Name _____ Email _____ 3-digit ID _____

Systems Breadth Exam

1 February 2006

Please read **all** of the following instructions carefully.

This is a **closed-book**, **closed-notes**, written examination. You must answer *all* questions in the "Core" section. You will then answer the questions in the other two sections you have selected. Unless otherwise instructed, you must answer **all** of the questions in your chosen section. You have **three hours**.

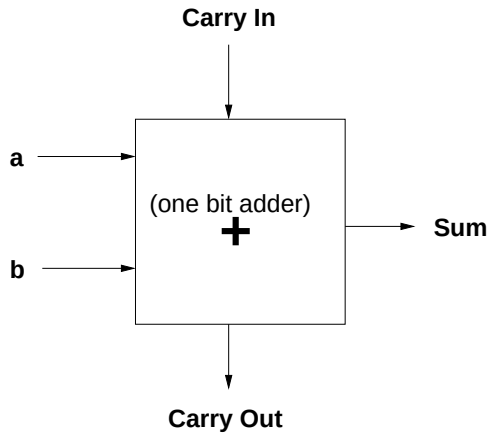
Do not write any answers on this paper. Write **only** your *name* and *email address* on this paper. Write your answers on the blank sheets provided. Use only one side of each answer sheet. Choose a **3-digit ID code** and write it at the top of this page, next to your name. The also **write it, along with the page number, at the top of each answer sheet.** In particular, **do not put your name on your answer sheets**.

1 Core

Answer all questions:

1. Give two reasons why computers usually store signed integers in two's complement format.

2. Suppose you were to build a one-bit adder circuit with the input and outputs shown in the following figure.



   (a) Show the *Sum* and the *CarryOut* output values that would result for the given input values:

Inputs			Outputs	
a	b	CarryIn	Sum	CarryOut
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

   (b) Draw a circuit using AND and OR gates that takes as input *a*, *b*, and *CarryIn* and outputs the *CarryOut* signal.

3. When a user removes a file, its blocks are generally put back on the free list. However, it is not clear what should be done to the contents of the blocks. One approach is to erase the contents of the blocks before placing them on the free list. Another approach is not to erase the blocks. Describe the advantages and disadvantages of the two approaches. Be sure to consider both security and performance factors in your answer.

4. Suppose two processes execute the following loops forever:

```
   Process 1                          Process 2
   ---------                          ---------
repeat forever {                   repeat forever {
    compute something                  compute something
      (non-blocking)                     (non-blocking)
}                                  }
```

We would like to have these two processes take turns using the CPU, but the OS does not use a preemptive scheduling algorithm. However, the OS does support semaphores that can be used to voluntarily give up the CPU. Insert the necessary semaphore system calls into the above code so that each process gives control of the CPU to the other process after executing the loop exactly 1 time.

2 Compilers

1. A b-palindrome is a sentence (or just word or a phrase) that spells the same backward as forwards after moving the first letter to the end. For example, the English word *grammar* is a b-palindrom.

   Design an LL(1) grammar G that generates lists of b-palindroms. A list consists of b-palindromes separated by commas (’,’). A comma may be succeded with a space. For example, *assess, banana, dresser, grammar, potato, revive, uneven, voodoo* is a list of b-palindroms.

   Use `a, b, c` plus comma (’,’) and space as the terminal alphabet. Show the parse tree for the following list: `abb, bcac`.

2. Describe the general structure of recursive descent (predictive) parsers for CFG. In your description use the concept of FIRST and FOLLOW sets.

   Give a simple example to illustrate. (You may use the grammar G from the above question if you wish.)

3 Databases

1. State at least three identities involving relationships between the selection operation σ, the projection operation π and the natural join operation ($\bowtie$). **At least one identity must involve natural join.**

2. Prove at least one of the identities you stated above; the one you prove must involve natural join.

3. Using no more than half a page, describe how these identities can be used to simplify query processing in SQL.

4 Distributed Operating Systems

1. Write a monitor-based solution for the following readers-writers problem:

   > *At most 10 readers can be in the critical section at any given time. Writers have priority over readers, i.e., a waiting or arriving reader gains access to the critical section only when there are no writers in the system.*

2. Assume a hypothetical distributed system where clocks at every process are perfectly synchronized. Describe, in enough detail to tell that it is correct, an algorithm for recording a consistent global snapshot of a distributed computation consisting of N processes in such a distributed system. Assume that communication channels are non-FIFO.

5 Networks

1. Suppose the polynomial used for a Cyclic Redundancy Check (CRC) is $C(x) = x^4 + x + 1$, and the original message we want to send is 10101110. What bits will be actually transmitted, if the CRC is used?

2. In this question, IP addresses and Network IDs are represented in dotted hexadecimal format, i.e., each byte of the address is represented by two hexadecimal digits. Suppose the forwarding table of an IP router using CIDR is as follows:

Network ID	Network Mask Length	Next Hop Address
D3.6D.02.0	23	A
D3.6D.08.0	21	B
D3.6D.C0.0	18	C
D3.6D.80.0	17	D
D3.64.0.0	14	E
D0.0.0.0	4	F
80.0.0.0	1	G

   Find the next hop address for the following IP addresses, using longest prefix matching. Show your work, i.e. how you find the answers.

   a. D3.6D.0A.02
   b. D3.66.C0.01
   c. D3.6D.31.2E
   d. D3.6D.A3.33
   e. D3.6D.03.15

3. The TCP adaptive retransmission needs to determine the *TimeOut* value for the timer. The original algorithm is as follows:

$$\begin{aligned} \textit{EstimatedRTT} &:= \alpha \times \textit{EstimatedRTT} + (1 - \alpha) \times \textit{SampleRTT}; \\ \textit{TimeOut} &:= 2 \times \textit{EstimatedRTT}; \end{aligned}$$

   The Jacobson/Karels algorithm calculates it in a different way:

$$\begin{aligned} \textit{Diff} &:= \textit{SampleRTT} - \textit{EstimatedRTT}; \\ \textit{EstimatedRTT} &:= \textit{EstimatedRTT} + (\delta \times \textit{Diff}); \\ \textit{Deviation} &:= \textit{Deviation} + \delta \times (|\textit{Diff}| - \textit{Deviation}); \\ \textit{TimeOut} &:= \mu \times \textit{EstimatedRTT} + \phi \times \textit{Deviation}; \end{aligned}$$

   Assume $\alpha = 0.8$, $\delta = 0.2$, $\mu = 1$, and $\phi = 4$. Compare these two algorithms and argue that the Jacobson/Karels algorithm is better than the original algorithm.

6 Programming Languages

1. Write a Lisp function that computes the nth Fibonnaci number. In other words, fill in the body for this code:

```
(function Fibonnaci (n) (
   -- body goes here
)
```

   What is the complexity of this program?

2. (a) Define the concept of a **first-class procedure**. Your definition should apply both to imperative and to functional languages.

   (b) First class procedures, in the presence of nested procedure declarations and deep binding, lead to what problem? Describe precisely what the problem is; just giving it a name is not sufficient.

   (c) Why does this problem not occur in C?

   (d) How can a language solve this problem but still allow any procedure to be first-class, allow procedures to nest, and employ deep binding?