Name ______________________

Expert Systems

EE 4785

Midterm Examination

16 March 1994, 10:40-11:30

Problem 1	________	(40 pts)
Problem 2	________	(40 pts)
Problem 3	________	(20 pts)
Exam Total	________	(100 pts)

Alias ______________________

Good Luck!

Problem 1: Write a Prolog procedure to modify a list, as follows. The Prolog procedure `modify(List,ModList)` converts the list, `List`, which contains atoms and `next(X)` structures into a list, `ModList`. `ModList` is the same as list, except wherever there is a `next(X)` structure in `List`, `ModList` contains the atom which is `X` positions to the right. The quantity `X` is a positive integer. If the list ends before the `X`'th position, the modified list should contain `next(X)` in that position. If the list contains a `next(Y)` in the `X`'th position, the atom which will replace `next(Y)` should also replace `next(X)`. For example,

```
| ?- modify([a,b,next(2),c,d,e],X).

X = [a,b,d,c,d,e]

Yes.

| ?- modify([a,b,next(2),next(3),d,e,f],X).

X = [a,b,d,f,d,e,f]

Yes.

| ?- modify([a,next(1),next(2),next(3),d,e],X).

X = [a,d,d,next(3),d,e]

Yes.
```

Problem 2: Write a Prolog procedure which will convert a rule for the shellPropBC shell into a Prolog clause. For example, the following goal would succeed:

```
| ?-  convert(rule1:if a and b then c,X).

X = (c:-a,b)

Yes.
```

- Call your procedure `convert`.
- Be sure that it can handle antecedents of arbitrary complexity, including those which contain the operators `or` and `not`.
- For full credit on this problem, when your procedure `convert(R,P)` succeeds, P must be bound to a structure corresponding to a Prolog clause, not simply a string.
- Note that your procedure only has to convert from one to the other; there is no need to check if the various assertions are true or askable.
- Note that `not` is used with both Prolog clauses and expert system rules.

The following operator declarations, from page 9 of the C-Prolog manual and shellPropBC, may be helpful[1]:

```
:-op(1200,xfx,:-), op(1100,xfy,;), op(1000,xfy,','), op(900,xfx,:),
op(850,fx,if), op(800,xfx,then), op(750,xfy,or), op(700,xfy,and), op(650,fy,not).
```

[1] The alert student may have noticed that `not` is defined with a precedence of 900 in the C-Prolog manual. Use the lower precedence given here.

Problem 3: Evaluate how an expert system might be used for each of the tasks below. Explain what information the expert system would get from the user and what conclusions the expert system would give to the user. Indicate the suitability of the expert system, as you described it, as being either: -2, definitely not suitable; -1, probably not suitable; 0, neutral; 1, probably suitable; and 2 definitely suitable. Explain your choice. (Most of the credit will be given for the explanation.)

Part 1: Finding the best diet for a person of average health.

Part 2: Choosing a major for an undecided freshman.

Part 3: Determining the reason for a malfunction in the space shuttle. (The malfunction occurs before launch. The launch is delayed, but no one is hurt.)