



# Racket Exercises

## تمرین اول (Racket Evaluation)

در این تمرین هدف این است که با نحوه تعریف متغیر و مقدار دهی آن ها در Racket آشنا شوید. به این منظور در کدی که داده شده است ، مقدار هر متغیر را مشخص کنید و چنانچه هر خطی از کد هنگام اجرا ، منجر به یک Error خواهد شد ، آن را مشخص کرده و دلیلش را بیان کنید . در ابتدا سعی کنید با استفاده از مطالبی که در کلاس آموخته اید ، کد را خوانده و مقادیر متغیر ها را مشخص کنید و در پایان با اجرای کد در بستر DrRacket پاسخ هایتان را ارزیابی کنید.

پاسخ ها بصورت زیر و در مقابل هر خط نوشته شود :

```
x |-> 7
y |-> 3
z: error -- attempt to divide a number by 0.
```

#lang racket

(define a 5)

(define b (\* a a))

(define c (+ (\* 2 a) (- b a)))

(define d (+ a b c))

(define e (+ a))

(define f (+))

(define g (\*))

(define h b)

(define i (if (< a 0) 7 (quotient 43 a)))

(define j (if (= a 0) 11 (remainder 43 a)))

(define k (/ i j))

(define l (< a b))

(define m (= b c))

(define n (< b c d))

(define o (+ a m))

(define p (< a m) )

(define q (if a b c))

```
(define r (if l m n))
```

```
(define s (if (if (< i j) m l)
```

```
    (if (> a c) (+ a c) (* a c))
```

```
    (if (< a b) (+ a b) (* a b))))
```

```
(define t (= (if (< b c)
```

```
    (* (remainder (- (* a b) (* 2 a)) c) (- (quotient d b)))
```

```
    (* (- (+ d (* (* a b) (quotient d a))))))
```

```
    (- (* (/ (- c a) (- a 1)) (quotient b (if (> a c) i j))))))
```

```
(define u (if (> i j) (< a 0) (< (< b c) a)))
```

```
(define v (if (<= i j) (< a 0) (< (< b c) a)))
```

## تمرین دوم (Big-step derivation)

برای عبارت زیر یک اشتقاق Big-Step در محیط `env5` با پیوند های  $a \mapsto 3, b \mapsto 6, c \mapsto 4$  بسازید.

```
(+ (if (< a b) (* a (+ b c)) (+ b #t))  
  (if (if (> a c) (< c b) (- c 4)) (* 2 b) a))
```

\* شما می توانید هر یک از دو فرمت conclusion-below-subderivations یا conclusions-above-subderivation را که ترجیح می دهید استفاده کنید ، در پایان فارغ از فرمت ، در داخل پرانتز مشخص کنید که کدام قانون در هر مرحله از اشتقاق استفاده می شود.

## تمرین سوم (Small-step derivation)

با همان شرایط ، یک اشتقاق Small-Step برای عبارت تمرین قبل بسازید.

۱. در هر مرحله از اشتقاق باید چپ ترین Redex را ارزیابی کرده و بین دو براکت `{ }` قرار دهد.

۲. نتیجه هر مرحله باید با نام قانونی که برای کاهش Redex از خط قبلی استفاده می شود در داخل براکت {} نوشته شود.

\*برای سهولت پاسخ دهی به تمارین ، می توانید به لینک زیر و فایلی که به همراه فایل تمرین بارگذاری شده است مراجعه کنید.

<https://docs.racket-lang.org/guide/index.html>

موفق و پیروز باشید