1999

p6q7
WFC
p13q7

Prolog Question 2. WFC

A trinary tree is constructed from 3-ary compound terms n(*a*,*b*,*c*) called nodes, where components *a*, *b* and *c* are either nodes or integers. Suppose integer components are restricted to the values 0 and 1.

(a) Write a Prolog program to return a list of all the 0's and a list of all the 1's in a given tree. For example, the goal enum(n(n(0,1, 0),1, 0), X, Y) should instantiate X to [0, 0, 0] and Y to [1,1]. The program will be expected to use difference lists.
[10 marks].

(b) A terminal node of the trinary tree is said to be of *odd parity* if the number of its 1 components is an odd number. For example, n(1,1,1) is of odd parity, and n(1,0,1) is not of odd parity. Write a Prolog program to count the number of terminal nodes in a tree that have odd parity. For example, the goal odd (n(n(0,1, 0),1, 0), X) should instantiate X to 1.
[10 marks].

Suggested Solution.

(a) This is similar in structure to the problem in Worksheet 29 in the notes.

```
enum(n(A, B, C), X1-X4, Y1-Y4) :-
        enum(A, X1-X2, Y1-Y2),
        enum(B, X2-X3, Y2-Y3),
        enum(C, X3-X4, Y3-Y4).
enum(0, [0|X]-X, Y-Y).
enum(1, X-X, [1|Y]-Y).
```

(b)

```
odd(n(A, B, C), Z) :- odd(A, An), odd(B, Bn), odd(C, Cn), Z is An+Bn+Cn.
odd(n(0,0,1), 1).
odd(n(0, 1,0), 1).
odd(n(1, 0, 0),1).
odd(n(1,1,1), 1).
odd(n(_, _, _), 0).
```

There is no doubt a tail recursive formulation as well.