

# CS 603: Programming Language Organization

---

Lecture 12

Spring 2004

Department of Computer Science

University of Alabama

Joel Jones

# Outline

- Questions
- Quick review of missed lecture
- $\mu$ -Scheme (cont.)
- Reading for next time

# Larger LISP Example

- Calculate prime numbers less than  $n$  using Sieve of Eratosthenes, where primes are found by removing all composite numbers from a list of positive integers.

# Sieve of Erasthathenes

```
(define divides (m n) (= (mod n m) 0))
(define seq (m n)
  (if (> m n) '()
      (cons m (seq (+ 1 m) n))))
(define remove-multiples (n L)
  (if (null? L) '()
      (if (divides n (car L))
          (remove-multiples n (cdr L))
          (cons (car L)
                (remove-multiples n (cdr L))))))
```

## Patterns:

- Helper functions
- return '() on null?
- cons head with recursive call

# Insertion Sort

- Insertion sort—given a list of  $n$  elements, sort the last  $n-1$  recursively, then insert the first in its proper position.

```
(define insert (x L)
  (if (null? L) (list1 x)
      (if (< x (car L))
          (cons x L)
          (cons (car L) (insert x (cdr L))))))

(define insertion-sort (L)
  (if (null? L) '()
      (insert (car L) (insertion-sort (cdr L)))))
```

Patterns:

- Helper functions
- return '() on null?
- user function combine head with recursive call