

פרויקט קומפילציה – חלק ראשון

שמות המגשימים:

1. שם: מור דהן
תז: 204500409

2. שם: דוד אדרי
תז: 312215809

3. שם: אביב ממן
תז: 205622244

דוגמאות הרצה:

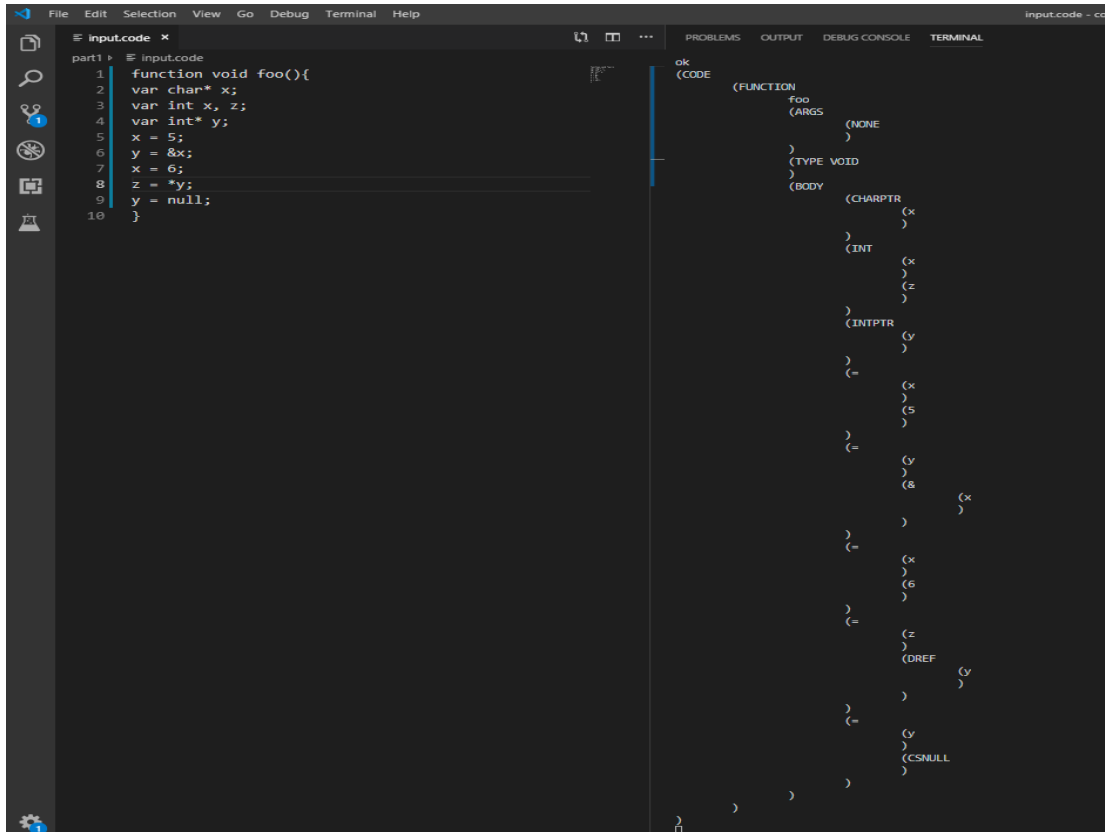
דוגמת ההרצה מהעבודה:

The screenshot displays a C compiler IDE with two main panels. The left panel shows the source code for a file named 'input.code':

```
1 function void foo(int x, y, z; real f){
2   if (x>y) {
3     x = x + f;
4   }
5   else {
6     y = x + y + z;
7     x = f * 2;
8     z = f;
9   }
10 }
11 function char goo(){
12   return 'a';
13 }
```

The right panel shows the abstract syntax tree (AST) for the same code. The root node is '(CODE)', which branches into '(FUNCTION foo (ARGS (INT (x), (INT (y), (INT (z), (REAL (f))))) (BODY (IF-ELSE (> (x), (y), (BLOCK (= (x), (+ (x), (f))))) (TYPE VOID) (BLOCK (IF-ELSE (> (x), (y), (BLOCK (= (x), (+ (x), (f))))) (TYPE CHAR) (BLOCK (RET ('a')))))

הגדרות משתנים מטיפוסים שונים והשמות

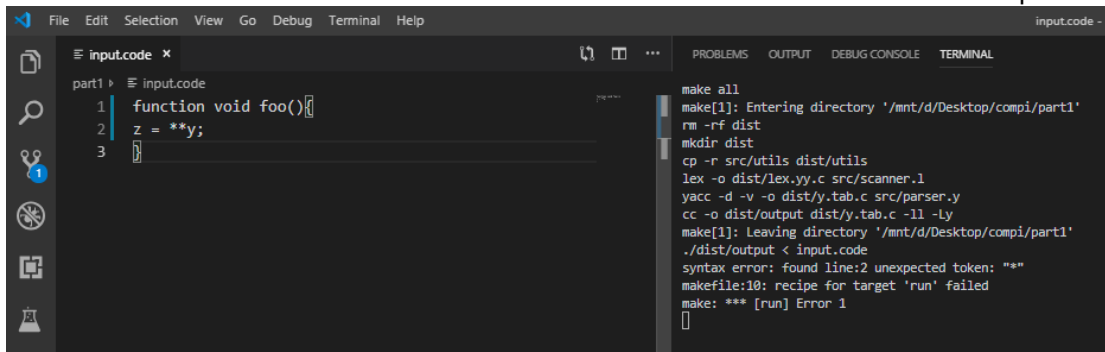


The screenshot shows the VS Code editor with a file named `input.code`. The code defines a function `foo()` with the following logic:

```
1 function void foo(){
2   var char* x;
3   var int x, z;
4   var int* y;
5   x = 5;
6   y = &x;
7   x = 6;
8   z = *y;
9   y = null;
10 }
```

To the right of the editor, the AST (Abstract Syntax Tree) is displayed. It shows the hierarchical structure of the code, starting from the `(FUNCTION` node, which contains the function name `foo`, its arguments `(ARGS)`, its return type `(TYPE VOID)`, and its body `(BODY)`. The body is further broken down into individual statements, such as `(CHARPTR` for the variable `x`, `(INT` for `x` and `z`, `(INTPTR` for `y`, and `(CNULL` for `null`.

