CS303 (Spring 2008) - Assignment 1
Due: 01/23/2008

For this and all homework assignments, you are expected to solve the problems "pretty much" on your own. That means that asking your classmates (or the TA or instructor) for quick hints is ok, but the bulk of the work should be your own. The solutions of most of these problems are probably available online, some quite easily found. It is **absolutely not ok** to search for solutions online and try to pass them off as your own.

This assignment explicitly tries to make you practice proofs by induction. Some of these problems can probably be proved using other techniques, but those solutions are not acceptable for this particular assignment. Some facts will seem really obvious; the reason why you are asked to prove them is to practice the "mechanics" of an induction proof on questions where the difficulty of the problem does not distract you.

(1) Prove the following two identities for sums:

(a) $\sum_{i=1}^{n} i(i+1) = n(n+1)(n+2)/3$.

(b) $\sum_{i=1}^{n} ia^i = \frac{na^{n+2}-(n+1)a^{n+1}+a}{(a-1)^2}$.

(2) Remember that a graph G consists of a node set V and an edge set E, where each edge connects two nodes $u, v \in V$. We use $n = |V|$ to denote the number of nodes. We say that T is a *spanning tree* of G if it contains all the same nodes V, and a subset $F \subseteq E$ of the edges that forms a tree (i.e., (V, F) is connected, and does not have any cycles). A graph can have many different spanning trees; for instance, a triangle has 3 spanning trees, each obtained by leaving out a different edge.

Prove that for each n, there is a graph G with n nodes that has at least $(n-1)!$ different spanning trees.

(3) A popular puzzle problem is the following. You start with a chocolate bar of $m \times n$ rectangular pieces, which you can consider a two-dimensional grid of size $m \times n$. The goal is to break this bar into the individual mn pieces. The rules are that in each iteration, you can pick up one piece, and break it along a horizontal or vertical axis. You cannot break multiple pieces at once, and you cannot break a piece other than along one crease. The puzzle then asks you what is the minimum number of break operations you need to make in order to produce the individual pieces. If you like puzzles, you may want to think about this a little before reading on.

The solution is that never mind how you sequence the breaks, it will take *exactly* $mn-1$ breaks. So any way is as good as any other. Prove formally that for each sequence of breaks, it will take exactly $mn-1$ breaks. (Hint: think about what exactly is the invariant that is maintained by the process. Also, think carefully about what to do induction on; n or m probably won't work as induction variables.)

(4) Prove formally that the following program correctly computes the sum of the first n numbers in the array `a`. (Yes, you can assume that `a` is indeed an array of size at least n which has been properly initialized and all. And you needn't worry about number overflows.)

```
sum = 0;
for (i = 0; i < n; i ++)
    sum = sum + a[i];
```