General Instructions

1. Use pencils to shade the circles in the scantron sheets - Do not tick or cross or strike.
2. You are required to return this question paper to the proctor after completing the test.
3. Use the blank sheet given to you for any calculations - Do not write or mark on the question paper.
4. Instructions on filling in the answer sheets
   a. Do not write or shade the fields corresponding to 'Name'.
   b. Write and shade you identification number.
   c. Use the special codes field to denote your section (Shade 0 under A for section 1 and shade 0 under B for section 2).
5. For answering in true or false format, use A as true and B as false.

1. Which of these are not shared between threads within a process
   i. Open files
   ii. Code
   iii. Stacks
   iv. Register context

   a) i, ii
   b) i, iii
   c) iii, iv ***
   d) i, iv

2. You want to implement a producer-consumer system. However, the buffer is unbounded (unlimited number of buffers), so it can hold any number of data elements. How many condition variables do you need?

   a) 0
   b) 1 ***
   c) 2
   d) 3

   Answer: The only condition variable that needs to be maintained is to track the empty buffers.

3. Which of the following will NOT guarantee that deadlock is avoided?
   a) Acquire all resources (locks) all at once, atomically
   b) Use locks sparingly ***
   c) Acquire resources (locks) in a fixed order
   d) Be willing to release a held lock if another lock you want is held, and then try the whole thing over again
4. Assuming a multi-threaded program has race condition. Which of the following would result in correct behavior of the program?
   a) Running the program in a single core machine
   b) System with non-preemptive scheduler
   c) Disabling interrupts when the application threads are running
   d) None of the above ***

Answer: (a) could still allow threads to be context switched on a single core. (b) and (c) does not prevent races in a multi-core machine.

5. Given the following events executing across two threads (The instructions of Thread1 are labeled 1-5 and instructions of Thread2 are labeled a-e):

<table>
<thead>
<tr>
<th>Thread 1</th>
<th>Thread 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. lock(&amp;mutex1);</td>
<td>a. lock(&amp;mutex2);</td>
</tr>
<tr>
<td>2. lock(&amp;mutex2);</td>
<td>b. lock(&amp;mutex1);</td>
</tr>
<tr>
<td>3. // critical section</td>
<td>c. // critical section</td>
</tr>
<tr>
<td>4. unlock(&amp;mutex2);</td>
<td>d. unlock(&amp;mutex1);</td>
</tr>
<tr>
<td>5. unlock(&amp;mutex1);</td>
<td>e. unlock(&amp;mutex2);</td>
</tr>
</tbody>
</table>

Sequence 1: 1->a->2
Sequence 2: 1->2->a
Sequence 3: a->1->b->2
Sequence 4: 1->2->3->4->a->b

Which of the above sequences will result in a deadlock?

a) 1 and 3 ***
b) 1, 3 and 4
c) 3 and 4
d) 3 ***
e) 1, 2 and 3

Answer: We accept (a) or (d) as an answer for this question.

6. Consider the following solution to the Dining Philosopher problem.

```c
acquire(int i) {
    if (i < 4) {
        sem_wait(chop[i]);
        sem_wait(chop[i+1]);
    } else {
        sem_wait(chop[0]);
        sem_wait(chop[4]);
    }
}
```
Which of the following options best applies?

a) This solves nothing. The code will deadlock.
b) This solves nothing. The code will livelock.
c) This solves the deadlock by preventing a circular wait. ***
d) This solves the deadlock by preempting the 4th philosopher.

7. Assume the following code snippet, where we have two semaphores, guard and notify.

<table>
<thead>
<tr>
<th>Thread 1</th>
<th>Thread 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>guard.wait();</td>
<td>guard.wait();</td>
</tr>
<tr>
<td>if(x &gt; 0)</td>
<td>x++;</td>
</tr>
<tr>
<td>notify.post();</td>
<td>notify.post();</td>
</tr>
<tr>
<td>guard.post();</td>
<td>guard.post();</td>
</tr>
<tr>
<td>notify.wait();</td>
<td></td>
</tr>
</tbody>
</table>

We want ‘guard’ to provide mutual exclusion between the two threads, and for ‘notify’ to provide a way for thread 2 to activate thread 1 when ‘x’ is greater than 0. What should the initial values of each of the two semaphores be? (Assume that ‘x’ is always positive or zero, and that there are only these two threads in the system).

What are the values to be used for initialization of the following semaphores?
sem_init(&guard, ___);
sem_init(&notify, ___);

a) <0, 0>
b) <1, -1>
c) <1, 0> ***
d) <0, -1>

8. Race conditions in C exist only because interrupts can happen between two lines of code in the C program.

a) True
b) False ***

9. Which synchronization primitive is best suited to service customers in a FIFO order?

a) CompareAndSwap
b) TestAndSet
b) FetchAndAdd ***
d) Loadlinked and Store conditional
10. Which of the following characteristics represent livelock better?

a) No activity and no progress  
b) No activity and progress  
c) Unwanted activity and no progress ***  
d) Unwanted activity and progress