שגיאות קוד והדפסת שגיאה:

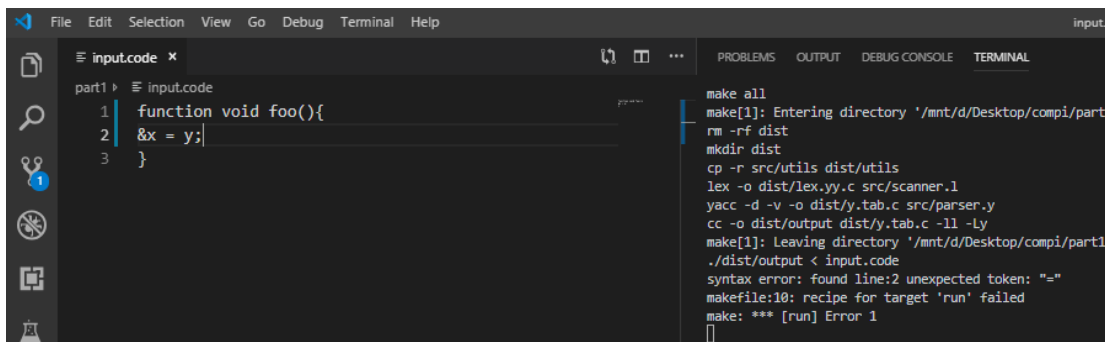


The screenshot shows the VS Code editor with a file named `input.code`. The code defines a function `foo()` with the following logic:

```
1 function void foo(){
2   z = **y;
3 }
```

To the right of the editor, the output of a `make` command is displayed. The output shows the following steps:

```
make all
make[1]: Entering directory '/mnt/d/Desktop/comp1/part1'
rm -rf dist
mkdir dist
cp -r src/utls dist/utls
lex -o dist/lex.yy.c src/scanner.l
yacc -d -v -o dist/y.tab.c src/parser.y
cc -o dist/output dist/y.tab.c -ll -ly
make[1]: Leaving directory '/mnt/d/Desktop/comp1/part1'
./dist/output < input.code
syntax error: found line:2 unexpected token: "**"
makefile:10: recipe for target 'run' failed
make: *** [run] Error 1
```



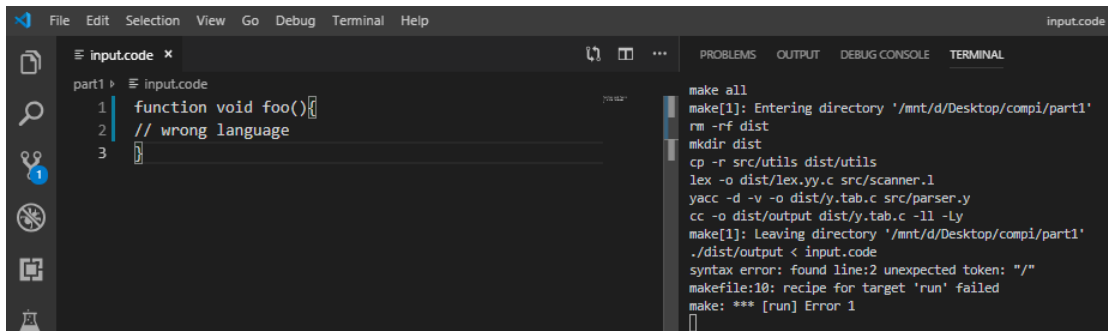
The screenshot shows the VS Code editor with a file named `input.code`. The code defines a function `foo()` with the following logic:

```
1 function void foo(){
2   &x = y;
3 }
```

To the right of the editor, the output of a `make` command is displayed. The output shows the following steps:

```
make all
make[1]: Entering directory '/mnt/d/Desktop/comp1/part1'
rm -rf dist
mkdir dist
cp -r src/utls dist/utls
lex -o dist/lex.yy.c src/scanner.l
yacc -d -v -o dist/y.tab.c src/parser.y
cc -o dist/output dist/y.tab.c -ll -ly
make[1]: Leaving directory '/mnt/d/Desktop/comp1/part1'
./dist/output < input.code
syntax error: found line:2 unexpected token: "="
makefile:10: recipe for target 'run' failed
make: *** [run] Error 1
```

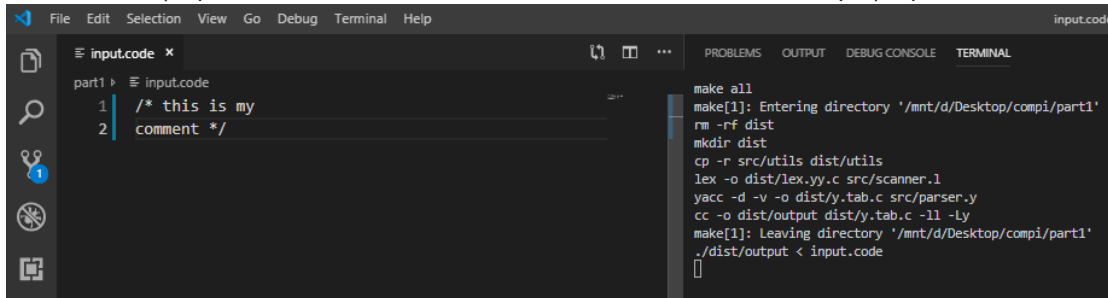
Comments בצורה לא נכונה לשפה



The screenshot shows the VS Code editor with a file named `input.code`. The code contains a C function definition with an invalid comment: `function void foo(){ // wrong language }`. The terminal on the right shows the output of a `make` command, which fails with a syntax error: `syntax error: found line:2 unexpected token: "/"`.

```
make all
make[1]: Entering directory '/mnt/d/Desktop/compi/part1'
rm -rf dist
mkdir dist
cp -r src/utls dist/utls
lex -o dist/lex.yy.c src/scanner.l
yacc -d -v -o dist/y.tab.c src/parser.y
cc -o dist/output dist/y.tab.c -ll -ly
make[1]: Leaving directory '/mnt/d/Desktop/compi/part1'
./dist/output < input.code
syntax error: found line:2 unexpected token: "/"
makefile:10: recipe for target 'run' failed
make: *** [run] Error 1
```

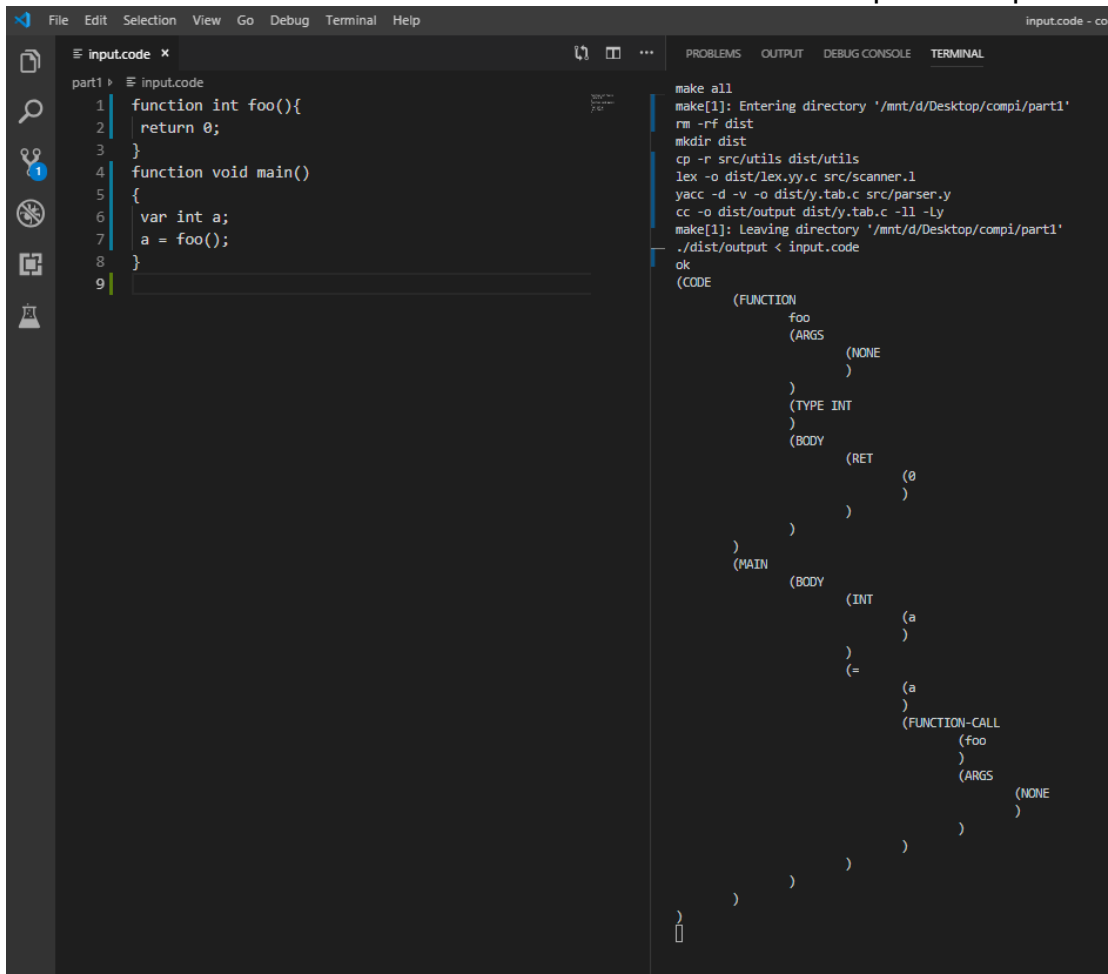
Comments בצורה תקינה(ניתן לראות שהפלט לא מדפיס כלום ומתעלם מהערות הרשומות בצורה תקינה)



