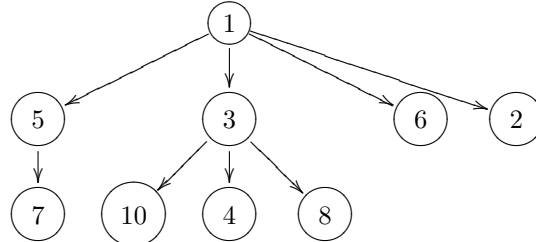


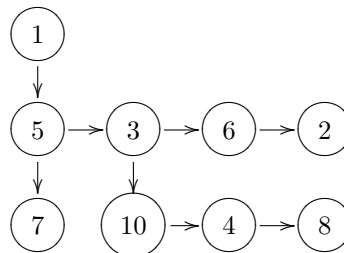
CEng242 Homework 1

Due 21st March 2004

In this homework you will deal with recursive data structures in ML. A n-ary tree can be represented as a binary tree. In this representation left link will be a link to first child and right link will be a link to next sibling. In other words children information will be a linked list of trees. Consider the following n-ary tree:



It can be represented as the binary tree:



Note that in this representation, root element has only one outgoing edge. For these two representations, you have following ML datatypes for integer trees:

```
datatype NaryTree = Node of (int*( NaryTree list));
datatype BinaryTree = Bin of (int*BinaryTree*BinaryTree) | Nil;
```

Write the following two functions to convert trees between these representations:

```
binton : BinaryTree →NaryTree
ntobin : NaryTree →BinaryTree
```

Sample run:

```
- val a=Node(1,[ Node(5,[Node (7,[])]),
                  Node(3,[ Node(10,[]), Node(4,[]), Node(8,[]) ]),
                  Node(6,[]),
                  Node(2,[])
                ]);
  val a = Node (1,[Node (#,#),Node (#,#),Node (#,#),Node (#,#)]) : NaryTree

- val b=Bin(1 , Bin (5, Bin (7,Nil,Nil),
                        Bin (3,Bin(10,Nil,
                                Bin(4,Nil, Bin(8,Nil,Nil) )
                              ),
                        Bin(6,Nil, Bin(2,Nil,Nil))
                      ),
            Nil);
  val b = Bin (1,Bin (5,Bin #,Bin #),Nil) : BinaryTree

- ntobin a;
  val it = Bin (1,Bin (5,Bin #,Bin #),Nil) : BinaryTree
  *** This has the same value with b***

- binton b;
  val it = Node (1,[Node (#,#),Node (#,#),Node (#,#),Node (#,#)]) : NaryTree
  *** This has the same value with a***
```

Follow the newsgroup for submission details. Cheaters will get 0 from all of the 6 homeworks.