Programming with Patterns (31050 and 32050)
Laboratory exercises: Week 8

Question 1 Modify the datatype declaration

```
datatype Btree a = Leaf of a | Node of Btree a and Btree a;;
```

so that `Leaf` is printed as `L` and `Node` is not printed at all. For example, `Node (Leaf 3) (Leaf 4)` prints as `(L 3)(L 4)`.

Question 2 Modify the `lessthan` relation on the binary trees above so that any node is bigger than any leaf.

Question 3 Declare a datatype of lambda-terms by

```
datatype Lambda =   Var of String
                  | Apply of Lambda and Lambda
                  | Lam of String and Lambda
```

Here the strings name the variables. Define a function

```
freeVars:Lambda -> List String
```

that determines the free variables of a lambda-term, so that `str` is not free in `Lam str e`.

Question 4 The function

```
let ext (free: a -> List String) = select(is _Int)
```

will find all the strings in a data structure. Extend `free` by a special case of `Lam` so that it can generalise `freeVars` of Question 3. Use this to modify the declaration of `Lambda`.

Question 5 Define a special case of the function `equal` for arrays, beginning

```
equal += | ((x as _array),(y as _array))  ->
```

Hint: Check out imperatives_core.bon in the standard prelude.