

```

% ----- CM20019 -----
% Coursework 3 - due 14/12/06 - sample solution
% -----

% ----- part (1)
findFrontier(T, [T]) :- atom(T).
findFrontier(T, F) :- \+atom(T), T=..[_|Args],
    auxfindFrontier(Args, [], F).

auxfindFrontier([], F, F).
auxfindFrontier([Arg|Args], Acc, F) :-
    findFrontier(Arg, Flat),
    append(Acc, Flat, Accnext),
    auxfindFrontier(Args, Accnext, F).

% ----- part (2)
alphabet([a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]).

oneOrdered([]).
oneOrdered([X]) :- alphabet(A),
    member(X, A).
oneOrdered([X, Y|Xs]) :- alphabet(A),
    append(_, [X|Rest], A),
    (X=Y ; member(Y, Rest)),
    oneOrdered([Y|Xs]).

% ----- part (3)
makeBinTree(T1, T2) :-
    findFrontier(T1, F1), oneOrdered(F1), remdups(F1, [], F2),
    maketree(F2, T2).

maketree([X], X).
maketree([X|Xs], t(X, T2)) :-
    maketree(Xs, T2).

remdups([], F2, F2).
remdups([F|Fs], Acc, F2) :-
    \+member(F, Acc),
    append(Acc, [F], Accnext),
    remdups(Fs, Accnext, F2).
remdups([F|Fs], Acc, F2) :-
    member(F, Acc),
    remdups(Fs, Acc, F2).

%+++++
%+++++
%+++++
%----- TESTING for part (1)
%
%?- findFrontier(a,F).
%
%F = [a] ;
%
%No
%
%-----
%?- findFrontier(f(a,f(b,a)),F).
%
%F = [a, b, a] ;
%
%No
%
%-----
%?- findFrontier(f(g(a), g(f(b,g(c)))) ,F).
%
%F = [a, b, c] ;
%

```

```

%No
%-----
%?- findFrontier(f(g(h(b,c,f(h(d,l,g(m)),a,e)))) ,F).
%
%F = [b, c, d, l, m, a, e] ;
%
%No
%-----
%+++++
%----- TESTING for part (2)
%
%?- oneOrdered([]).
%
%Yes
%-----
%?- oneOrdered([A]).
%
%A = a ;
%
%A = b ;
%
%A = c ;
%
%A = d ;
%
%A = e ;
%
%A = f ;
%
%A = g ;
%
%A = h ;
%
%A = i ;
%
%A = j ;
%
%A = k ;
%
%A = l ;
%
%A = m ;
%
%A = n ;
%
%A = o ;
%
%A = p ;
%
%A = q ;
%
%A = r ;
%
%A = s ;
%
%A = t ;
%
%A = u ;
%
%A = v ;
%
%A = w ;
%
%A = x ;
%
%A = y ;
%

```

```
%A = z ;
%
%No
%-----
%?- oneOrdered([a]).
%
%Yes
%-----
%?- oneOrdered([abc, def, ghi, jkl, mno] ).
%
%No
%-----
%?- oneOrdered([a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,z,y] ).
%
%No
%-----
%?- oneOrdered([a,B,c,d,E] ).
%
%B = a
%E = d ;
%
%B = a
%E = e ;
%
%B = a
%E = f ;
%
%B = a
%E = g ;
%
%B = a
%E = h ;
%
%B = a
%E = i ;
%
%B = a
%E = j ;
%
%B = a
%E = k ;
%
%B = a
%E = l ;
%
%B = a
%E = m ;
%
%B = a
%E = n ;
%
%B = a
%E = o ;
%
%B = a
%E = p ;
%
%B = a
%E = q ;
%
%B = a
%E = r ;
%
%B = a
%E = s ;
%
%B = a
```

%E = t ;
%
%B = a
%E = u ;
%
%B = a
%E = v ;
%
%B = a
%E = w ;
%
%B = a
%E = x ;
%
%B = a
%E = y ;
%
%B = a
%E = z ;
%
%B = b
%E = d ;
%
%B = b
%E = e ;
%
%B = b
%E = f ;
%
%B = b
%E = g ;
%
%B = b
%E = h ;
%
%B = b
%E = i ;
%
%B = b
%E = j ;
%
%B = b
%E = k ;
%
%B = b
%E = l ;
%
%B = b
%E = m ;
%
%B = b
%E = n ;
%
%B = b
%E = o ;
%
%B = b
%E = p ;
%
%B = b
%E = q ;
%
%B = b
%E = r ;
%
%B = b
%E = s ;

%
%B = b
%E = t ;
%
%B = b
%E = u ;
%
%B = b
%E = v ;
%
%B = b
%E = w ;
%
%B = b
%E = x ;
%
%B = b
%E = y ;
%
%B = b
%E = z ;
%
%B = c
%E = d ;
%
%B = c
%E = e ;
%
%B = c
%E = f ;
%
%B = c
%E = g ;
%
%B = c
%E = h ;
%
%B = c
%E = i ;
%
%B = c
%E = j ;
%
%B = c
%E = k ;
%
%B = c
%E = l ;
%
%B = c
%E = m ;
%
%B = c
%E = n ;
%
%B = c
%E = o ;
%
%B = c
%E = p ;
%
%B = c
%E = q ;
%
%B = c
%E = r ;
%

```

%B = c
%E = s ;
%
%B = c
%E = t ;
%
%B = c
%E = u ;
%
%B = c
%E = v ;
%
%B = c
%E = w ;
%
%B = c
%E = x ;
%
%B = c
%E = y ;
%
%B = c
%E = z ;
%
%No
%-----
%?- oneOrdered([a,a,a,b,b,r,s,t,t,v] ).
%
%Yes
%-----
%?- oneOrdered([aB,zY] ).
%
%No
%%++++++++++++++++++++++++++++++++++++++++++++++++++++
%----- TESTING for part (3)
%
%?- makeBinTree(f(a,g(h(a,a,f(a,a)))) ,T2).
%
%T2 = a ;
%
%No
%-----
%?- makeBinTree(f(a,g(h(b,c,f(d,e)))) ,T2).
%
%T2 = t(a, t(b, t(c, t(d, e)))) ;
%
%No
%-----

```