The screenshot shows the VS Code editor with a file named `input.code`. The code contains a C function definition with a valid comment: `/* this is my comment */`. The terminal on the right shows the output of a `make` command, which completes successfully without any errors.

```
make all
make[1]: Entering directory '/mnt/d/Desktop/compi/part1'
rm -rf dist
mkdir dist
cp -r src/utls dist/utls
lex -o dist/lex.yy.c src/scanner.l
yacc -d -v -o dist/y.tab.c src/parser.y
cc -o dist/output dist/y.tab.c -ll -ly
make[1]: Leaving directory '/mnt/d/Desktop/compi/part1'
./dist/output < input.code
```

דוגמת קריאה לפונקציה



The screenshot shows the VS Code editor with a file named `input.code`. The code contains a C program with a function `foo` and a `main` function that calls `foo`. The terminal on the right shows the output of a `make` command, which completes successfully. Below the terminal output, the program's execution is visualized as a tree structure showing the function calls and return values.

```
make all
make[1]: Entering directory '/mnt/d/Desktop/compi/part1'
rm -rf dist
mkdir dist
cp -r src/utls dist/utls
lex -o dist/lex.yy.c src/scanner.l
yacc -d -v -o dist/y.tab.c src/parser.y
cc -o dist/output dist/y.tab.c -ll -ly
make[1]: Leaving directory '/mnt/d/Desktop/compi/part1'
./dist/output < input.code
ok
(CODE
  (FUNCTION
    foo
    (ARGS
      (NONE)
    )
    (TYPE INT)
    (BODY
      (RET
        (0)
      )
    )
  )
  (MAIN
    (BODY
      (INT
        (a)
      )
      (=
        (a)
      )
      (FUNCTION-CALL
        (foo)
        (ARGS
          (NONE)
        )
      )
    )
  )
)
```

פונקציות עם ארגומנטים מטיפוסים שונים

[illegible]

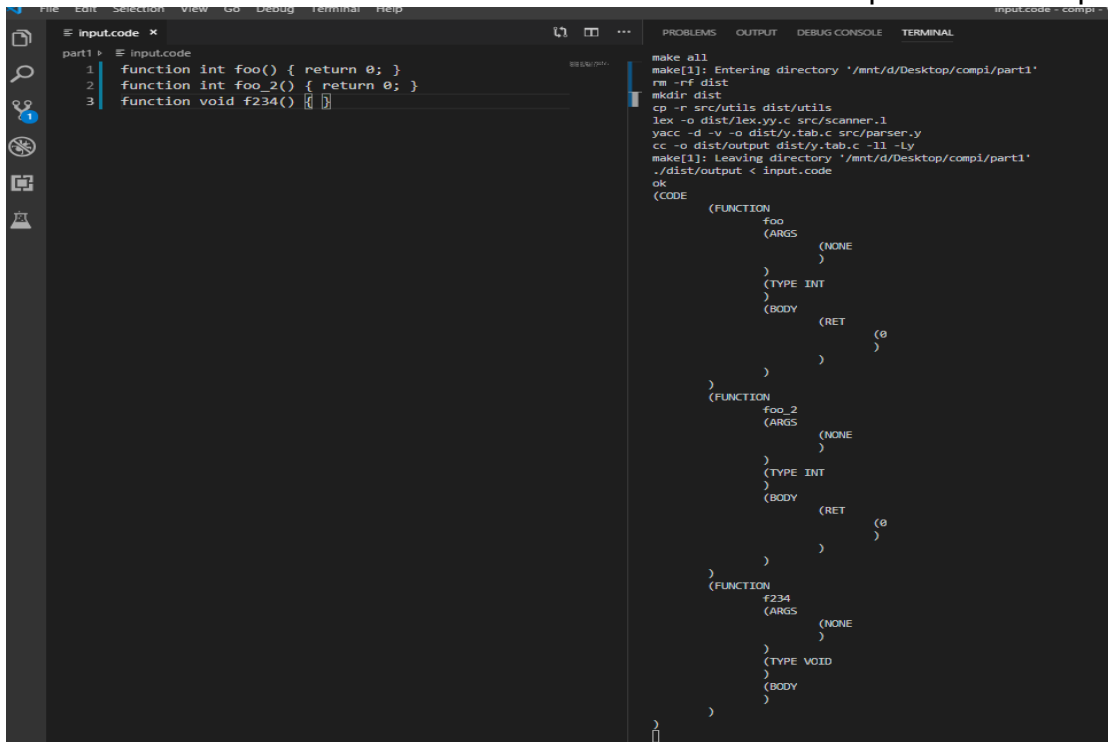
```

File Edit Selection View Go Debug Terminal Help
input.code - con

part1 ▸ input.code
1 function void goo(int i, j, k)
2 {
3     function bool fee(int l, m, n)
4     {
5         return true;
6     }
7     fee(2,3,4);
8 }
9

make[1]: Entering directory '/mnt/d/Desktop/comp1/part1'
rm -rf dist
mkdir dist
cp -r src/utlis dist/utlis
lex -o dist/lex.yy.c src/scanner.l
yacc -d -v -o dist/y.tab.c src/parser.y
cc -o dist/output dist/y.tab.c -ll -ly
make[1]: Leaving directory '/mnt/d/Desktop/comp1/part1'
./dist/output < input.code
ok
(CODE
(FUNCTION
  BOO
  (ARGS
    (INT
      )
    )
    (INT
      (i
      )
      )
    (j
      )
    (k
      )
    )
  )
  (TYPE VOID
  )
  )
  (BODY
    (FUNCTION
      fee
      (ARGS
        (INT
          )
          (l
          )
          (m
          )
          (n
          )
          )
        )
        (TYPE BOOL
        )
        )
        (BODY
          (RET
            (TRUE
            )
            )
          )
        )
      )
    )
    (FUNCTION-CALL
      (fee
      )
      (ARGS
        (2
        )
        (3
        )
        (4
        )
        )
      )
    )
  )
)
  
