Department of Computer Science and Engineering
University of California at San Diego

CSE 230
Winter 1998

Problem Set 6

DUE TUESDAY FEBRUARY 17 IN CLASS.

1. Explain the semantics of functions and macros in Common Lisp, as described in pp. 9–20, 82–94, 106-111, and 118–124 of *On Lisp: Advanced Techniques for Common Lisp* by Paul Graham (Prentice-Hall, 1994).

   The Lisp operators that you should explain include `#'`, `lambda`, and `funcall`, the comma `,` and backquote `` ` ``, and `destructuring-bind`. The issues that you should discuss include closures and lexical scoping, how macros have "two layers" of evaluation, variable capture and how to prevent it, and whether the difference between interpretation and compilation is visible to the programmer.

   In your answer, try to be precise but general, and concise but explicit. Use concepts discussed in class and correspondences with ML as much as possible.

2. In lazy ML, the sieve of Eratosthenes algorithm for generating the prime numbers can be implemented as follows.

```
- fun map f l = if l=[] then []
                        else (f (hd l))::(map f (tl l));
- fun remove pred l =
     if l=[] then []
     else if pred (hd l) then remove pred (tl l)
     else (hd l)::(remove pred (tl l));

- fun multiple p = (fn x => (x rem p) = 0);

- val rec series = 2 :: map (fn x => x+1) series;

- fun sieve (p::rest) = p::(sieve (remove (multiple p) rest));
```

   Make minimal but systematic changes to the code above to make it work in standard ML. Your changes should be systematic, that is they should follow a clearly visible pattern, because the point here is not to compute primes, but rather to illustrate how a lazy-evaluation style of programming is possible in ML.