1. (1 point) The final exam for this class is:
   A. a take home exam due by May 6, 1:00pm.
   B. **May 6th, 6:30-9:00pm, in MSC E208**
   C. May 6th, 6:30-9:00pm, in MSC W303 (usual classroom)
   D. May 6th, 8:30-11:00am, in MSC E208
   E. May 6th, 8:30-11:00am, in MSC W303 (usual classroom)

2. (2 points) How can you recursively drink a pitcher of root beer?
   A. (1) take one swallow, then (2) take another swallow.
   B. **(1) If there is one swallow of root beer left, drink it; otherwise (2) take one**
      swallow, then **drink the rest of the pitcher**.
   C. (1) Finish the pitcher in one enormous gulp, and (2) wish you hadn’t.
   D. (1) drink one pitcher, and (2) drink another pitcher.

3. (2 points) How do you recursively study for your CS170 final?
   A. Read a page in the book, and (2) read the next page until you come to the end of the book.
   B. **(1) If there is one page left, read it, and you are done; otherwise (2) study one page,**
      then **study the rest of the book**.
   C. Rip the book in two, and (2) study each half.
   D. Read all the pages in one horrible cram session the night before the final, and (2) forget
      everything the next day.
4. (10 points) Prof. Summet is trying to write a recursive method `sum` which computes the sum of the integers between the numbers `a` and `b` (inclusive). For example, a call to `sum(3,5)` should return 12 (calculated as 3+4+5). For each of the following functions, state whether Prof. Summet’s code correctly calculates the sum, assuming that `a` is less than or equal to `b`. If her code does not calculate the sum correctly, give the value that would be returned by a call to `sum(3,5)`. If the code would not return any value, explain why (briefly).

(a) ```java
   public static int sum(int a, int b) {
      if (a == b) {
         return 0;
      } else {
         return a + sum(a+1, b);
      }
   }
```  
   **Solution:** Incorrect. Sum is not inclusive of `b`. So `sum(3,5)` will only return 7 instead of 12.

(b) ```java
   public static int sum(int a, int b) {
      if (a == b) {
         return a;
      } else {
         return a + sum(a+1, b);
      }
   }
```  
   **Solution:** Correct.

(c) ```java
   public static int sum(int a, int b) {
      if (a == b) {
         return b;
      } else {
         return b + sum(a+1, b);
      }
   }
```  
   **Solution:** Incorrect. A call to `sum(3,5)` will return 15 instead of 12. In fact, it will calculate multiples of `b` instead of a sum.

(d) ```java
   public static int sum(int a, int b) {
      if (a == b) {
         return 1;
      } else {
         return a + sum(a+1, b);
      }
   }
```  
   **Solution:** Incorrect due to base case. A call to `sum(3,5)` will return 8 (3 + 4 + 1) instead of 12.

(e) ```java
   public static int sum(int a, int b) {
      if (a > b) {
         return 0;
      }
```
} else {
    return a + sum(a+1, b);
}
}

Solution: Correct.