MA2006 Ex wk 3

1. Prove the following:

   (i) $m * 0 = 0 * m$, $\forall\, m \in \mathbf{N}$;

   (ii) $1 * m = m$, $\forall\, m \in \mathbf{N}$;

   (iii) $s(m) * n = (m * n) + n$, $\forall\, m, n \in \mathbf{N}$
   (assuming $(m + n) + p = m + (n + p)$ and $m + n = n + m$ for all $\forall\, m, n, p \in \mathbf{N}$);

   (iv) $m * n = n * m$, $\forall\, m, n \in \mathbf{N}$.

2. State the ASSOCIATIVE law for * and the DISTRIBUTIVE law for * and +. Check that you have stated them correctly and then prove them.

N.B. Remember that, in many of the following questions, you know what the answers ought to be.

3. Evaluate the following:

   iszero(zero),
   succ(zero),
   iszero(*succ*(zero)),
   pred(*succ*(zero)),
   iszero(*pred*(*succ*(zero))),
   succ(*pred*(*succ*(*succ*(zero)))),
   pred(*succ*(*pred*(*succ*(zero))))
   and *pred*(zero).

4 Evaluate, using the recursive definitions:-

   add(3, 0), *add*(3, 1), *add*(3, 2),

   mult(3, 0), *mult*(3, 1) and *mult*(3, 2).

5. Suppose we wish to test the equality of two natural numbers by means of the function $eq : natno \times natno \to Boolean$.

   i) give a recursive definition of $eq(x, y)$
   and ii) evaluate $eq(2, 0)$, $eq(2, 1)$ and $eq(2, 2)$ from your definition.

6. Use your function *eq* to write a function *gte* : *natno* **x** *natno* $\rightarrow$ *Boolean* that will test if $x \geq y$ for all $x, y \in$ *natno*.

7. Implement a limited form of subtraction in *natno* by defining

   issub : *natno* **x** *natno* $\rightarrow$ *Boolean*

   and *sub* : *natno* **x** *natno* $\rightarrow$ *natno*

   so that *issub*(x, y) tests if y can be subtracted from x in *natno* and *sub*(x, y) gives x-y if it is possible in *natno*.

8. Define an ADT *modno* (modular numbers) designed to do modular arithmetic. Use the rules of *natno* but add two more functions

   last$\rightarrow$ *modno* and *islast*: *modno* $\rightarrow$ *Boolean*.

   You will need to alter the axioms relating to *pred* and *succ*.

9. Define addition and multiplication in *modno*.

10. Evaluate the following where *modno* = {0, 1, 2, 3, 4}:
    last, *islast*(4), *islast*(*pred*(0)), *add*(3, 1) and *add*(3, 2).

11. Define *indic* : *natno* $\rightarrow$ *natno* such that *indic*(x) returns 0 if x is 0 and 1 otherwise.