csc324 SOLUTION for Quiz #2 Monday 11 October 1999

Duration: 15 minutes

Aids allowed: None

Family Name: ____________________ **Lecturer:** Diane Horton

First Name: ____________________ **Tutor:** ____________________

Student #: ____________________

Grade: ____________

csc324 SOLUTION for Quiz #2 Tuesday 12 October 1999

1. Assume that the following two functions have been defined:

```
(defun freep (fcn x)              (defun bloop (fcn x)
    (apply fcn (cdr x))              (funcall fcn (cdr x))
)                                 )
```

Give the value of each of the following expressions, or write "error" if it would cause an error.

Expression	Value (or "error" if causes an error)
`(freep '+ '(1 3 5 2))`	10
`(bloop '+ '(1 3 5 2))`	Error: (3 5 2) is treated as the single argument to +
`(freep 'length '(1 3 5 2))`	Error: tries to evaluate (length 3 5 2)
`(bloop 'length '(1 3 5 2))`	3

Note: You did not have to explain the error.

2. Recall the tree representation from assignment 1, for example: `(1 (2 (3) (4) (5)) (6))`.
Complete the `height` function, but do not assume that the tree is binary. You may use Lisp's `max` function.
Hint: You may want your own helper function.

```
; Precondition: 'tree' is a well-formed tree, not necessarily binary.
(defun height (tree)
    (cond ( (null tree)  0 )
          ; I'm as tall as my tallest kid, plus 1.
          ( t   (+ (maxheight (cdr tree)) 1) )
    )
)

; Returns the height of the tallest tree in treelist.
; A helper is needed because we are recursing on a *list* of trees, not
; a single tree, which heigh expects.
(defun maxheight (treelist)
    (cond ( (null treelist) 0 )
          ( t  (max (height (car treelist))
                    (maxheight (cdr treelist))
               )
          )
    )
)
```

3. Recall the `prune` function from class:
`(prune list test)` returns a new list with only the elements of 'list' that pass the 'test'.
Precondition: 'test' is a predicate that can be applied to elements of 'list'.

(a) Assume `intlists` contains a list of lists of integers. Complete the following call so that it will keep only lists whose first element is smaller than its second:

```
(prune intlists
          (lambda (x) (< (car x) (cadr x)))
)
```

(b) Assume that `trees` contains a list of trees, and that the `height` function above has been written. Complete the following call so that it will keep only trees with height at least 5:

```
(prune trees
          (lambda (x) (> (height x) 5))
)
```