CS 376	**Operating Systems**	Test 1
Name:	Winter 1999	Max Pts: 41

Important: This is an open book test. You can use any books, notes, or paper. *Do not log into the computer during the test.* Any calculations and rough work can be done on the back side of the test pages. You will loose five points for not writing your name.

1. [6 pt] What is the difference between an abstract resource and a physical resource. Give two examples of each.

2. [10 pt] Suppose a machine does not have the `test_and_set` instruction but has the indivisible `swap` instruction with the following semantics.

```
void swap ( bool& a, bool& b )
{
    bool tmp;
    tmp = a;
    a = b;
    b = tmp;
}
```

Can you use it to implement the `P` and `V` operations in semaphores? Provide the code for each.

3. [10 pt] Show the equivalence of message passing and semaphores by giving pseudocode to implement semaphores by message passing. You can use shared buffer area to hold mailboxes, each mailbox consisting of an array of message slots.

4. [10 pt] Does the following code meet the stated requirements for the solution to critical section problem. If yes, argue its correctness. If not, provide a counterexample and discuss the cause for failure. The code is meant for two processes; p1 is given while p2 can be assumed from it.

```
ext int turn = 0;                                    // Shared variable
extern bool flag[2] = { false, false };              // Shared flags

void process ( int i )          // i is the process number (0 or 1)
{
    while ( 1 )
    {
        compute();              // Section to perform computations

        // Entry section

        try:
            flag[i] = true;
            while ( flag[(i+1) % 2] )
            {
                if ( turn == i ) continue;
                flag[i] = false;
                while ( turn != i );
                goto try;
            }

        critical_section();

        // Exit section

        turn = ( i + 1 ) % 2;
        flag[i] = false;
    }
}
```

5. [5 pt] In the solution to the producer-consumer problem, it was argued that the order fo execution of `wait` in the producer is important. Does the same argument hold for the order of signal at the end of the loop in producer. The code is provided here for your convenience.

```
void producer ( void )
{
    do
        produce ( item );
        wait ( empty );       // empty is semaphore
        wait ( mutex );       // mutex is semaphore
        put ( item );
        signal ( mutex );
        signal ( full );
    while ( 1 );
}

void consumer ( void )
{
    do
        wait ( full );
        wait ( mutex );
        remove ( item );
        signal ( mutex );
        remove ( item );
        signal ( mutex );
        signal ( empty );
        consume ( item );
    while ( 1 );
}
```