```

תקינות שמות הפונקציות

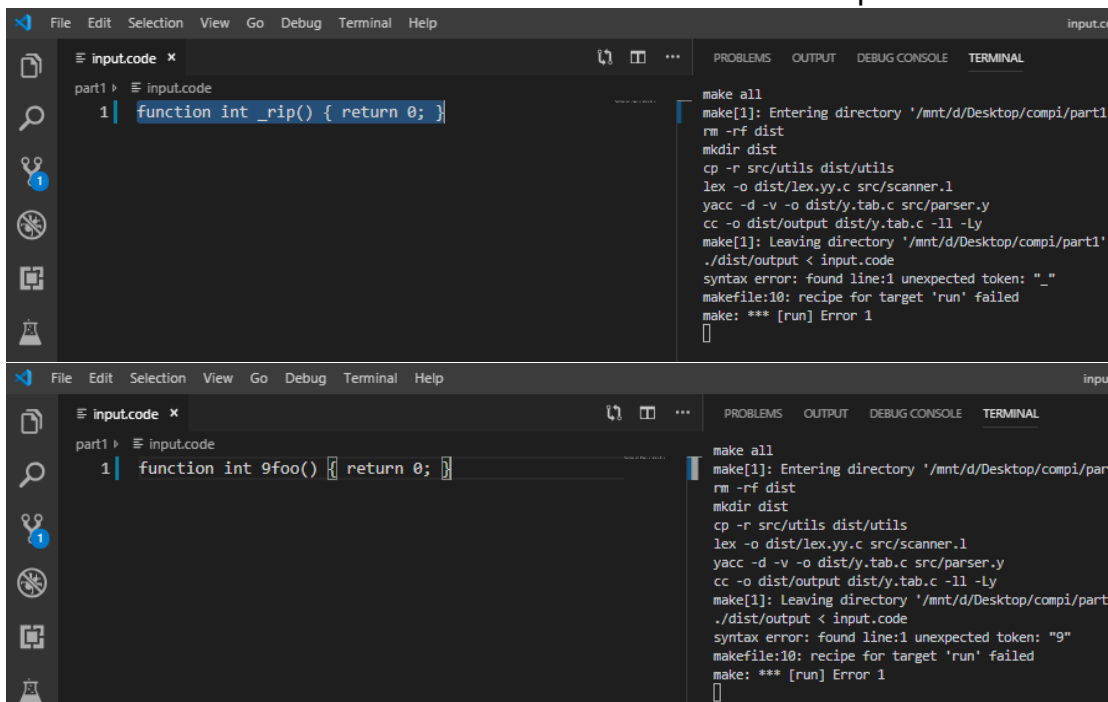


The screenshot shows the VS Code editor with a file named `input.code` containing three function definitions:

```
1 function int foo() { return 0; }
2 function int foo_2() { return 0; }
3 function void f234() {}
```

The terminal output shows the successful execution of the `make` command, including the generation of `dist/lex.yy.c` and `dist/y.tab.c`, and the final compilation of `dist/output` from `input.code`. The output also displays the internal representation of the functions as generated by the compiler.

שגיאה בשמות הפונקציות



The first screenshot shows the VS Code editor with a file named `input.code` containing a function definition with an invalid name:

```
1 function int _rip() { return 0; }
```

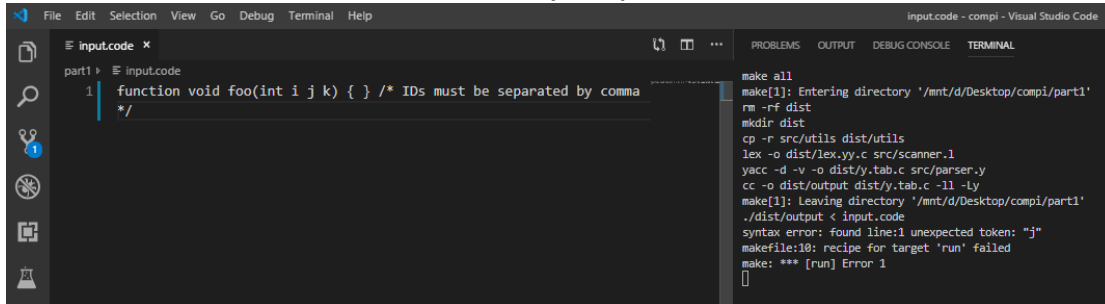
The terminal output shows the `make` command failing with the error: `syntax error: found line:1 unexpected token: "_"`.

The second screenshot shows the VS Code editor with a file named `input.code` containing a function definition with an invalid name:

```
1 function int 9foo() { return 0; }
```

The terminal output shows the `make` command failing with the error: `syntax error: found line:1 unexpected token: "9"`.

שגיאה בארגומנטים השפה מחייבת פסיקים בין שמות המשתנים

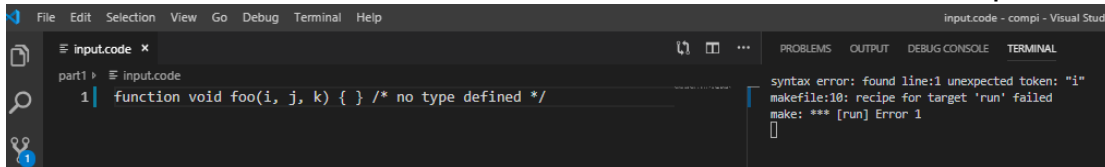


```
File Edit Selection View Go Debug Terminal Help
input.code - comp1 - Visual Studio Code

part1 > input.code
1 function void foo(int i j k) { } /* IDs must be separated by comma */
/*

make all
make[1]: Entering directory '/mnt/d/Desktop/comp1/part1'
rm -rf dist
mkdir dist
cp -r src/utlis dist/utlis
lex -o dist/lex.yy.c src/scanner.l
yacc -d -v -o dist/y.tab.c src/parser.y
cc -o dist/output dist/y.tab.c -ll -ly
make[1]: Leaving directory '/mnt/d/Desktop/comp1/part1'
./dist/output < input.code
syntax error: found line:1 unexpected token: "j"
makefile:10: recipe for target 'run' failed
make: *** [run] Error 1
```

