

עקרונות שפות תכנות - תרגיל 2

רקורסיות, רשימות, Sequence operations

תאריך הגשה: 12.5.2025 אופן ההגשה: בזוגות, בלימוד בלבד

יש להגיש קובץ **scheme** יחיד ובו כל ההגדרות. את החוזים ממשו כהערות מעל הפרוצדורות המתאימות. שימו לב כי התכניות שלכם נבדקות ע"י סקריפט ולכן חייבות להיות ללא שגיאות תחביר, ולהחזיר **בדיוק** את הפלטים הנדרשים

בכל הסעיפים בכל השאלות רשמו את הטיפוס ואת תנאי הקדם כהערה מעל להגדרת הפונקציה

אין להשתמש ב `defines` פנימיים או במנגנונים שלא למדנו (למשל `memq`, `set!`, ועוד) – שימושים אלה יובילו לציון 0 בשאלה, **שימו לב!**

1. (40 נק')

- א. (4 נק') ממשו פונקציה רקורסיבית בשם `make-dup-list` שקבל כקלט רשימה ותחזיר רשימה שבה כל איבר מהרשימה המקורית מופיע פעמיים. למשל עבור הרשימה (1 2 5 6) תוחזר הרשימה (1 1 2 2 5 5 6)
- ב. (10 נק') ממשו פונקציה בשם `make-list-pairs` המקבלת רשימת מספרים שבה לפחות שני איברים (וודאו זאת ע"י `length`) ומחזירה רשימה חדשה שבה כל איבר הוא זוג האיבר הנוכחי ברשימה המקורית עם האיבר הבא.
למשל, (`make-list-pairs 1 2 5 6`) תחזיר ((1 . 2) (2 . 5) (5 . 6))
- ג. (8 נק') ממשו פונקציה רקורסיבית בשם `reverse-li` שתחזיר רשימה עם האיברים מהרשימה `li` בסדר הפוך.
- ד. (8 נק') ממשו פונקציה רקורסיבית בשם `is-special-form?` שתקבל כקלט רשימה סימבולים לא ריקה המיצגת אופרטורים עבור צורות מיוחדות, למשל הרשימה (`(if lambda define)`), ותחזיר `true` אם הביטוי הוא צורה מיוחדת. למשל הביטוי (`(lambda (x) x)`) הוא צורה מיוחדת, אולם הביטויים (`(2 (lambda (x) x))`) ו-`if` אינם צורות מיוחדות. העזרו בפונקציה `member` הבודקת שיכות לרשימה.
- ה. (10 נק') איבר **סימטרי** של איבר `e1`, הנמצא בחצי **השמאלי** של רשימה `li`, הוא איבר `e2` הנמצא בחצי **הימני** של הרשימה ומרחקו מסוף הרשימה זהה למרחקו של `e1` מתחילת הרשימה. למשל – אם הרשימה `li` היא (1 5 7 8) אז האיברים הסימטרי של 1 הוא 8, האיבר הסימטרי של 5 הוא 7. לאיבר 7 **לא** מוגדר איבר סימטרי, כי הוא בחצי **הימני** של הרשימה. אם ברשימה מספר אי זוגי של איברים, האיבר האמצעי הוא האיבר הסימטרי של עצמו.
- ממשו את `get-symm` המקבלת כקלט רשימה שבה לפחות 2 איברים (וודאו זאת), ומחזירה רשימת זוגות מקוננת שבה כל איבר מחצי הרשימה השמאלי מופיע עם האיבר הסימטרי שלו. הגדירו פונקציה עזר רקורסיבית עם פרמטרים כרצונכם (ממשו גם לה טיפוס).
- לדוגמא: הקריאה (`(get-symm (list 1 5 7 8))`) תחזיר את הרשימה המקוננת ((1 . 8) (5 . 7)) והקריאה (`(get-symm (list 1 2 3))`) תחזיר את הרשימה המקוננת ((1.3) (2.2))
- רמז:** אפשרות אחת לפתרון היא להשתמש ב `list-ref` שהגדרנו בכיתה (הגדירו אותו מחדש) יחד עם אופרטור `length`.

