CS2LAP: Logic and Prolog 2000/2001	**Handed out: 12/10/00**
Coursework 2	**Due date: 25/10/00**

Important note. Answers to questions 2, 3 and 4 have to be submitted electronically. Put all programs into a single file called `cw2.pl` located in a directory called `cs2lap` under your home directory. The file will be collected automatically at midnight on the due date. If you have any doubts whether you understand and are able to follow these instructions, make sure you ask well *before* the deadline.

Question 1. Give natural deduction proofs to show that the following two formulas are tautologies.

(a) $(P \Rightarrow (Q \Rightarrow R)) \Rightarrow ((P \wedge Q) \Rightarrow R)$

(b) $(P \wedge \neg P) \Rightarrow Q$

Note: Use *only* the following rules and indicate what rule you are applying to which formula(s) in every single proof step.

$\wedge$-introduction ($\wedge$I) $\dfrac{A, B}{A \wedge B}$	$\wedge$-elimination ($\wedge$E) $\dfrac{A \wedge B}{A} \quad \dfrac{A \wedge B}{B}$
$\vee$-introduction ($\vee$I) $\dfrac{A}{A \vee B} \quad \dfrac{B}{A \vee B}$	$\vee$-elimination ($\vee$E) $\dfrac{A \vee B, A \Rightarrow C, B \Rightarrow C}{C}$
$\Rightarrow$-introduction ($\Rightarrow$I) if $\dfrac{C_1, \ldots, C_k, A}{B}$ then $\dfrac{C_1, \ldots, C_k}{A \Rightarrow B}$	modus ponens ($\Rightarrow$E) $\dfrac{A, A \Rightarrow B}{B}$
$\neg$-introduction ($\neg$I) $\dfrac{A \Rightarrow B, A \Rightarrow \neg B}{\neg A}$	$\neg$-elimination ($\neg$E) $\dfrac{\neg A \Rightarrow B, \neg A \Rightarrow \neg B}{A}$

Marks: 20 (10 each)

Question 2. [**Electronic submission**] Write a Prolog predicate `reverse_list/2` that works like the built-in predicate `reverse/2` (without using `reverse/2`). Example:

```
?- reverse_list( [tiger, lion, elephant, monkey], List).
List = [monkey, elephant, lion, tiger]
Yes
```

Marks: 10

Question 3. **[Electronic submission]** Implement a Prolog predicate called `remove_duplicates/2` that removes all duplicate elements from a list given in the first argument and returns the result in the second argument position. Example:

```
?- remove_duplicates( [a, b, a, c, d, d], List).
List = [b, a, c, d]
Yes
```

Marks: 10

Question 4. **[Electronic submission]** Without using numbers or the `length/2`-predicate, define a Prolog predicate `odd_list/1` that succeeds if and only if the argument given is a list with an odd number of elements. Examples:

```
?- odd_list( [ceci, n, est, pas, une, liste]).
No

?- odd_list( [zebra, monkey, crocodile]).
Yes

?- odd_list( zebra).
No
```

Marks: 10