

## Project - part 1 (Scanner and Parser)

יש לכתוב scanner ב-lex ו- parser ב-yacc לשפה שתיאור שלה פורסם במודל. ה- parser אמור לקבל כל קוד אפשרי בשפה הנ"ל ולבנות Abstract Syntax Tree (AST). אם יש טעות כלשהי (טעות בטוקנים או טעות תחבירית) על ה- parser להדפיס הודעת שגיאה בהתאם לסוג של הטעות ויש לפרט מה בדיוק הטעות. יש לכתוב פונקציה אשר סורקת עץ ב- preorder ומדפיסה את ה-AST במידה והקוד תקין. **אל הפונקציה להשתמש בהדפסה בסוגריים ובהזחות (בדומה לדוגמא) על מנת להדגיש את הקינון של הקוד.**

לדוגמא, עבור הקוד:

```
function void foo(int x, y, z; real f){
    if (x>y) {
        x = x + f;
    }
    else {
        y = x + y + z;
        x = f * 2;
        z = f;
    }
}
```

```
function char goo(){
    return 'a';
}
```

הדפסה של העץ:

```
(CODE
  (FUNCTION
    foo
    (ARGS
      (INT x y z)
      (REAL f)
    )
    (TYPE VOID)
    (BODY
      (IF-ELSE
        (> x y)
        (BLOCK
          (= x
            (+ x f)
          )
        )
        (BLOCK
          (= y
            (+
              (+ x y)
              z
            )
          )
        )
      )
    )
  )
  (FUNCTION
    goo
    (ARGS NONE)
    (TYPE CHAR)
    (BODY
      (RET 'a')
    )
  )
)
```

## הערות הגשה:

יש להגיש קובץ ZIP ב-moodle הכולל:

- קוד של ה-scanner ו-parser הכתובים ב-C, lex ו-yacc.
- קובץ PDF עם שמות ות.ז. של כל המשתתפים בקבוצה. בנוסף יש להוסיף דוגמאות ריצה של ה-parser על הקלטים השונים הכוללים כל האלמנטים של השפה (פונקציות הכוללות הגדרות משתנים מטיפוסים שונים, לולאות מקוננות ותנאים). יש גם לצרף דוגמאות עם השגיות.

**יש להגיש את הקובץ רק פעם אחת, ע"י אחד מחברי הקבוצה בלבד.**

**בהצלחה!**