CS181: Programming Languages
Vladimir Vacic, Christos Koufogiannakis
University of California, Riverside

Lab Assignment 2

1. Implement the following Prolog rules:

a) length(L, N)	N is length of the list L
b) member(E, L)	E is an element of the list L
c) prefix(P, L)	P is a prefix of list L
d) suffix(S, L)	S is a suffix of list L
e) append(L1, L2, BigL)	BigL is the result of concatenating L1 and L2
f) sum(L, S)	S is the sum of all elements in L
g) product(L, P)	P is the product of all elements in L
h) split(L, N, L1, L2)	The list L1 contains the first N elements of the list L, the list L2 contains the remaining elements.

Note: some of the aforementioned rules may be already existing predicates in Prolog. Before deciding to name your rule in a certain way, you can use help(*something*) to check if *something* is already defined in Prolog.

Prefix: something attached to the front of a word to produce a derivative word. For example: *river* in *river*side.

Suffix: something attached to the end of a word to produce a derivative word. For example: *side* in river*side*.

You can take that the product of elements in an empty list is 1, as the base case.

2. Suppose that we have the following database:

```
(0).
f(1) :- !.
f(2).
```

Write the answers to the following queries:

a) ?- f(X).
b) ?- f(X), f(Y).
c) ?- f(X), !, f(Y).

3. Use cuts to define the if_then_else rule

if_then_else(C, S1, S2) If C is true then execute S1, else execute S2

Then use the if_then_else_rule to write a rule that finds the minimum of two numbers.

Hint: look at the min/max examples in the lab notes.

4. Use cuts to write the rule delete_first(E, L, A), which deletes only the first occurrence of element E from list L, producing answer A. For example:

```
?- delete_first(a, [b,a,n,a,n,a], A).

A = [b, n, a, n, a] ;

No
```

5. (**extra credit**) Implement the rule bubblesort(L, S) that sorts the list L using bubble sort and stores the result in a sorted list S.