Lab: Racket Expressions

Use *ONLY* first, rest, and cons.

```
(define a '())
(define b '(()()()))
(define c '(Bob (Jane)))
(define d '(((Bob)(Jane))))
(define e 'Jane)
(define f '((Bob)Jane))
```

Section A. Given the following definitions:

Variable	Definition
a	'( )
b	'(( ) ( ) ( ))
c	'(Bob (Jane))
d	'(((Bob) (Jane)))
e	'Jane
f	'((Bob) Jane)

Note:

You are defining **Jane** as an **atom**;

therefore **e** will be equal to **Jane**, *not* **(Jane)**

Evaluate the following expressions and write your answer in the appropriate space. If the expression cannot be evaluated, write "Cannot be evaluated".

(first a)	Cannot be evaluated.
(first b)	'()
(first c)	'Bob
(first d)	'((Bob) (Jane))
(first e)	Cannot be evaluated.
(rest f)	'(Jane)
(rest c)	'((Jane))
(rest d)	'()
(rest e)	Cannot be evaluated.
(cons a c)	'(() Bob (Jane))
(cons a d)	'(() ((Bob) (Jane)))
(cons a e)	Cannot be evaluated.
(cons f c)	'(((Bob) Jane) Bob (Jane))
(cons e d)	'(Jane ((Bob) (Jane)))
(cons a (rest c))	'(() (Jane))
(cons e (rest f))	'(Jane Jane)
(cons f (rest a))	Cannot be evaluated.
(cons a (rest b))	'(() () ())
(first (rest f))	'Jane
(first (rest (first d)))	'(Jane)
(first (rest (cons a f)))	'(Bob)
(rest (rest d))	Cannot be evaluated.
(first (rest f))	'Jane

Section B. Given the following definitions:

Variable	Definition
a	`(5 4 3 2 1)
b	`((5) ((4) (3) 3 ((2))))
c	`(5 (4 (3 (2 (1)))))
x	`(a b c (d) e f)
y	`((5) ((4 3) 2 (1)))

Write an expression that will output the following:

Using...	Output should be...	What is the expression?
List **a**	'(4 3 2 1)	(rest a)
	'(3 2 1)	(rest (rest a))
	4	(first(rest a))
List **b**	'((4) (3) 3 ((2)))	(first(rest b))
	'((3) 3 ((2)))	(rest (first (rest b)))
	'()	(rest (rest b))
List **c**	'(4 (3 (2 (1))))	(rest c)
	5	(first c)
	'((3 (2 (1))))	(rest (first (rest c)))
List **x** and **y**	'(a 5)	(cons (first x) (first y))
	'(b ((4 3) 2 (1)))	(cons (first (rest x)) (rest y))
	'((5) a b c (d) e f)	(cons (first y) x)