Name:
SSN :

CSE 428

Fall 1997

Midterm #2

19 November 1997

The exam consists of 5 problems on 5 pages, totaling 100 points. Read each question carefully and use your time judiciously.

Write your name/number on every page.

1. Give the *most general* types for each of the following function declarations. (20 pts)

(a)
```
fun noc (x,w) = w::x;
```

(b)
```
fun accumulate(f,a,nil) = a
   |accumulate(f,a,x::xs) = accumulate(f, f(a,x), xs);
```

(c)
```
fun tc 0 f x = x
   |tc n f x = f(tc (n-1) f x);
```

(d)
```
fun invert nil = nil
   |invert ((x,v)::L) = (v,x)::invert L;
```

2. Using the following function definitions, give the value produced by each of the expressions below. (20 pts)

```
fun noc (x,w) = w::x;

fun accumulate(f,a,nil) = a
   |accumulate(f,a,x::xs) = accumulate(f, f(a,x), xs);

fun map(f,nil) = nil
   |map(f,x::xs) = (f x)::map(f,xs);

fun tc 0 f x = x
   |tc n f x = f(tc (n-1) f x);

fun invert nil = nil
   |invert ((x,v)::L) = (v,x)::invert L;
```

(a) `accumulate(noc,nil,[1,2,3])`

(b) `tc 3 (fn n => n+1) 0`

(c) `map(invert,[[(1,2)],[(3,4),(4,5)]])`

(d) `map(map,[(fn n=>n+1,[1,2]), (fn n=>n+n,[1,2])])`

3. Define the following concrete datatypes. Your definitions should represent exactly the trees specified - nothing more and nothing less. (20 pts)

   (a) A datatype for I-trees in which an I-tree is a node containing an integer value and having zero or more descendants, all of which are also I-trees. (A leaf is just a node with zero successors.)

   (b) A datatype for 2-3 α-trees. A 2-3 α-tree is either a leaf containing an α-value, a node containing an α-value and having exactly 2 descendants, or a node containing an α-value and having exactly 3 descendants.

4. Recall the definition of Church numerals and booleans: (20 pts)

$$\begin{aligned} \overline{n} &= \texttt{fn f => fn x => f(f(...(f x)...))} \quad (n \text{ applications of } \texttt{f}) \\ \overline{true} &= \texttt{fn x => fn y => x} \\ \overline{false} &= \texttt{fn x => fn y => y} \end{aligned}$$

(a) Write the function `churchnot` which takes a church boolean and returns its complement. **Your solution may not use `tochurchbool`, `fromchurchbool`, or any other expression of type `bool`.**

(b) Write the function `even` which takes a church numeral and returns the churchboolean $\overline{true}$ if the numeral represents an even number, and returns $\overline{false}$ otherwise. **Your solution may not use `tochurch`, `fromchurch`, `tochurchbool`, `fromchurchbool`, or any other expression of type `bool` or `int`.**

5. For each of the following terms, give its β-normal form if it has one, or state that it has no β-normal form. (20 pts)

(a) $(\lambda x.\lambda y.x)\,(\lambda u.\lambda v.u)\,p\,(\lambda z.z)\,r$

(b) $(\lambda f.((\lambda x.f(x\,x))\,(\lambda x.f(x\,x))))\,(\lambda z.z)$

(c) $(\lambda x.\lambda y.\lambda z.(x\,z)(y\,z))\,(\lambda w.w)$

(d) $\lambda x.(x\,(\lambda y.y))$