

עקרונות שפות תכנות - תרגיל 1

תאריך הגשה: 24.4.2025 אופן ההגשה: בזוגות בלבד

יש להגיש קובץ scm או rkt יחיד עם כל הקוד הנדרש. התכניות שלכם ייבדקו ע"י סקריפט ולכן חייבות להיות ללא שגיאות תחביר, ולהחזיר בדיוק את הפלטים הנדרשים.

בכל סעיף בשאלות שלהלן נדרש לכתוב גם טיפוסים. הוסיפו אותן מעל לקוד הפרוצדורה המתאימה כהערות בקוד (בעזרת סימן ; ב-Scheme). למשל:

Type: Number->Number

הניחו כי כל הקלטים תקינים. חלק מהניקוד בכל שאלה, בכל סעיף ינתן עבור הטיפוס

1. (20 נק') בתרגיל זה תממשו פעולות המקבלות כקלט קבוצת מספרים, ומבצעות בדיקות הקשורות לממוצע שלהם.

א. (4 נק') ממשו פרוצדורת משתמש בשם cubicroot המקבלת מספר חיובי num ומחזירה את השורש השלישי שלו (כלומר num בחזקת שליש). היעזרו באופרטור expt שראינו בכיתה.

ב. (6 נק') בעזרת הפרוצדורה מהסעיף הקודם ממשו את הפונקציה geomavg3 המקבלת 3

מספרים חיוביים ומחזירה את הממוצע ההנדסי שלהם לפי הנוסחה: $\sqrt[3]{a * b * c}$.

ג. (6 נק') בעזרת 2 הפרוצדורות מהסעיפים הקודמים בלבד (וללא שימוש בפעולות חשבון נוספות כלשהן), ממשו פרוצדורה בשם geomavg9 המקבלת כקלט 9 מספרים ומחזירה את הממוצע ההנדסי שלהם.

ד. (4 נק') כתבו פרוצדורה בשם numLarger המקבלת כקלט שלושה מספרים ומחזירה את מספר המספרים הגדולים ממש מהממוצע ההנדסי של המספרים. למשל

(numLarger 1 2 4)

→ 1

(numLarger 1 4 4)

→ 2

הדרכה: הגדירו משתנה לוקאליים מתאימים ע"י let.

2. (40 נק')

א. (2 נק') ממשו את הפרוצדורה compose2 המקבלת כארגומנט שתי פרוצדורות f1, f2 ומחזירה פרוצדורה של משתנה אחד (x) המבצעת הפעלה של f1 ולאחר מכן הפעלה של f2. – בסימון

מתמטי: $f2(f1(x))$

רוצים לבדוק את הזהות: $\log_m(x^n) = n(\log_m x)$, והסעיפים הבאים נוגעים לכתיבת

פרוצדורות שיממשו ויבדקו אותה.

- ב. (4 נק') ממשו, באמצעות `compose2`, את הפרוצדורה `logmul` המקבלת המקבלת כקלט שני שלמים m, n ומחזירה את פונקציה של משתנה יחיד x המחשבת את $f(x) = n \log_m x$
- ג. (8 נק') ממשו, ללא שימוש ב `compose2`, את הפרוצדורה `make_log_expt` המקבלת כקלט שני שלמים m, n ומחזירה את פונקציה של משתנה יחיד x המחשבת את $f(x) = \log_m x^n$.
למשל הקריאה (`make-log-expt 2 3`) תחזיר את הפונקציה המתמטית: $f(x) = \log_2 x^3$
אשר אם תופעל על הערך 1 תחזיר 0, ולכן (`make-log-expt 2 3`) 1 תחזיר 0
- ד. (8 נק') ממשו את `make-log-expt` במימוש נוסף (קראו לו `make-log-expt-v2`), ע"י שימוש ב `compose2` שתקבל כקלט שתי פרוצדורות **ללא שם**. בדומה לסעיף הקודם, הקריאה (`make-log-expt-v2 2 3`) 1 תחזיר 0.

