ממ"ן 14 שפות תכנות

שאלה 1:

השינוי שנדרש בסעיף א' ו הפרוצדורה שנדרשה בסעיף ג' נמצאים בקבצים בתקייה 1Q.

א. נמצא בתיקייה 1Q בתוספת הצגה של מגבלות הכריכה הדינמית

ב. השינוי לכריכה דינמית מאפשר לחישוב של ערך הפרוצדורה והארגומנטים שלה בעת הפעלתה וכפועל יוצא עם הסביבה הקיימת בזמן הפעלה בניגוד לכריכרה לקסיקלית בה החישוב מתבצע על הסביבה שהפונקציה הכירה בהגדרת.

ג. קיימים 2 טסטים בהם הגדרתי את פונקציה העצרת וחישבתי ערכים חוקיים עבורה:

a. $\boldsymbol{let\ Factorial\ =\ proc(n)\ if\ zero?\ (-(n,0))\ then\ 1\ else\ *(n,(Factorial\ -(n,1)))}$

שאלה 2:

א. טסט עבור הפרוצדורה נמצאה בתקיה 2Q בקובץ test.scm

a. $\boldsymbol{let\ istwice\ =\ proc\ (x)proc\ (y)zero?\left(-\left(x,-\left(y,-(0,y)\right)\right)\right)}$

ב. התוכנית מחזירה num-val 12, מאחר והפרוצדורה הנתונה מכפילה את הערך שניתן לה ב4, והיא קיבלה ערך 3. נסביר, Maker היא call-exp לכן היא מתנהגת כמו הפעלת פרוצדורה ונשים לב לדקות שהיא איננה פרוצדורה בעצמה, הפרוצדורה makemult קוראת לעצמה ובכך Maker מקבלת ערך של פרוצדורה ויכולה לפעולה. בכל הפעלה שלה אנו נותנים לה את עצמה ומורידים את הערך X שניתן לנו באחד ובעצם קוראים רקורסיבית לפונקציה כל פעם עד שנגיע לאפס.לאחר מכן תוצאת ביטוי החיסור (שבעצם הינו חיבור בגלל 4-) נוסיף 4 לפי עומק הרקורסיה ובעצם הכפלנו את הערך שניתן לנו ב4.

ג. הסבר מפורט וקריא יותר כולל וטסטים נמצאת בקבצים בתיקייה Q2, ובכל זאת זה הקוד:

```
let ishalf = proc (counter) proc (x)
            if zero? (x) then 0
            else if zero? (-(x,-(0,1))) then -(0,1)
            else -(((counter counter) -(x,2)),-1)
            in let half = proc (x)((ishalf ishalf)x) ;;;rutrs half of x

in let ispower = proc (multiplior)
        proc (n)
        proc (y)
        proc(halfn)
            if zero? (-(n,y)) ;;;;;;if true it means n is 1 or power of 2
            then zero? (0)
            else if zero? (halfn) ;;;;;
if we halfn is 0 it means we y is now greater then n so we can be sure n is not 2^n
            then zero? (1)
            else (((( (multiplior multiplior)n)
        -(y,-(0,y)))(half halfn)) ;; multipal y by 2 and recheck when
multiplior = ispower and devide halfn as a break - cond


            in let start = proc (n)
                if zero? (-(n,1))
                then zero? (0)
            else ((((ispower ispower)n) 2)
-(n,-(0,n))) ;; init the multiplayor to 1 to make sure we compair only powers of 2
                                        ;; and break
- cond to 2n so we can be sure wehn we passed n and found nouthing
        in (start n)) ;; assign n with a const num val and get the aswer.
```

שאלה 3:

נפרט על השינויים שביצענו בשפה על מנת לממש את הפונקציות בעלות הערך הדיפולטיבי.

- נשנה את הקובץ lang כך :

```
(expression
 ("proc" "(" (arbno identifier "=" expression) ")" expression)
 proc-exp)

(expression
 ("(" expression (arbno identifier "=" expression) ")")
 call-exp)
```

- מבחינת סוגי ביטויי בחרתי לשנות את proc כך שבמקום משתנה אחד שהוא סימבול היא תחזיר שתי רשימות רשימה אחת תהייה של שמות משתנים ואחת של ביטויים. כך :

```
;;;;;;;;;;;;;;;;; procedures ;;;;;;;;;;;;;;;;;

  ;; proc? : SchemeVal -> Bool
  ;; procedure : Var * Exp * Env -> Proc
  (define-datatype proc proc?
    (procedure
      (vars  (list-of symbol?))
      (defs (list-of expression?))
      (body expression?)
      (env environment?)))
```

- בעת הפעלת הפוקנציה :

1. נחלץ את הרשימות.
2. נבדוק האם הרשימה החדשה לא ארוכה יותר ושכל המשתנים שלה הם משתנים של הפונקציה המקורית
3. נשלח את הרשימות המעודכנות לפונקציה שתוסיף את הערכים של הרשימות לסביבה. קודם את הישנים ואחר כך את החדשים כך הערך האחרון שהסביבה תכיר יהיה הערך החדש ולא הדיפולטי
4. נחשב את ערך הפונקציה עם הסביבה החדשה.
5. בגלל שאנחנו לא משנים את הערך הדיפולטי ברשימה מובטח לנו כי בהפעלה אחרת יישאר הערך הדיפולטיבי במידה ולא שינו אותו גם בהפעלה האחרת.