CMSC330 Spring 2015 Quiz #3

Name _____________________________________________

Discussion Time (circle one): 10am  11am  12pm  1pm  2pm  3pm

Discussion TA (circle one): Amelia  Casey  Chris  Mike  Elizabeth  Eric  Tommy

Instructions
• Do not start this test until you are told to do so!
• You have 15 minutes for this quiz.
• This is a closed book exam. No notes or other aids are allowed.
• Answer essay questions concisely in 2-3 sentences. Longer answers are not needed.
• For partial credit, show all of your work and clearly indicate your answers.
• Write neatly. Credit cannot be given for illegible answers.

1. (6 pts) Lambda Calculus
   Evaluate the following λ-expressions as much as possible. Show each alpha conversion and/or beta-reduction.
   Recall that application is left-associative, i.e., x y z is equivalent to (x y) z
   a.) (1 pt) (λf.λx.f) a
      => (λx.a)

   b.) (1 pt) (λx.x) (λx.x) b
      => (λx.x) b
      => b

   c.) (2 pts) (λf.λx.f (f x)) (λu.z) a b
      => (λx.(λu.z) ((λu.z) x)) a b
      => (λx.(λu.z) z) a b
      => (λx.z) a b
      => z b

   d.) (2 pts) (λf.λy.λx.x (y f)) y x f
      => (λaλb.λc.c (b a)) y x f  (* alpha conversion *)
      => (λb.λc.c (b y)) x f
      => (λc.c (x y)) f
      => f (x y)
2. (9 pts) Consider the OCaml type definition \(myTree\):

\[
type myTree =
  \text{Nil}
  |  \text{Leaf of int}
  |  \text{Node of myTree * myTree}
\]

A value of the type \(myTree\) is made up of \(\text{Nil}\) elements; \(\text{Leaf}\) elements, which have an associated integer value; and \(\text{Node}\) elements, which have two \(myTree\) children. Here are some example \(myTree\) values:

\[
\text{Node(Leaf 1, Leaf 2)}
\text{Node(Node(Leaf 1,Leaf 2), Leaf 3)}
\text{Node(Leaf 1, Node(Leaf 2, Node(Leaf 3, Nil)))}
\]

Write a function called \(\text{switch}\) that swaps the children of each \(\text{Node}\) in a \(myTree\). E.g.:

\[
\text{switch Node(Leaf 1, Leaf 2)} \Rightarrow \text{Node(Leaf 2, Leaf 1)} \\
\text{switch Node(Leaf 1, Node(Leaf 2, Nil))} \Rightarrow \text{Node(Node(Nil, Leaf 2), Leaf 1)} \\
\text{switch (Leaf 1)} \Rightarrow \text{Leaf 1}
\]

Your code must work in linear time (i.e. don’t cycle through a \(myTree\) multiple times). You are not allowed to use any OCaml library functions. You may use helper functions.

\[
\text{let rec switch n = match n with}
  \text{Nil} \rightarrow \text{Nil}
  \text{Leaf v} \rightarrow \text{Leaf v}
  \text{Node (left,right)} \rightarrow \text{Node (switch right,switch left)}
\]

\;