UNIVERSITY OF OTTAWA
FACULTY OF ENGINEERING
SCHOOL OF IT AND ENGINEERING

CSI 2165

Midterm
October 31, 2005
8-10 am

Examiner:
Dr. Diana Inkpen

Name	
Student Number	

Total marks: **35**
Duration: **110** minutes
Total Number of pages**:** **8**

Important Regulations:
1. **No calculators are allowed.**
2. **A student identification card (or another photo ID) is required.**
3. **An attendance sheet shall be circulated and should be signed by each student.**
4. **Please answer all questions on this paper, in the indicated spaces.**

A. [5 marks]
Which of the following are syntactically correct Prolog objects? If yes, identify the types of object they are (atom, number, variable, structure). If not, use a "No" as an answer.

	Term	Type
1.	y	Atom
2.	_v12	Variable
3.	12.5ab	No
4.	[1, [2], 3]	Structure (list)
5.	country(A(b), c)	No

B. [10 marks]

Show the variable instantiations (the values of X and Y) if the matching between the term in the first column and the term in the second column succeeds. Use "No" otherwise.

	Term1	Term2	Instantiations
Example	[1, 2, 3]	[X | Y]	X = 1 Y = [2, 3]
1.	[1| [2,3]]	[X | Y]	X = 1 Y = [2, 3]
2.	[1, 2 | 3]	[X | Y]	X = 1 Y = [2, 3]
3.	[[a], a]	[X | X]	X = [a]
4.	[[a, b], c]	[X | Y]	X = [a, b] Y = [c]
5.	f(g(Y), a)	f(g([a]),X)	X = a Y = [a]
6.	f(X, g(Y))	f([a], g([b, c | X]))	X = [a] Y = [b, c, a]
7.	[a, Y]	[X | X]	X = No Y = No
8.	[[b, c], a]	[X, Y]	X = [b, c] Y = a
9.	f(X,Y)	[a, b]	X = No Y = No
10.	[a, b, c]	[f(X), c, d]	X = No

C. [10 marks]

Read the following programs and provide the answer to the query (only the first answer that SWI-Prolog would produce).

1.
 	
Program
	`prog1([]).`
	
	`prog1([H|T]):- not(is_list(H)),`
	`member(H,[1,2,3,4]),`
	`prog1(T).`
	
	`prog1([H|T]):- is_list(H), prog1(H), prog1(T).`

Query	`?- prog1([1, [[2, [5]], 3]]).`
Answer	`No`

2.
 	
Program
	`prog2(X,[X,_,_]).`
	
	`prog2(X, [_|T1]):- prog2(X, T1).`

Query	`?- prog2(X,[a,b,c,d]).`
Answer	`X = b`

3. Program	`prog3([],[]).` `prog3([X|T], [X,X,X|T1]):- prog3(T, T1).`
Query	`?- prog3([1,2],R).`
Answer	`R = [1, 1, 1, 2, 2, 2]`

4.
Program
	`prog4(_,_,[],[]).`
	`prog4(X, Y, [X|T], [Y|T1]):- prog4(X, Y, T, T1).`
	`prog4(X, Y, [Z|T], [Z|T1]) :- X \= Z,`
	`prog4(X, Y, T, T1).`

Query	`?- prog4(b, 1, [a, b, b, c, b], R).`
Answer	`R = [a, 1, 1, c, 1]`

5. Program	prog5([],[]). prog5([X],[X]). prog5([F,_|T], [F|T1]):- prog5(T, T1).
Query	?- prog5([1, 2, 3, 4, 5], R).
Answer	R = [1, 3, 5]

C. [4 marks]

Given the following two versions of the predicate `del_one`, fill the table below with the SWI-Prolog's response to the corresponding query. Note that you have to include all answers by assuming you hit";" instead of "ENTER". (If the first and second arguments are given, the predicate deletes the first occurrence of the element given as first argument in the list given as second argument, in order to produce the resulting list in the third argument).

```
del_one_A(X, [X|T], T).
del_one_A(X, [Y|T], [Y|T1]) :-  X\=Y, del_one_A(X, T, T1).
```

```
del_one_B(X, [Y|T], [Y|T1]) :-  X\=Y, del_one_B(X, T, T1).
del_one_B(X, [X|T], T).
```

Query	`?- del_one_A(a,R,[b]).`	`?- del_one_B(a,R,[b]).`

Answers	`R = [a, b] ;`
	`R = [b, a] ;`
	`No`
		`R = [b, a] ;`
		`R = [a, b] ;`
		`No`

D. [2.5 marks]

Write a predicate that adds an element to the end of a list. The first argument is the element to be added, the second argument is a list (assume it is non-nested), and the third argument is the resulting list.

```
?- add(a, [b,c,d], R).
R = [b,c,d,a]
```

Solution:

```
add(X, [], [X]).
add(X, [H|T], [H|T1]) :-     add(X, T, T1).
```

E. [3.5 marks]

Write a predicate named `double` that multiplies by 2 all the occurrences of an element from any level of a nested list that contains only numbers. The first argument is the input list, and the second one is the result. Example:

```
?- double([1, 2, 3, [4, [1, 2], 5]], R).
R = [2, 4, 6, [8, [2, 4], 10]]
```

Solution:

```
double([],[]).
double([H|T],[H1|T1]):- not(is_list(H)), H1 is H*2, double(T, T1).
double([H|T], [H1|T1]):- is_list(H), double(H, H1), double(T, T1).
```