Name: KEY

CSSE304 Fall 202210 Final Exam Paper Part

B. [7 points] What does this scheme code print? (if it infinite loops, say that)

```
(let ((foo #f))
  (display "a")
  (call/cc (lambda (bar)
             (set! foo bar)
             (display "b")))
  (display "c")
  (call/cc (lambda (baz)
             (let ((oldfoo foo))
               (set! foo baz)
               (display "d")
               (oldfoo "e"))))
  (display "f"))
```

for first call/cc

r_1 = (lambda (bar) (set! foo bar) (display "b"))

k_1 = (escaper (λ (v) (begin (display "c") and other code

(r_1, k_1) => k_1 put in foo

for second call/cc

r_2 = (λ (baz) (let ((old foo foo)) (set! foo baz)

k_2 = (escaper ((λ (v) (display "f")) (old foo "e"))

for third call/cc

same as 2nd

Output

abcdcdf

A. [7 points] What does this scheme code evaluate to?

```
(define (fun lst proc)
  (if (null? lst)
      '()
      (let ((recurse (fun (cdr lst) proc)))
        (cons (proc (car lst)) recurse))))
(list (call/cc (lambda (foo) (fun '(1 2) foo))))
```

for first call/cc

$r_1 = (\lambda (foo)\ (fun\ '(1\ 2)\ foo))$

$k_1 = (escaper\ (\lambda (v)\ (list\ v)))$

we recurse first so

$(fun\ '(1\ 2)\ foo) \Rightarrow (fun\ '(2)\ foo) \Rightarrow (fun\ '()\ foo)$

last call returns to $(fun\ '(2)\ foo)$ and we eval

$(k_1\ 2) \Rightarrow (2)$

C. [6 points] What does this scheme code evaluate to?

```
(list (call/cc (call/cc (lambda (x) (lambda (y) (+ 3 (y 4))) ))))
```

inner call/cc

$r_1 = (\lambda (x)\ (\lambda (y)\ (+\ 3\ (y\ 4))))$

$k_1 = (escaper\ (\lambda (v)\ (list\ (call/cc\ v))))$

$(r_1\ k_1) \Rightarrow (\lambda (y)\ (+\ 3\ (y\ 4)))$

outer call/cc

$r_2 = (\lambda (y)\ (+\ 3\ (y\ 4)))$

$k_2 = (escaper\ (\lambda (v)\ (list\ v)))$

$(r_2\ k_2) \Rightarrow (k_2\ 4)$

(4)