

**CS61A, Fall 1997
Midterm #1
Professor Harvey**

Problem #1 (3 points)

What will Scheme print in response to the following expressions? If an expression produces an error message or runs forever without producing a result, you may just say "error"; you don't have to provide the exact text of the message. If the value of an expression is a procedure, just say "procedure"; you don't have to show the form in which Scheme prints procedures. Assume that no global variables have been defined before entering these expressions.

`(first (bf (first (bf (bf '(back in the ussr))))))`

`'(+ 6 7)`

`((lambda (x y) y) 8 3)`

`(+ '(3 4 5))`

`(let ((a +)`

`(* 3))`

`(a * *))`

`(if 6 7 8)`

Problem #2 (1 point)

`(define (triangle n)`

`(if (= n 0)`

`0`

`(+ n (triangle (- n 1))))`

When invoked, does the triangle procedure generate an iterative process or a recursive process?

Problem #3 (4 points)

Write a procedure named `diagonal` that takes a sentence as its argument. It should return a sentence containing the first letter of the first word of the sentence, the second letter of the second word, the third letter of the third word, and so on. You may assume that all the words are long enough.

For example:

`> (diagonal '(in my life))`

`(i y f)`

`>(diagonal '(your mother should know))`

`(y o o w)`

You may use this procedure to help out:

```
(define (item n stuff)
  (if (= n 1)
      (first stuff)
      (item (- n 1) (butfirst stuff))))
```

Problem #4 (4 points)

Write a procedure named `table` that takes three arguments: a function, a starting value, and an ending value. It should return a sentence with a table of values of that function for integers between the starting value and the ending value. For example:

```
>(table square 3 6)
(9 16 25 36)
```

```
> (table (lambda (x) (+ x 3)) 5 9)
(8 9 10 11 12)
```

Problem #5 (4 points)

Write a procedure named `echo` that takes a positive integer `n` as its argument. It should return a procedure that takes a word as argument, returning a sentence with `n` copies of the word. For example:

```
> (define triple (echo 3))
triple
```

```
>(triple 'hello)
(hello hello hello)
```

```
>((echo 5) 'baz)
(baz baz baz baz baz)
```

Problem #6 (3 points)

```
(define (foo n)
  (if (= n 0)
      '()
      (sentence (foo (- n 1))
                 n
                 (foo (- n 1)))))
```

(a) What is the value of `(foo 3)`?

(b) How many times is `foo` invoked while computing `(foo 3)`?

(c) Show how to change the base case of `foo` so that it returns the same answer, but with about half as many invocations.

**Posted by HKN (Electrical Engineering and Computer Science Honor Society)
University of California at Berkeley
If you have any questions about these online exams
please contact examfile@hkn.eecs.berkeley.edu.**