Types

Consider the following datatype and function declarations in Standard ML:

```ml
datatype tree = Leaf | Node of tree * tree;
fun iter x f Leaf = x
  | iter x f (Node(y,z)) = f(iter x f y)(iter x f z);
```

You are required to encode the datatype `tree` as a closed type \( \tau \) in the second-order lambda calculus, \( \lambda_2 \). Find a suitable type \( \tau \) and closed \( \lambda_2 \) terms in \( \beta \)-normal form, \( L, N, \) and \( I \) say, corresponding to \( \text{Leaf}, \text{Node} \) and \( \text{iter} \) respectively. You should demonstrate for your choices that

\[
\vdash L : \tau \\
\vdash N : \tau \rightarrow \tau \rightarrow \tau \\
\vdash I : \forall \alpha. \alpha \rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha) \rightarrow \tau \rightarrow \alpha
\]

are derivable typing assertions, and that \( I_\alpha x f L \) and \( I_\alpha x f (Nyz) \) are \( \beta \)-convertible to the \( \lambda_2 \) terms corresponding respectively to the right-hand sides of the clauses in the declaration of \( \text{iter} \). [14 marks]

Now add to the above Standard ML declarations the function declarations

```ml
fun rev Leaf = Leaf
  | rev (Node(y,z)) = Node(rev z, rev y);
fun div Leaf = Leaf
  | div (Node(y,z)) = div(Node(z,y));
```

Using \( I \), or otherwise, show that there is a closed \( \lambda_2 \) term of type \( \tau \rightarrow \tau \), \( R \) say, for which \( RL \) and \( R(Nyz) \) are \( \beta \)-convertible to the \( \lambda_2 \) terms corresponding respectively to the right-hand sides of the clauses in the declaration of \( \text{rev} \). Is there a closed \( \lambda_2 \) term \( D \) with similar properties for the declaration of \( \text{div} \)? [6 marks]