BROCK UNIVERSITY

Final exam
Course: COSC 2P93 Logic Programming
Date of exam: Thursday, April 24, 2003
Time of exam: 1900-2200
Instructor: B. Ross

# pages: 9
# students: 35
# hours: 3

NAME (print): ____________________________________

STUDENT NUMBER: _________________________________

There are 6 questions totalling 86 marks.
No aids are permitted. Use or possession of unauthorized materials will automatically result in a grade of zero for this examination.
Please answer all questions on the exam paper. Use the backs of pages if necessary.
Read the questions carefully. Keep written answers brief and to the point. Write legibly.
For programming questions, feel free to write as many auxiliary predicates as necessary.
Do **NOT** assume the existence of builtin predicates. Define predicates such as append/3, member/2, etc.
A grade of 40% is required on this exam to pass the course.

Question	**Total**	**Mark**
1	20	
2	10	
3	12	
4	10	
5	14	
6	20	
TOTAL:	**86**	

Question 1 [20] Define and briefly compare and contrast the following pairs of terms:

(a) assert(X) and retract(X)

(b) consult(F) and compile(F)

(c) red cuts and green cuts

(d) declarative Prolog and extralogical Prolog

(e) Prolog unification and Java assignment

Question 2 [10] Consider the following program:

```
boolean(t, t) ❶.
boolean(f, f).
boolean(and(A, B), Out) :-
      boolean(A, OutA),
      boolean(B, OutB), ❷
      and_table(OutA, OutB, Out) ❸.
boolean(not(A), Out) :-
          not_table(A, Out).
```

```
and_table(f, f, f).
and_table(f, t, f).
and_table(t, f, f).
and_table(t, t, t).

not_table(t, f).
not_table(f, t).
```

Given the query, "?- boolean(X, Y)", discuss what happens when...

(a) no cuts are used:

(b) cut is placed at position ❶:

(c) cut placed at position ❷:

(d) cut placed at position ❸:

(e) no cuts are used, but the query is, "?- one(boolean(X,Y))", using one/1 discussed in class:

Question 3 [12]

(a) [6] Write a predicate odd_calc/2. It takes a list of numbers, negates every second number, and sums the result. For example,

```
?- odd_calc([1,2,3,4,5], Ans).
Ans = 3       % 1-2+3-4+5 = 3
```

Question 3 (cont)

(b) [6] Rewrite odd_calc/2, but this time let it skip over non-numbers, moving them to another list argument. For example,

```
?- odd_calc([1,2, cat, 3, dog, 4, 5], Ans, Stuff).
Ans = 3
Stuff = [cat, dog]
```

Question 4 [10]

A palindrome is a string that reads identical front-to-back and back-to-front. Write a predicate palindrome/1. It succeeds if its list argument is a palindrome. For example,

```
?- palindrome([w,o,w,b,o,b,w,o,w]).
yes
?- palindrome([n,o,t,a,s,o,l,u,t,i,o,n]).
no
```

Question 5 [14]

Write a utility readList/1. It will read in a sequence of terms from the user, giving a prompt for each term to be read. The prompt consists of a counter, starting at 1, and incrementing for each new item read. Duplicate items are not permitted. As soon as the user enters a term that has already been read, a message is written, and that item is not used. It quits when the term ‘end’ is read. These terms are appended to a list in the order read. The list is returned as a result. For example,

```
?- readList(Ans).
Enter item 1: cat.
Enter item 2: dog.
Enter item 3: cat.
*** duplicate, ignored ***
Enter item 3: horse.
Enter item 4: end.
Ans = [cat, dog, horse]
```

Question 6 [20]

Write a predicate equalSplitter(L, A, B). It takes a list L of integers. It then divides the list into two subsets A and B, such that the sum of A and B are equal. Alternate solutions are returned upon backtracking. In addition, duplicate solutions and permutations of the same solution are not returned. For example,

```
?- equalSplitter([1,2,3,4], A, B).
A = [1,4],
B = [2,3] ? ;
no

?- equalSplitter([1,2,3,1,2,3,1,2,3], A, B).
A = [1,2,3,1,2],
B = [3,1,2,3] ? ;

A = [1,2,1,2,1,2],
B = [3,3,3] ? ;

A = [1,3,1,3,1],
B = [2,2,2,3] ? ;
no
```

Question 6 (cont)

***** student_mood(happy) :- exam(over), summer(here), !. *****