שגיאה אין טיפוס משתנה לפני שמות הארגומנטים

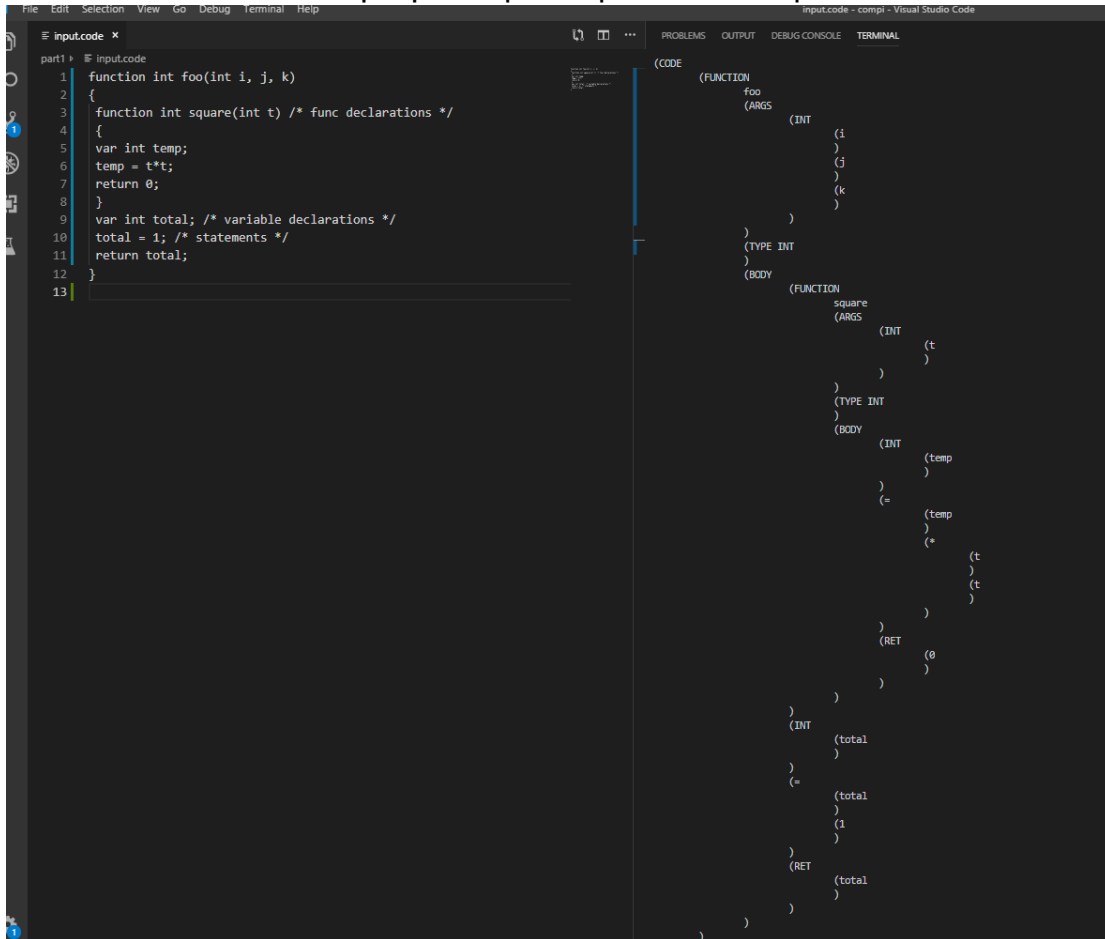


```
File Edit Selection View Go Debug Terminal Help
input.code - comp1 - Visual Studio Code

part1 > input.code
1 function void foo(i, j, k) { } /* no type defined */

syntax error: found line:1 unexpected token: "i"
makefile:10: recipe for target 'run' failed
make: *** [run] Error 1
```

השפה מחייב הכרזות פונקציות ומשתנים קודם ורק לאחר מכן הצהרות

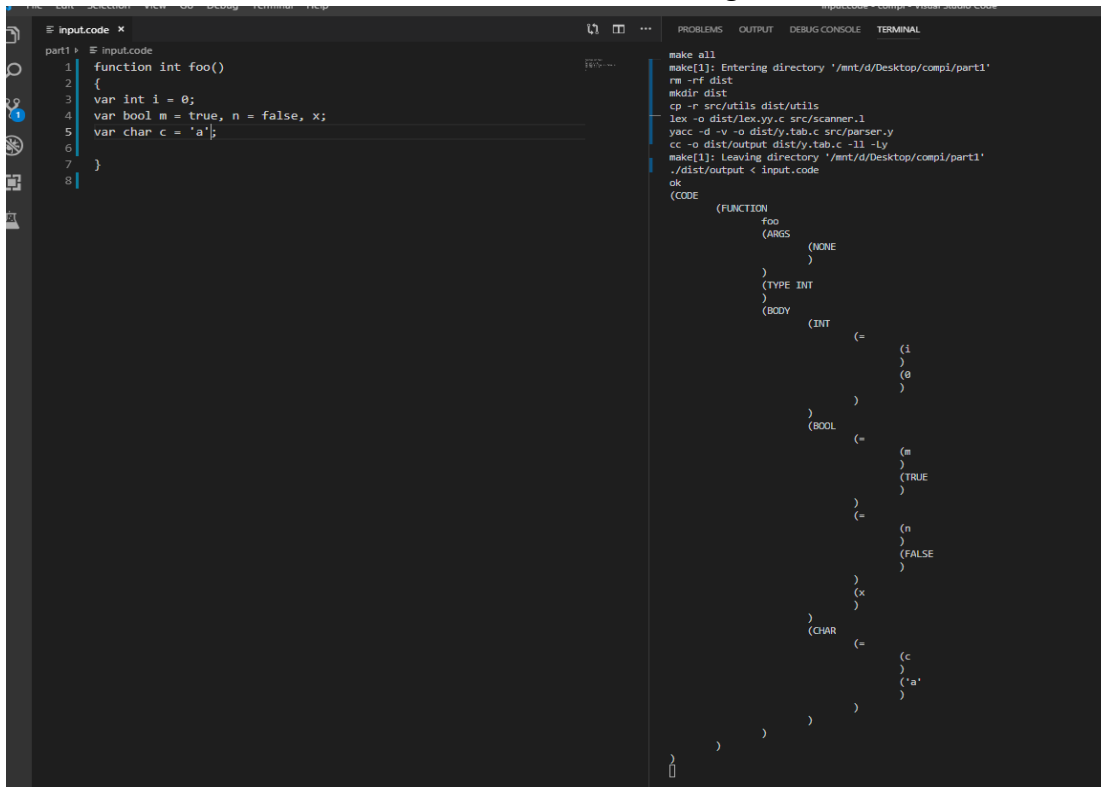


```
File Edit Selection View Go Debug Terminal Help
input.code - comp1 - Visual Studio Code

part1 > input.code
1 function int foo(int i, j, k)
2 {
3     function int square(int t) /* func declarations */
4     {
5         var int temp;
6         temp = t*t;
7         return 0;
8     }
9     var int total; /* variable declarations */
10    total = 1; /* statements */
11    return total;
12 }
13

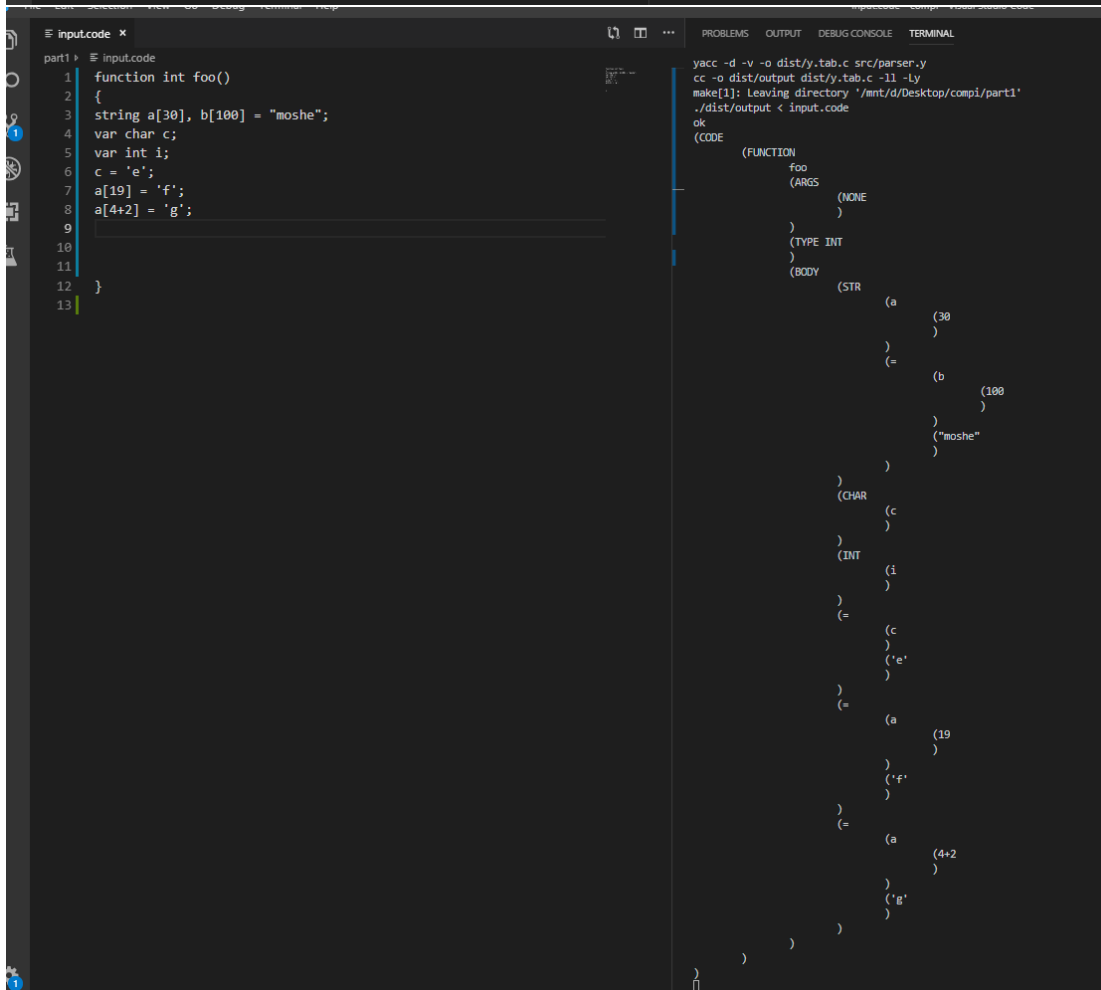
(CODE
(FUNCTION
foo
(ARGS
(INT
i
)
(j
)
(k
)
)
)
(TYPE INT
)
(BODY
(FUNCTION
square
(ARGS
(INT
t
)
)
)
(TYPE INT
)
)
(BODY
(INT
temp
)
)
(=
(temp
)
*
(t
)
)
)
)
(RET
0
)
)
)
(INT
total
)
)
(=
(total
)
1
)
)
(RET
total
)
)
)
```

