

---

**CS 451**  
**Programming Assignment #3**  
**Functional Programming**  
**Scheme**

Due 10/15/08, by 6:00am sharp!

**Introduction**

The goal of this assignment is to learn some scheme by writing a program to search for things. This assignment will also illustrate the passing of functions as parameters. You will write a search function (and some auxiliary functions). The search function takes a data structure, an element to search for, and four functions: `current-item`, `done`, `found`, and `next`. Function `current-item` current item at the front, top, whatever of the data structure. When passed the element and the data structure, `done` returns true iff the search should stop with failure. When passed the element and the current item of the data structure, `found` returns true iff the head of the data structure is the element being searched for. Finally, passed the element and the data structure, `next` returns the part of the data structure to be searched next.

**Example.** To search an unordered list, we might use the following:

```
(define unsorted-list '(6 7 4 45 7 76 3 67 7 63 19))
(define (second_empty first second) (equal? second ()))
(define (cdr_second a b) (cdr b))

(search unsorted-list '67 car second_empty equal? cdr_second)
```

After writing your search function, you should write the auxiliary functions and the calls to search a sorted list and a binary tree:

```
(define sorted-list '(2 6 10 14 55 65 78 99 102))
(define binary-search-tree
  '(10 (8 (4 () (6 () ()))) (9 () ())) (15 () (18 () ())) ))
```

**What to Hand In**

- (1) Email your scheme code. **NO ATTACHMENTS**
- (2) Email a script[man script(1)] of your program searching the list, the ordered list, and the binary tree. Be sure to include searches that are successful and searched that are unsuccessful.

**Extra Credit**

In addition to the value found, return a list that contains a \$ for every time next is called. For example,

```
(search unsorted-list '67 car second_empty equal? cdr_second)
produces
($ $ $ $ $ $ $ 67)
```