Problem 2

Below is where you would want to substitute in cuts.

```
substitute(Old,New,Old,New) :- !.
substitute(Old,New,Term,Term) :-  constant(Term), !, Term \= Old.

substitute(Old,New,Term,Term1) :-
        compound(Term), !,
        functor(Term,F,N),
        functor(Term1,F,N),
        substitute(N,Old,New,Term,Term1).

substitute(N,Old,New,Term,Term1) :-
         N > 0, !,
        arg(N,Term,Arg),
        substitute(Old,New,Arg,Arg1),
        arg(N,Term1,Arg1),
         N1 is N-1,
        substitute(N1,Old,New,Term,Term1).

substitute(0,Old,New,Term,Term1) :- !.
```

I think that a cut-fail combination would be useful due to the fact that you do not want to backtrack to the old term when substituting in a new term, to begin with. Also, the explicit conditions can be omitted because if you use a cut fail combination some of the conditions would not be used. This is because the cut fail would stop before those conditions are met.