עזאלדין אלקרעאן : 207101429

עומאר חמדיה : 206635922

ריזק אבו מדיעים : 211606801

קלט:

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

func goo() return char{
    return 's';
}
proc Main()
{
```

פלט:

```
(CODE
    (
        (PROC
            foo
            (ARGS
                ( INT x y z  )
                ( REAL f  )
            )
            (BLOCK
                (IF-ELSE
                    ( > x y )
                    (BLOCK
                        (=
                            x
                            ( + x f )
                        )
                    )
                    (BLOCK
                        (
                            (
                                (=
                                    y
                                    (+
                                        x
                                        ( + y z )
                                    )
                                )
                                (=
                                    x
                                    ( * f 2 )
                                )
                            )
                            ( = z f )
                        )
                    )
                )
            )
        )
        (FUNC
            goo
            ARGS NONE
            ( RET CHAR )
            (BLOCK
                ( RET 's' )
            )
        )
    )
    ( MAIN BLOCK NONE )
)
```

קלט:

```
1 func foo() return int
2 {
3  return 0;
4 }
5 proc Main()
6 {
7  var a : int;
8  a = foo();
9 }
```

פלט:

```
(CODE
      (FUNC
            foo
            ARGS NONE
            ( RET INT )
            (BLOCK
                  ( RET 0 )
            )
      )
      (MAIN
            (BLOCK
                  (
                        ( INT a  )
                        (=
                              a
                              ( FUNC CALL foo )
                        )
                  )
            )
      )
)
```

קלט:

```
1 proc Main()
2 {
3  var a : int;
4  a = foo();
5 }
6 func foo() return int
7 {
8  return 0;
9 }
```

פלט:

```
(CODE
      (MAIN
            (BLOCK
                  (
                        ( INT a  )
                        (=
                              a
                              ( FUNC CALL foo
                        )
                  )
            )
      )
)
syntax error , line number: 6
parser caused by: 'func'
```

קלט:

```
func foo(i, j, k : int) return int
{
 func fee(l, m, n : int) return bool
 {
 return true;
 }
 return 0;
}
proc goo(i, j, k : int)
{
 func fee(l, m, n : int) return bool
 {
 return true;
 }
 fee(2,3,4);
}
proc Main (){
```

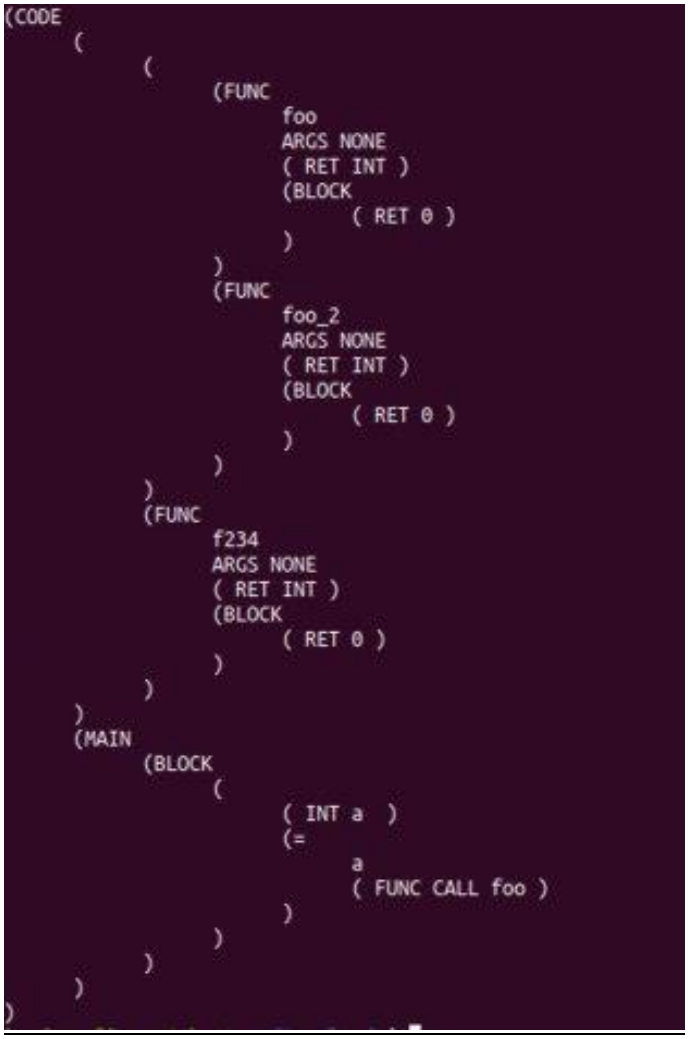
פלט:

```
(CODE
    (
        (FUNC
            foo
            ( INT i j k  )
            ( RET INT )
            (BLOCK
                (FUNC
                    fee
                    ( INT l m n  )
                    ( RET BOOL )
                    (BLOCK
                        ( RET BOOL )
                    )
                )
                ( RET 0 )
            )
        )
        (PROC
            goo
            ( INT i j k  )
            (BLOCK
                (
                    (FUNC
                        fee
                        ( INT l m n  )
                        ( RET BOOL )
                        (BLOCK
                            ( RET BOOL )
                        )
                    )
                    (FUNC CALL
                        fee
                        (
                            (
                                (  2 )
                                3
                            )
                            4
                        )
                    )
                )
            )
        )
    )
    ( MAIN BLOCK NONE )
)
```

קלט:

