

Project - part 2 (Semantics)

הוסיפו ל-parser שכתבתם את הבדיקות הסמנטיות הבאות (יש לממש ולהעזר ב-symbol table):

- (1) קיימת פרוצדורה Main בקוד והיא יחידה.
- (2) Main לא מקבל ארגומנטים.
- (3) לא קיימות שתי פונקציות או פרוצדורות עם אותו שם באותו scope.
- (4) לא קיימים שני משתנים עם אותו שם באותו scope.
- (5) פונקציות ופרוצדורות הוגדרו לפני שמפעילים אותן.
- (6) משתנים הוגדרו לפני שמשתמשים בהם.
- (7) מספר הארגומנטים האקטואליים שווה למספר הארגומנטים הפורמליים של הפונקציה/פרוצדורה - כמות הארגומנטים בקריאה לפונקציה/פרוצדורה צריכה להיות שווה לכמות הארגומנטים בהגדרת הפונקציה/פרוצדורה.
- (8) טיפוסים של הארגומנטים בקריאה לפונקציה/פרוצדורה תואמים לטיפוסים בהגדרת הפונקציה/פרוצדורה.
- (9) טיפוס הערך המוחזר מהפונקציה תואם לטיפוס ההחזרה המוכרז בכותרת של הפונקציה. וטיפוס ההחזרה של הפונקציה לא יכול להיות מחרוזת.
- (10) טיפוס הערך המוחזר מהפונקציה תואם לטיפוס המשתנה שלתוכו נכנס הערך שמוחזר מהפונקציה.
- (11) טיפוס התנאי ב-if הוא מטיפוס bool.
- (12) טיפוס התנאי ב-while הוא מטיפוס bool.
- (13) טיפוס הביטוי המופיע כאינדקס ב-[] של מחרוזת הוא מטיפוס int.
- (14) לא משתמשים באופרטור [] בשום טיפוס חוץ ממחרוזת.
- (15) טיפוס של המשתנה מצד שמאל של האופרטור השמה (=) תואם לטיפוס הביטוי מצד ימין. לשים לב שלתאים של המחרוזת מותר להכניס רק תווים ו-null יכול להיות רק מטיפוס מצביע.

16) טיפוסים בביטויים (expressions) תואמים. הכללים הם:

- עבור אופרטורים (+, -, *, /) האופרנדים יכולים להיות int או real והתוצאה היא int אם שני האופרנדים מטופס int, אחרת הטיפוס של הביטוי הוא real.
- עבור אופרטורים (&&, ||) שני האופרנדים חייבים להיות bool והתוצאה היא bool.
- עבור האופרטורים (<, <=, >, >=) האופרנדים חייבים להיות int או real והתוצאה היא bool.
- עבור האופרטורים (!=, ==) שני האופרנדים יכולים להיות שני int, שני bool, שני real, שני char, או שני מצביעים לאותו טיפוס. התוצאה היא bool.
- אופרטור ערך מוחלט (| |) יכול להיות מופעל על מחרוזות והתוצאה היא int.
- אופרטור (!) יכול להיות מופעל על bool והתוצאה היא bool.

17) אופרטור & מופעל רק על משתנים מטיפוס int, real, char או string[i].

18) אופרטור אונרי ^ מופעל רק על מצביעים.

יש להדפיס הודעה מתאימה לכל סוג של טעות סמנטית.

הקוד עבור ה-compiler שאתם כותבים אמור לזהות את כל הסוגי השגיאות: tokens, תחביר וסמנטיקה ולהדפיס הודעת שגיאה מתאימה.

הוראות הגשה:

יש להגיש קובץ ZIP ב-moodle הכולל:

- קוד של ה-parser, scanner ומנתח סמנטי הכתובים ב-C, lex ו-yacc.
- קובץ PDF עם מספר קבוצה, שמות ות.ז. של כל המשתתפים בקבוצה. בנוסף יש להוסיף דוגמאות ריצה של המנתח הסמנטי על הקלטים השונים הכוללים את כל האלמנטים של השפה (פונקציות/פרוצדורות מקוננות הכוללות הגדרות משתנים מטיפוסים שונים, לולאות מקוננות ותנאים). **יש לצרף דוגמאות של ריצה עם כל סוגי השגיאות הסמנטיות (דוגמא אחת לפחות לכל סעיף מלמעלה).**

בהצלחה!