Principles of Programming, Spring 2006
Practice 7

Daejun Park, Heejong Lee
Programming Research Lab.@SNU

April 20, 2006

1. Below is a sequence of pairs of expression and result printed by the interpreter in response to each expression.

```
> 'foo
foo
> ''foo
'foo

> (define a 10)
> a
10
> 'a
a

> (a 1 2)
procedure application: expected procedure, given: 10;
arguments were: 1 2
> '(a 1 2)
(a 1 2)

> '(1 2)
(1 2)
> `(1 2)
(1 2)

> `(1 ,(+ 2 3) 4)
(1 5 4)

> (let ((x `y)) `(,x))
(y)

> `(list ,(+ 1 2) 4)
```

```
(list 3 4)
> (eval `(list ,(+ 1 2) 4))
(3 4)
)
```

What is the result printed by the interpreter in response to the following expressions? Assume that the sequence is to be evaluated in the order in which it is presented.

```
(define (or a b)
  `(let ((v ,a))
     (if v v ,b)))
(or '#t '#f)
(eval (or '#t '#f))
```

2. Define a procedure `spower` that takes a natural number as its argument, namely b, and returns the procedure, which takes a natural number as its argument, namely a, and computes a^b effectively. For example, the following expressions return same result.

```
((spower 10) 2)

(power 2 10)
```

Note that `spower` has better performance than `power` because of optimization.

3. Define a procedure `smap` that takes a list as its argument and returns the `map` procedure optimized for given list. For example, the following expressions return same result.

```
((smap (list -1 -2 -3)) abs)

(map abs (list -1 -2 -3))
```

Note that `smap` has better performance than `map` because of optimization.

4. Devise a self-generating program, which generates a copy of its own source text as its complete output. That is, when the program is run, it prints the strings containing an exact copy of its own source code. There is one example of the self-generating program in the following.

```
((lambda (x)
  (list x (list (quote quote) x)))
 (quote (lambda (x)
    (list x (list (quote quote) x)))))
```