
3. **(40 points) Do 4 of these 6 questions. Each question counts for 10 points.**

(a) Alice and Bob are students want to communicate in a secure fashion avoiding Eve, the evil eavesdropper.

(1) Alice finds Bob’s public key and sends him a message. Without getting into the mathematical details of how this works, describe the idea behind public key cryptosystems and why Alice and Bob can feel comfortable knowing that an eavesdropper will not be able to read the message.

(2) Bob receives Alice’s message and wants to be sure that it was actually Alice who sent the message. Explain why he would have this concern and how Alice can sign the message to assure him that she was the one who sent it?

(b) In this problem, you are asked to supply various details of how the internet works.

(1) To begin, briefly define the following terms

   (a) Peering points
   (b) Root servers
   (c) TCP/IP
   (d) ICANN

(2) How many root servers are there?

(3) Why are we not concerned that terrorists can take down all of the root servers since there are not so many?
(c) Consider the following facts:

1. It has been estimated that Google has enough bandwidth to move 4 gigabits of information per second.

2. On a recent day, it was estimated that the Google index of the web contained 4.8 billion webpages.

3. The average webpage contains 2 megabytes of information.

Based on these facts, if Google wanted to do a web crawl to gather the information on all websites once, how long would it take? Note that, in reality, when Google crawls the web, it doesn’t visit each website on each crawl.
(d) In this problem, you will show how the memory manager in an operating system works. To begin, suppose I have a machine with 1 GB of RAM. The operating system requires 256 MB to operate. This yields a diagram with 256 MB of RAM in use and the rest free. I have provided you with a picture of this situation at the start. For each of the following actions, modify the diagram on the next page to show what the RAM looks like.

1. I start Word which begins with 128 MB of RAM and Chrome which requires 256 MB of RAM
2. My Word document grows and I require an additional 128 MB of RAM
3. I start Chrome which requires 256 MB of RAM
4. My Word document grows again and so I require another 128 MB of RAM
5. I close Chrome
6. I start Firefox which requires 64 MB of RAM
(e) Part of a file system has the following structure with sizes of files given in bytes:

1. Directory D1 contains directories D2, D3 and D4
2. D2 contains directory D5 and the file ps1.html (12456 bytes).
3. Directory D5 contains file ps1old.html (12546 bytes)
4. Directory D3 contains files ps2.html (34523 bytes) and ps2ans.html (23424 bytes)
5. Directory D4 contains files ps3.html (42453 bytes) and ps3ans.html (72310 bytes)

Here is a diagram showing the directory tree

Each directory requires 1 block of storage in which are recorded the files (including directories) it contains and their sizes.

(i) Determine how many blocks (including directory blocks) are required to store everything under D2

(ii) How many blocks including directory blocks would the file system have to read to determine if D1/D2/D5/ps1old.html and D1/D2/ps1.html have the same contents?

(iii) How many blocks, including directory blocks, would the file system have read to list the names and sizes of all html files?
(f) Provide an estimate for the number of cups of coffee consumed in the dining hall of Rockefeller College during reading period. Explain the assumptions that you made in creating your estimate.