ה. (8 נק') בעזרת `make-log-expt` בלבד **וללא** שימוש בפעולות חשבון נוספות, ממשו פרוצדורה בשם `log2` המקבלת פרמטר יחיד x ומחזירה את $\log_2 x$, כך שלמשל `log2 8` תחזיר 3.

- ו. (8 נק') עתה רוצים לבחון את הזהות $\log_m x^n = n \log_m x$, כאשר את הפונקציות המתאימות ממשותם בסעיפים ב,ג. בחרו 3 זוגות ערכים שונים עבור m, n ולכל אחד מהם צרו שני פונקציות למשתנה יחיד x , הפעילו כל אחת על ערך x כלשהו, והדפיסו `true` אם התוצאות שונות ו `false` אחרת. למשל עבור הזוג $n=3, m=2$ יש ליצור את $\log_2 x^3$ ואת $(\log_2 x)^3$, להפעיל אותן על שני ערכי x שונים ולכל ערך x להדפיס קודם שורה חדשה (`newline`) בכל מקרה ואז הערך `true` אם מתקבל ערך זהה ו-`false` אחרת. אם הכל תקין צריכות להתקבל 3 הדפסות של `true`. **בנוסף עד 2 נק'**: ממשו פונקציה בשם `verify-eq` עבור בדיקה אחת, ופונקציה מעטפת בשם `check_n` שתקבל כפרמטר את מספר הבדיקות שיש לבצע ותקרא לפונקציה `verify-eq` עם ערכים אחרים בכל פעם

3. (40 נק') תרגיל זה עוסק בהחזרת פרוצדורות
- א. (4 נק') נניח כי נתונות הפונקציה (`(define (make_times_n n) (lambda (x) (* x n)))`) והפונקציה (`(define (square x) (* x x))`). ממשו את הפרוצדורה `make-parabola` שראינו בתרגול, במקום פעולות חזקה או כפל השתמשו בשני הפונקציות הנתונות.
- ב. (6 נק') כתבו את טיפוס הפרוצדורה `make-parabola` ואת ה `preconditions` שלה. הניחו כי חייב להיות ביטוי ריבועי (כלומר $0x^2$ אינו חוקי)
- (6 נק') פונקציה משולשת מעל a, b, c מוגדרת כפונקציה: $(x - a)(x - b)(x - c)$
כתבו פרוצדורה בשם `make-triple` המקבלת כקלט a, b, c ומחזירה את הפונקציה המשולשת מעל a, b, c .
- ג. (6 נק') ממשו פרוצדורה בשם `make_triple_or_parabola_N` המקבלת כקלט a, b, c ומספר N . הפרוצדורה תשווה בין שני הערכים:

$$aN^2 + bN + c$$

$$(N - a)(N - b)(N - c)$$

אם הראשון גדול יותר, נחזיר פרבולה עם a, b, c , ואם לא – נחזיר את הפונקציה המשולשת מעל a, b, c . חובה לעשות בסעיף זה שימוש מקסימאלי בסעיפים הקודמים ומינימום חזרה על קוד. מימוש שלא יענה על דרישה זו, יקבל עד 2 נקודות בלבד.

ד. (18 נק') כתבו פרוצדורה בשם `make_triple_parabola` המקבלת כקלט a, b, c ומחזירה פונקציה של משתנה x , שמחזירה ערכים **שונים מ-0** באופן הבא:

a. ראשית נפעיל גם את הפרבולה וגם את הפונקציה המשולשת על x ונחזיר את הפעלת הפונקציה שלא החזירה 0 (עם עדיפות לפרבולה)

b. אם שתי הפונקציות מחזירות 0 – נבדוק את x : אם אינו 0 נחזיר אותו, ואם ערכו 0 נחזיר 1

חובה להשתמש ב `let` ולכתוב פונקציה קצרה ככל האפשר, עם שימוש מקסימאלי בסעיפים קודמים. מימוש שלא יעמוד בתנאים יזכה בניקוד חלקי בלבד.

בהצלחה