2. (20 נק')

א. (9 נק') ממשו את הפונקציה **apply-f-list** המקבלת כארגומנט רשימת פרוצדורות, שכל אחת מהן מקבלת מספר יחיד ומחזירה מספר, ובנוסף מספר יחיד x , ומחזירה רשימה שבה האיבר במקום i התקבל מהפעלת הפונקציה במקום i על x .
לדוגמא – נניח כי מוגדרת רשימת הפרוצדורות הבאה ואחריה ההפעלה:

```
(define f-li
  (list (lambda (x) (+ x 1))
        (lambda (x) (* x 3))
        (lambda (x) x)
        (lambda (x) (- 5 x))
  ))
(apply-f-list f-li 1)
```

אז נקבל כפלט את הרשימה: (2 3 1 4)

ב. (11 נק') ממשו את הפונקציה **apply-fs-single** המקבלת רשימת פרוצדורות f -li שכולן מקבלות ומחזירות אותו טיפוס, ואיבר e המתאים לטיפוס, ומפעילה את הפרוצדורות ברשימה בזו אחר זו על האיבר e כאשר f מופעלת ישירות על האיבר, ואז כל פונקציה תופעל על פלט הפונקציה הקודמת. כלומר אם הרשימה היא $(f1\ f2\ f3)$ אז הפלט שיתקבל הוא $((f3\ (f2\ (f1\ e)))$
חובה לכתוב רקורסיית זנב!
לדוגמא – עבור הקוד הבא יתקבל הפלט 21

```
(define fs (list (lambda (x) (* x 2))
                 (lambda (x) (* x 10))
                 (lambda (x) (+ x 1))))
(apply-fs fs 1)
```

3. (40 נק') שאלה זו עוסקת בפעולות על רשימות ומספרים ו- sequence operations.

א. (5 נק') ממשו פונ' בשם $dist_pwr$ שתקבל כקלט שני מספרים $num1, num2$ וחזקה n ותחשב את:

$$\sqrt[n]{(num1 - num2)^n}$$

n צריך להיות מספר טבעי. אם $n=1$ יש לקחת ערך מוחלט. השתמשו בפונקציה `expt` המבצעת את פעולת החזקה ב-Scheme

ב. (5 נק') ממשו **ללא** שימוש ב-`map` פונ' בשם `compute_dists` שתקבל כקלט רשימת מספרים li , מספר num וחזקה n , ותחזיר רשימה חדשה שבה כל איבר i יתקבל ע"י הפרש בחזקה n בין הערך num לבין האיבר המתאים ברשימה, לפי הנוסחא הבאה: $\sqrt[n]{(li_i - num)^n}$
כאשר li_i הוא האיבר באותו אינדקס מהרשימה המקורית. למשל – הקריאה

```
(compute_dists (list 1 2 3 4) 5 2)
```

תחשב את ריבועי הפרשי המרחקים של כל איבר מהמספר 5 ונקבל את הרשימה: (16 9 4 1)

ג. (7 נק') ממשו פונקציה דומה בשם `compute_dists_map` המבצעת חישוב זהה ומשתמשת ב-`map`

ד. (7 נק') בעזרת קריאה **אחת** ל-`filter` ממשו פונקציה בשם `dists_in_range` שתקבל פרמטרים כמו הפונקציה בסעיף הקודם, ובנוסף $num1, num2$ – שני מספרים נוספים, ותשאיר ברשימה רק את המרחקים שאינם בקטע $[min, max]$. השתמשו באחד המימושים מהסעיפים הקודמים

ה. (8 נק') ללא קשר לסעיף הקודם, ממשו פונקציה בשם `compute_dists_pwr_range` שתתנהג כמו הפונקציה מסעיף ג' (או ד') ובנוסף min_pwr, max_pwr ותחזיר מטריצה (רשימה של רשימות)

שבה השורה הראשונה היא חישוב מרחקים בחזקת \min_pwr , השניה בחזקת \min_pwr+1 והאחרונה בחזקת \max_pwr .

ו. **(8 נק')** בעזרת הפונקציות מסעיפים $a+b$ ממשו פונקציה בשם `dist_mat` שתקבל כקלט רשימה מספרים וערך `num` ותחזיר את מטריצה המרחקים בין כל האיברים כאשר כל מרחק הוא הפרש בחזקת `num`. למשל `(dist_mat (list 2 4 5) 2)` תחזיר מרחקים בחזקת 2 בין כל האיברים ונקבל:

`((0 4 9) (4 0 1) (9 1 0))`

בהצלחה