Examinee: ________________ (number only)

1. (15 pts)
   Assume we wish to add dynamic arrays (i.e., arrays whose bounds can only be known
   at run time). Sketch a run time data structure for arrays, and discuss the structure
   of the code that needs to be generated for array accesses.

2. (5 pts)
   Consider the formal language \( \{a^n b^2, a^n b^{2n} | n \geq 0 \} \). Is there an LL(1) grammar for this
   language? Justify your answer.

3. (10 pts)
   SoftHead Software Inc., in Cary, NC, has decided to design programming language A
   where comments can be nested. For example
   
   
   /* This is an outer */ this is the inner portion */ portion */
   
   would be a legal comment in this language. As a compiler writer your task is review
   this design decision. Explain, briefly, with technical justification, your opinion of this
   design decision.

4. (5 pts)
   Switch statements in C/C++, and case statements in Pascal, are generally of the form:
   
   switch (e) {
     case c1: ..
     case c2: ...

     ...
   }

   where e and ci's are of discrete type. Dilbert's boss (who happens to be your boss
   too) would like to know why these languages do not permit e to be real/float type. As
   a compiler writer your task is to justify, to your boss, this design decision.

5. (15 pts)
   Write a simple procedure which brings out the difference between call by name, call by
   reference and call by need. Explain what the results of your program are under these
   three calling mechanisms.
6. A certain OS programming environment (environment here means a programming language and a runtime library) provides monitors and condition variables as its fundamental process synchronization primitives. In this system monitor semantics require a process to leave the monitor immediately after executing a signal on a condition variable.

Given the semaphore based solution to the bounded buffer problem shown below,

a. show how to recode this to an equivalent solution using the monitor/condition variable primitives, and
b. as an alternative approach, show how to build the semaphores in terms of monitors.
(Note: You must work both a, and b.)

**Bounded Buffer Solution:**

mutex is a binary semaphore initially $true
empty is a counting semaphore initially $N
full is a counting semaphore initially 0

**Structure of producer process:**

... 
produce an item
... 
wait(empty)
wait(mutex)
... 
add item to buffer
... 
signal(mutex)
signal(full)
...

**Structure of Consumer process:**

... 
wait(full)
wait(mutex)
...
remove item from buffer
...
signal(mutex)
signal(empty)
...
consume item
...
Examinee:____________________
(number only)

7. Assume you have a page reference string for a process with $f$ frames (initially all empty). The page reference string has length $l$ with $d$ distinct page numbers occurring in it. For any page reference algorithm,
   a. What is a lower bound on the number of page faults?
   b. What is an upper bound on the number of page faults?
   c. What is the maximum number of this process’s pages that can be in physical memory at any one time?

8. There are three ways to handle deadlocks: (i) prevention, (ii) avoidance, and (iii) detection and recovery. A system is currently unsafe. Which strategy is this system using? Justify your answer.

9. Three processes share data in critical sections in a distributed system that employs a fully distributed approach to mutual exclusion based on timestamps generated from logical clocks. Process $P_1$ requests entry with timestamp 10, and $P_3$ requests entry with its timestamp value at 8. $P_2$ does not require its critical section.
   a. Which process gains entry to its critical section first? Describe how this is determined.
   b. What if both timestamps are equal? That is, if both timestamps are equal, which process gains entry first, and how is this determined?