Project - part 2 (Semantics)

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

1) קיימת פונקציה _main_ בקוד והיא יחידה.

2) _main_ לא מקבלת ארגומנטים ולא מחזירה ערך.

3) לא קיימות שתי פונקציות עם אותו שם באותו scope.

4) לא קיימים שני משתנים עם אותו שם באותו scope. שימו לב ש-var מגדיר משתנים עבור -begin end שבא מיד אחריו, כלומר מוגדרים ב-scope של begin-end.

5) פונקציות הוגדרו לפני שמפעילים אותן.

6) משתנים הוגדרו לפני שמשתמשים בהם.

7) מספר הפרמטרים האקטואליים שווה למספר הפרמטרים הפורמליים של הפונקציה - כמות הפרמטרים בקריאה לפונקציה צריכה להיות שווה לכמות הפרמטרים בהגדרת הפונקציה.

8) טיפוסים של הפרמטרים בקריאה לפונקציה תואמים לטיפוסים בהגדרת הפונקציה.

9) טיפוס הערך המוחזר מהפונקציה תואם לטיפוס ההחזרה המוכרז בכותרת של הפונקציה. וטיפוס ההחזרה של הפונקציה לא יכול להיות מחרוזת.

10) טיפוס התנאי ב-if הוא מטיפוס bool.

11) טיפוס התנאי בלולאות הוא מטיפוס bool.

12) טיפוס הביטוי המופיע כאינדקס ב-[] של מחרוזת הוא מטיפוס int.

13) לא משתמשים באופרטור [] בשום טיפוס חוץ ממחרוזת.

14) 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.

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

- עבור אופרטורים (**+**,**-**,*****,**/**) האופרנדים יכולים להיות int או real והתוצאה היא int אם שני האופרנדים מטופוס 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.

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

17) אופרטור **אונרי** * מופעל רק על מצביעים.

18) פרמטרים בפונקציות מסודרות לפי סדר. כלומר, קודם par1, אחריו par2 וכו'.

יש להדפיס הודעה מתאימה לכל סוג של טעות סמנטית.

הקוד עבור ה-compiler אמור לזהות את כל הסוגי השגיאות: tokens, תחביר וסמנטיקה ולהדפיס הודעת שגיאה מתאימה. במידה והקוד עבר קומפילציה יש להדפיס עץ כמו בחלק 1.

<u>הוראות הגשה:</u>

יש להגיש קובץ ZIP ב-moodle הכולל:

- קוד של ה-parser, scanner ומנתח סמנטי הכתובים ב- lex, C ו-yacc.
- קובץ PDF עם שמות ות.ז. של כל המשתתפים בקבוצה.

בהצלחה!