דוגמאות להשמות עם טיפוסים char, string



```
1 function int foo()
2 {
3   var int i = 0;
4   var bool m = true, n = false, x;
5   var char c = 'a';
6 }
7
8
```

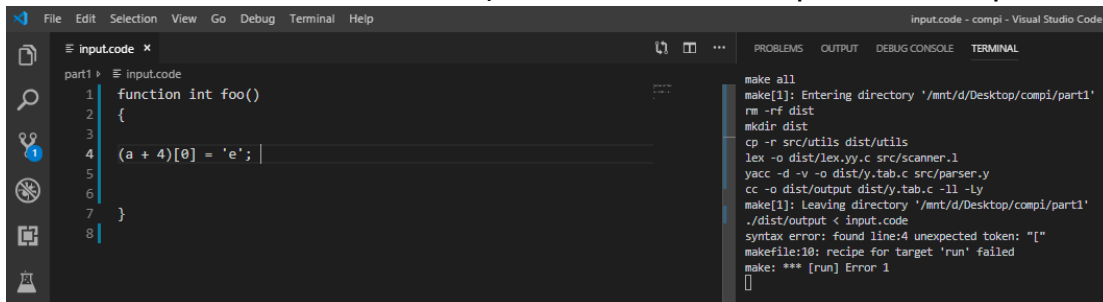
```
make all
make[1]: Entering directory '/mnt/d/Desktop/comp1/part1'
rm -rf dist
mkdir dist
cp -r src/utils dist/utils
lex -o dist/lex.yy.c src/scanner.l
yacc -d -v -o dist/y.tab.c src/parser.y
cc -o dist/output dist/y.tab.c -ll -ly
make[1]: Leaving directory '/mnt/d/Desktop/comp1/part1'
./dist/output < input.code
ok
(CODE
  (FUNCTION
    foo
    (ARGS
      (NONE)
    )
    (TYPE INT)
    (BODY
      (INT
        (=
          (i
            )
          )
          (0
            )
          )
        )
      (BOOL
        (=
          (m
            )
          )
          (TRUE
            )
          )
        )
      (=
        (n
          )
        )
        (FALSE
          )
        )
      (x
        )
      )
      (CHAR
        (=
          (c
            )
          )
          ('a'
            )
          )
        )
      )
    )
  )
)
```



```
1 function int foo()
2 {
3   string a[30], b[100] = "moshe";
4   var char c;
5   var int i;
6   c = 'e';
7   a[19] = 'f';
8   a[4+2] = 'g';
9 }
10
11
12 }
13
```

```
yacc -d -v -o dist/y.tab.c src/parser.y
cc -o dist/output dist/y.tab.c -ll -ly
make[1]: Leaving directory '/mnt/d/Desktop/comp1/part1'
./dist/output < input.code
ok
(CODE
  (FUNCTION
    foo
    (ARGS
      (NONE)
    )
    (TYPE INT)
    (BODY
      (STR
        (a
          (30
            )
          )
        )
        (=
          (b
            (100
              )
            )
          )
          ("moshe"
            )
          )
        )
      (CHAR
        (c
          )
        )
      (INT
        (i
          )
        )
      (=
        (c
          )
        )
        ('e'
          )
        )
      (=
        (a
          (19
            )
          )
        )
        ('f'
          )
        )
      )
      (=
        (a
          (4+2
            )
          )
        )
        ('g'
          )
        )
      )
    )
  )
)
```

