

ASSIGNMENT #4

Due Tuesday, February 17, 2004

1. Write a Lisp function (`replace atom1 atom2 list`) that takes two atoms, `atom1` and `atom2`, and a list as parameters and replaces all occurrences of `atom1` in the list with `atom2`, no matter how deeply the first atom is nested. Test this function using the following calls:

```
(replace 'nothing 'something '())
(replace 'a '1 '(a b r a c a d a b r a))
(replace 'b '0 '(no bs here))
(replace 'nest 'here '(nest (second nest level) (third (nest) level) (((big nest))))))
```

2. Write a Lisp function (`structtypeeq type1 type2`) that returns true if `type1` and `type2` are structurally equal according to the following rules:

- Basic types (e.g. `int`, `float`, etc.) are structurally equal if they are identical.
- Arrays are denoted by (`array lowerbound upperbound elementtype`) and two arrays are structurally equal if they have the same element type and number of elements.
- Record structures are denoted by (`record (var1 type1) (var2 type2) ... (varn typen)`) and two record structures are structurally equal if their elements have the same types.
- Pointers are denoted by (`pointer type`) and two pointer types are structurally equal if they point to structurally equal types.

Test this function using the following calls:

```
(structtypeeq '(array 1 10 int) '(array 0 9 int))
(structtypeeq '(record (a int) (b float)) '(record (x int) (y float)))
(structtypeeq '(record (a int) (b float)) '(record (a int) (b float) (c int)))
(structtypeeq '(record (n int) (trees (array 1 100 (pointer tree))))
 '(record (n int) (trees (array 1 100 (pointer tree))))
```

3. Write a Lisp function (`delete atom list`) that returns the list with all occurrences, no matter how deep, of `atom` deleted. The returned list cannot contain anything in place of the deleted atoms. Test this function using the following calls:

```
(delete 'nothing '())
(delete 'a '(a b r a c a d a b r a))
(delete 'b '(no bs here))
(delete 'nest '(nest (second nest level) (third (nest) level) (((big nest))))))
```