```
Ruckel Exercises"
                                                                                MFFTION : Words of the
Mobin Heilary
itulent No: 994421017
                                                                                            ترك اول ا
 41.
 #land tacket
                                                                                          N
                                             (driner (+ a))
(define a 5) al->5
                                               by erfor - + (+) lequites at least 7 argument, but only a
(defineb ( * a a)) b1->25
                                                  diven. The expression (+ a) thes to add the value of
( define ( (+ (* 2 a) (- ba))) (1-) 30
                                                  a to an entry list of alguments.
(define d (+ a b ()) d1-> 60
                fierer -- (+) requires at least tagunent, but onto given. The expression (+) thes
(define f (+1)
                 to add an empty list of arguments.
                 g: error - (1 requires at least 1 argument, but only e given. The expression() they to
define j(x))
                multiply an empty list of arguments.
 (define Ab) ht > 25
(define 1(if ((uc) 7 (quotient 43 a))) it-) 8
(define & (if (= a o) 17 (lemainder 43 al))
                                                JI-> 43 ( when a is not equal to 0, the remainder of 43
                                                      biviled by a is 43, and thus jis bound to 43)
( define k (/id))
    k 1-) = .186046 5776 2790697 (when also not equal to soils and is 43, sokis bound
(define I ((ab)) 11-) #1 (since 5 is loss than 25)
(define m (= b ()) m 1-) # f (since 25 is not equal to 30)
(define n (16 cl)) n 1-) # + (since 25 is less than 60)
( define o (+ a m)) o 1- > 30 ( since m ix false, the 14 explession in the definition of a evaluates to (+ a a)
                    which is equal to a, and a is 5, so o is bound to 30)
edefine P((am)) PI-) Af (Since rais bound togs, undsisned kis than 30)
(define 2 (if a b (1) 21->25 (since a is 5, b is 25, and ( is 30, the condition a evaluates to trace so 2 is bound
idefine Y (if I m n)) r 1 -> #f (since I is thue, and m is false, the if explosion exclusion exclusion to a, which is #1)
define s (illif ((is) m))
                               5 1-30 (since the condition (if ((id) m 1) cubhatesto m which is fake,
(if () a c) (+ a c) (* a c))
                                     the if explession evaluates to (if (lab) (+ ab) (* ab)), which
If ((+4) (+ab) (+ab)))
                                      15 (+ a b), which is 30)
```

رام وام فافادائه : مين عيري

```
(definet (= (it ((bc)
               (* (remainder (- (* a b) (+2 a))c) (-quotient d b) ))
               (* (-(+d(* (* ab) (quotient d a))))))
           (- (* (/ (- ca) (- a1)) (quotient b (if (> a c) is))))))
+: effet -- (-) requires at least 1 ununent, but only a given. The expression (- (* (/ (- ca) (-a1)) (quedient
 h (if ( > a c) i J)))) this to subtract on empty list of arguments from the value of the condition
 in the outermost define expression.
(define 4 (1f (7 1 d) (( a o) (( ( 4 b c) a)))
(16 c) evaluates to # to and thus u is bound to #f)
(define v (if ( l= i 5) ( l a 0) ( l a 0) ( l ( l b c) a) 1)
(1 b c) evaluates to #t, and thus V is bound to #t)
envs, (+ (if (( ab) ( + a (+ bc)) (+ b + t))
          (if (if ()ac) ((cb) (-c4)) (+2 b) a))
--> [B-PLUS]
taus, (if (1 a b) (x b (+ b()) (+ b #t))
       + [R-PLUS]
EAV5, (if (if ( ) a c) ( ( c b) (- (4)) (* 26) a)
--> [B-JF-TRUE]
nus, ( * a (+ b ())
-) [B-STAR]
N 5, 4
--> [B-]F-FALSE]
ovs (+ b #1)
-> [B-ADD]
nus, 6
-> [B -IF-TRUE]
rvs, (* 2 b)
--) [B-STAR]
anvs, 12
--> [B-]F-FALSE]
envs, a
--> [R-ADD]
env 5, 15
```

u 1-) #f (since 1 is 6 and 8 is 43, (> iJ) evaluates to #f, a 155, b is 25, and C is 30, so the Condition VI-> #1 (sine ; is B and sis 43, ((= 10) evaluates to #t, a is 5, 6 is 26, and Cisso, so the andition B-Plus: evaluates the sum of two expressions B-IF-TRUE: evaluates the conveybent of an if expression when the predicate evaluates to true. B-IF-FALSE : evaluates the alternate of an it expression when the delicate evaluates to Falsa R-STAR : evaluates the pholluct of two explessions. B-ADD: evaluates the run of two value.

```
envs, (+ (if ((ah) (+ a (+ b()) (+ b #1))
          (it(if (> * () ((cb) (-c4)) (+ 2b) 0))
--> fR-PLUSY
enus, (if ( a b) ( + a (+ b c)) (+ b #1))
     + (if (1) () a () ( ( ( h) (- (4)) ( x 2 b) +)
--> [B-]F -TRUE }
CAUS. (A a (+ 6 c))
   + (if (if () ac) ((cb) (- c4)) (+ 2 4) a)
-->fr-STAR?
CAUS, (* 5 (+ b ())
  + (if (if ( > ac) ((cb) (- c4)) ( 2 b) a)
--> fe - Anoi
CAV 5, (+ 3 b)
   + (if (if () a () (( ( b) (- (4)) (*2b) a)
--> frif-True}
envs, (* 31-)
     + (x 2 b)
--> /F-STAR}
```

env 5 , 30 + (# 2 h)

--> (B-ADD) enus, 30

+0 --> (BADD } envs , 42

472

--) [B-ADD] env 5, 45

B-PLUS: cultuales the sum of two explessions. B-IF-TRUE: rubluates the confequent of unif explession when the predicate eschusies to true. B-IF-FALSE: EVALUATES the alternate of an if expression when the exclinate evaluates to foliage RIMR: Evaluates the product of two expressions.

B-NOD: evolutes the sun of two values.