שגיאה אין אפשרות להוסיף שום דבר לטיפוס מסוג array



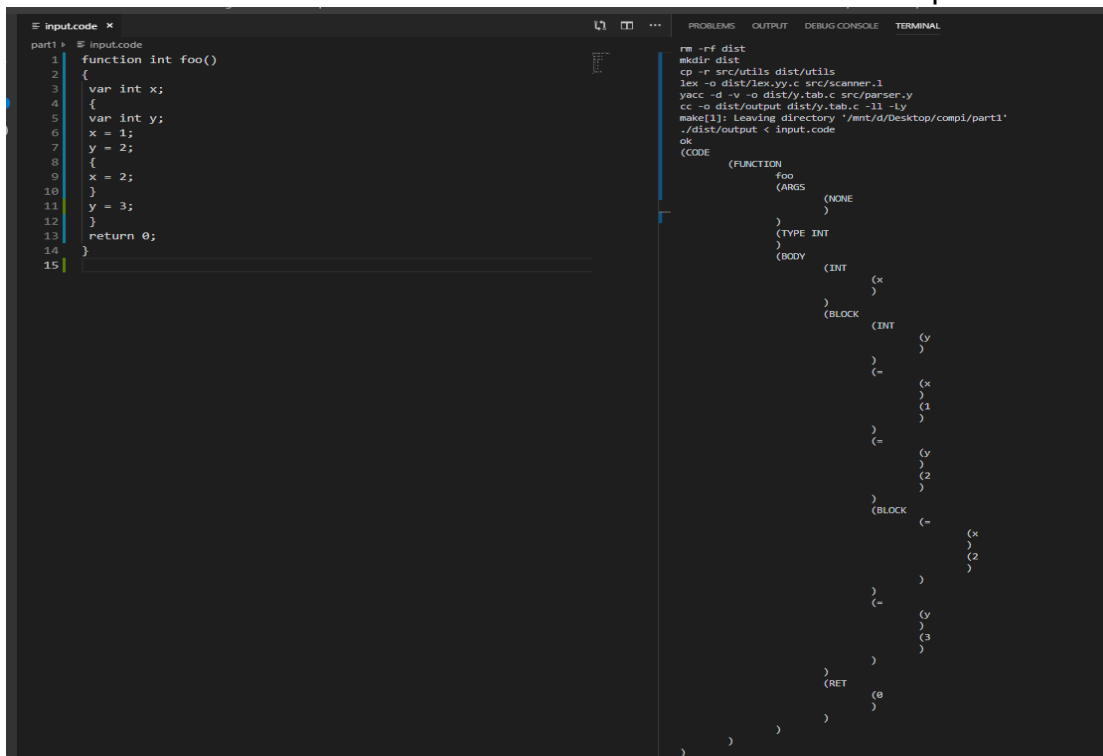
The screenshot shows the Visual Studio Code editor with a file named `inputcode`. The code in the editor is:

```
1 function int foo()
2 {
3
4 (a + 4)[0] = 'e';
5
6
7 }
8
```

The terminal output on the right shows the execution of a `make` command. It reports a syntax error on line 4:

```
make all
make[1]: Entering directory '/mnt/d/Desktop/compil/part1'
rm -rf dist
mkdir dist
cp -r src/utils dist/utils
lex -o dist/lex.yy.c src/scanner.l
yacc -d -v -o dist/y.tab.c src/parser.y
cc -o dist/output dist/y.tab.c -ll -ly
make[1]: Leaving directory '/mnt/d/Desktop/compil/part1'
./dist/output < inputcode
syntax error: found line:4 unexpected token: "["
makefile:10: recipe for target 'run' failed
make: *** [run] Error 1
```

דוגמאות לבלוקים

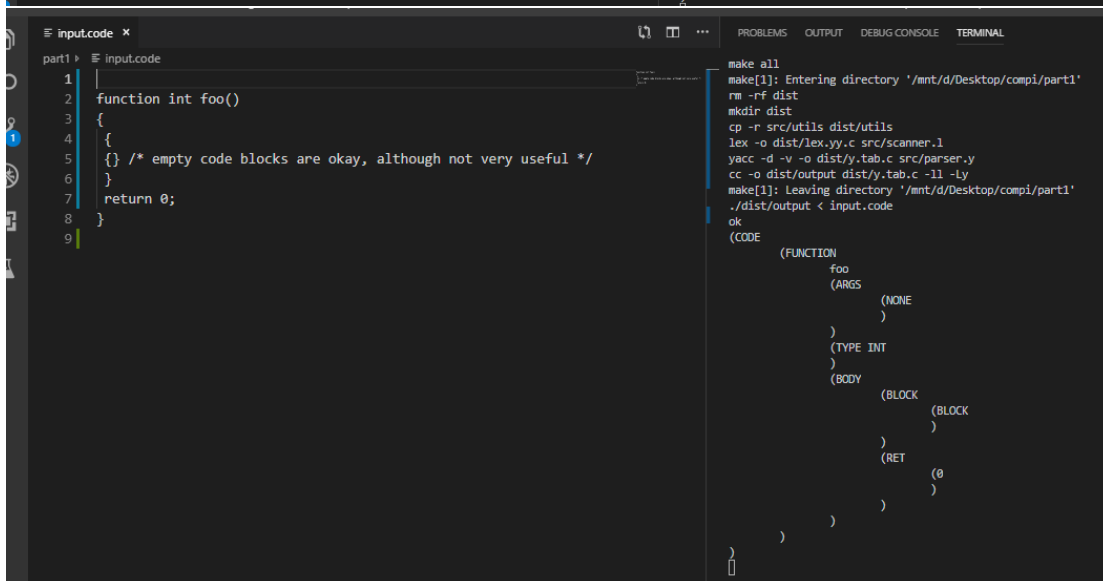


The screenshot shows the Visual Studio Code editor with a file named `inputcode`. The code in the editor is:

```
1 function int foo()
2 {
3   var int x;
4   {
5     var int y;
6     x = 1;
7     y = 2;
8     {
9       x = 2;
10    }
11   }
12   y = 3;
13   return 0;
14 }
15
```

The terminal output on the right shows the execution of a `make` command. It reports a syntax error on line 4:

```
rm -rf dist
mkdir dist
cp -r src/utils dist/utils
lex -o dist/lex.yy.c src/scanner.l
yacc -d -v -o dist/y.tab.c src/parser.y
cc -o dist/output dist/y.tab.c -ll -ly
make[1]: Leaving directory '/mnt/d/Desktop/compil/part1'
./dist/output < inputcode
ok
(CODE
  (FUNCTION
    foo
    (ARGS
      (NONE)
    )
    (TYPE INT
    )
    (BODY
      (INT
        (x)
      )
      (BLOCK
        (INT
          (y)
        )
        (~
          (x)
          (1)
        )
        (~
          (y)
          (2)
        )
        (BLOCK
          (~
            (x)
            (2)
          )
        )
        (~
          (y)
          (3)
        )
      )
      (RET
        (0)
      )
    )
  )
)
```



The screenshot shows the Visual Studio Code editor with a file named `inputcode`. The code in the editor is:

```
1 |
2 function int foo()
3 {
4 {
5 } /* empty code blocks are okay, although not very useful */
6 }
7 return 0;
8 }
9
```

The terminal output on the right shows the execution of a `make` command. It reports a syntax error on line 4:

```
make all
make[1]: Entering directory '/mnt/d/Desktop/compil/part1'
rm -rf dist
mkdir dist
cp -r src/utils dist/utils
lex -o dist/lex.yy.c src/scanner.l
yacc -d -v -o dist/y.tab.c src/parser.y
cc -o dist/output dist/y.tab.c -ll -ly
make[1]: Leaving directory '/mnt/d/Desktop/compil/part1'
./dist/output < inputcode
ok
(CODE
  (FUNCTION
    foo
    (ARGS
      (NONE)
    )
    (TYPE INT
    )
    (BODY
      (BLOCK
        (BLOCK
          (BLOCK
            (BLOCK
              (RET
                (0)
              )
            )
          )
        )
      )
    )
  )
)
```


שגיאה הכרזת משתנה אחרי הצהרה

```
File Edit Selection View Go Debug Terminal Help
input.code - compi - Visual Studio Code

part1 > input.code
1 function int foo()
2 {
3   var int x;
4   {
5     x = 1;
6     var int y;
7     /* must declare all variables before any statement */
8   }
9   return 0;
10 }
11 |

make all
make[1]: Entering directory '/mnt/d/Desktop/compi/part1'
rm -rf dist
mkdir dist
cp -r src/utils dist/utils
lex -o dist/lex.yy.c src/scanner.l
yacc -d -v -o dist/y.tab.c src/parser.y
cc -o dist/output dist/y.tab.c -ll -ly
make[1]: Leaving directory '/mnt/d/Desktop/compi/part1'
./dist/output < input.code
syntax error: found line:6 unexpected token: "var"
makefile:10: recipe for target 'run' failed
make: *** [run] Error 1
```

```
part1 > input.code
1 function int foo()
2 {
3   3 || 2;
4   (3 + 2) / 3 - 5 * 2;
5   true && false || false;
6   5;
7   3.234;
8   true;
9
10
11 }

rm -rf dist
mkdir dist
cp -r src/utils dist/utils
lex -o dist/lex.yy.c src/scanner.l
yacc -d -v -o dist/y.tab.c src/parser.y
cc -o dist/output dist/y.tab.c -ll -ly
make[1]: Leaving directory '/mnt/d/Desktop/compi/part1'
./dist/output < input.code
ok
(CODE
(FUNCTION
  foo
  (ARGS
    )
  )
  (TYPE INT
  )
  (BODY
    (||
      (3
      )
      (2
      )
    )
    (-
      (/
        (+
          (3
          )
          (2
          )
        )
        )
      )
      (*)
      (5
      )
      (2
      )
    )
    )
    (||
      (&&
        (TRUE
        )
        (FALSE
        )
      )
      )
      (FALSE
      )
    )
    (5
    )
    (3.234
    )
    (TRUE
    )
  )
)
)
```

```
part1 > input.code
1 function int foo()
2 {
3   -5;
4   *x;
5   *(p + 5);
6   !false;
7   a -- b;
8
9
10 }

make all
make[1]: Entering directory '/mnt/d/Desktop/compi/part1'
rm -rf dist
mkdir dist
cp -r src/utils dist/utils
lex -o dist/lex.yy.c src/scanner.l
yacc -d -v -o dist/y.tab.c src/parser.y
cc -o dist/output dist/y.tab.c -ll -ly
make[1]: Leaving directory '/mnt/d/Desktop/compi/part1'
./dist/output < input.code
ok
(CODE
(FUNCTION
  foo
  (ARGS
    (NONE
    )
  )
  (TYPE INT
  )
  (BODY
    (-5
    )
    (DREF
      (x
      )
    )
    (DREF
      (+
        (p
        )
        (5
        )
      )
    )
    )
    (NOT
      (FALSE
      )
    )
    )
    (==
      (a
      )
      (b
      )
    )
  )
)
)
```

דוגמאות ללולאות

inputcode

part1

```
1 function int foo()
2 {
3   while(true) { l = 2; k = 1 + j; }
4   if(true) i = 5;
5   if(true) { j = 3; } else x = x - 1;
6
7
8
9 }
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

(CODE)

(FUNCTION
foo
(ARGS
)
)
(TYPE INT
)
(BODY
(WHILE
(TRUE
)
(BLOCK (= (1
) (2
)
)
(= (x
) (+ (1
) (3
)
)
)
(IF
(TRUE
)
(= (i
) (5
)
)
(IF-ELSE
(TRUE
)
(BLOCK (= (j
) (3
)
)
(= (x
) (- (x
) (1
)
)
)
)

inputcode

part1

```
1 function int foo()
2 {
3   while(false) x = x + 1;
4   for (i=0; i<10; i=i + 2){ a = a + 1;}
5
6
7
8 }
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

(CODE)

(FUNCTION
foo
(ARGS
)
)
(TYPE INT
)
(BODY
(WHILE (FALSE
)
(= (x
) (+ (x
) (1
)
)
)
(FOR (= (1
) (0
)
)
(= (1
) (10
)
)
(= (i
) (+ (i
) (2
)
)
)
(BLOCK (= (a
) (+ (a
) (1
)
)
)
)
)

שגיאה בשפה אפשר לקחת כתובת של משתנה או array ו- x הוא ביטוי

```
part1 > input.code
1 function int foo()
2 {
3   x = &(x);
4   |
5
6 }
```

```
make all
make[1]: Entering directory '/mnt/d/Desktop/comp1/part1'
rm -rf dist
mkdir dist
cp -r src/utils dist/utils
lex -o dist/lex.yy.c src/scanner.l
yacc -d -v -o dist/y.tab.c src/parser.y
cc -o dist/output dist/y.tab.c -ll -ly
make[1]: Leaving directory '/mnt/d/Desktop/comp1/part1'
./dist/output < input.code
syntax error: found line:3 unexpected token: "("
makefile:10: recipe for target 'run' failed
make: *** [run] Error 1
```

```
make all
make[1]: Entering directory '/mnt/d/Desktop/comp1/part1'
rm -rf dist
mkdir dist
cp -r src/utils dist/utils
lex -o dist/lex.yy.c src/scanner.l
yacc -d -v -o dist/y.tab.c src/parser.y
cc -o dist/output dist/y.tab.c -ll -ly
make[1]: Leaving directory '/mnt/d/Desktop/comp1/part1'
./dist/output < input.code
ok
(CODE
(FUNCTION
  foo
  (ARGS
    (NONE
    )
  )
  (TYPE INT
  )
  (BODY
    (DO-WHILE
      (<=
        (i
        )
        (10
        )
      )
      (BLOCK
        (=
          (a
          )
          (+
            (a
            )
            (i
            )
          )
        )
        (=
          (i
          )
          (+
            (i
            )
            (1
            )
          )
        )
      )
    )
  )
)
)
```

שגיאה אין כתובת לביטוי

```
part1 > input.code
1 function int foo()
2 {
3   x = &(1+3);
4   |
5
6 }
```

```
make all
make[1]: Entering directory '/mnt/d/Desktop/comp1/part1'
rm -rf dist
mkdir dist
cp -r src/utils dist/utils
lex -o dist/lex.yy.c src/scanner.l
yacc -d -v -o dist/y.tab.c src/parser.y
cc -o dist/output dist/y.tab.c -ll -ly
make[1]: Leaving directory '/mnt/d/Desktop/comp1/part1'
./dist/output < input.code
syntax error: found line:3 unexpected token: "("
makefile:10: recipe for target 'run' failed
make: *** [run] Error 1
```