```
1 func foo() return int { return 0; }
2 func foo_2() return int { return 0; }
3 func f234() return int { return 0;}
4
5 proc Main()
6 {
7  var a : int;
8  a = foo();
```

פלט:



קלט:

```
1 func 9foo() return int { return 0; }
2 func _rip() return int { return 0; }
3 proc Main()
4 {
5  var a : int;
6  a = foo();
7 }
```

פלט:

```
syntax error , line number: 1
parser caused by: '9'
```

קלט:

```
1 proc foo(i, j, k) { } /% no type defined %/
2 proc foo(i j k : int) { } /% IDs must be separated by comma %/
3 proc Main()
4 {
5  var a : int;
6  a = foo();
7 }
```

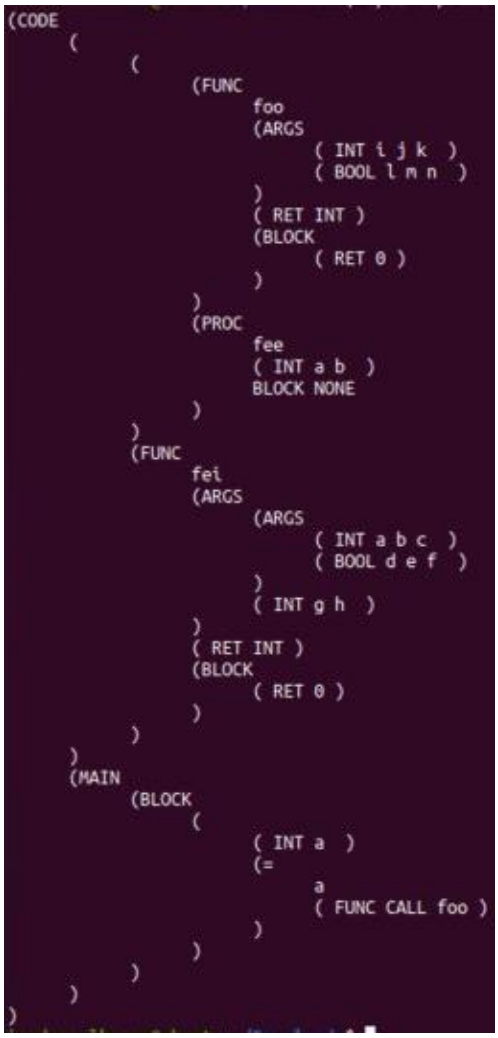
פלט:

```
syntax error , line number: 1
parser caused by: ')'
```

קלט:

```
1 func foo(i, j, k : int; l, m, n : bool) return int { return 0; }
2 proc fee(a, b : int) { }
3 func fei(a, b, c : int; d, e, f : bool; g, h : int) return int { return
4 0; }
5 proc Main()
6 {
7  var a : int;
8  a = foo();
9 }
```

פלט:



קלט:

```
func foo(i, j, k : int) return int
{
 func square(t : int) /% func/proc declarations %/ return int
 {
 var temp : int;
 temp = t*t;
 return 0;
 }
 var total : int; /% variable declarations %/
 total = 1; /% statements %/
 return total;
}
proc Main()
{
 a = foo();
```

פלט:

```
(CODE
      (FUNC
            foo
            ( INT i j k  )
            ( RET INT )
            (BLOCK
                  (
                        (
                              (FUNC
                                    square
                                    ( INT t  )
                                    ( RET INT )
                                    (BLOCK
                                          (
                                                ( INT temp  )
                                                (=
                                                      temp
                                                      ( * t t )
                                                )
                                          )
                                          ( RET 0 )
                                    )
                              )
                              ( INT total  )
                        )
                        ( = total 1 )
                  )
                  ( RET total )
            )
      )
      (MAIN
            (BLOCK
                  (=
                        a
                        ( FUNC CALL foo )
                  )
            )
      )
)
```

קלט:

```
func foo() return int
{
 var x : int;
 {
 var y : int;
 x = 1;
 y = 2;
 {
 x = 2;
 }
 y = 3;
 }
 return 0;
}
func foo() return int
{
 {
 {} /% empty code blocks are okay, although not very useful %/
 }
 return 0;
}
proc Main()
{
 a = foo();
}
```

פלט:

```
(CODE
    (
        (FUNC
            foo
            ARGS NONE
            ( RET INT )
            (BLOCK
                (
                    ( INT x  )
                    (BLOCK
                        (
                            ( INT y  )
                            (
                                (
                                    (
                                        ( = x 1 )
                                        ( = y 2 )
                                    )
                                    (BLOCK
                                        ( = x 2 )
                                    )
                                )
                                ( = y 3 )
                            )
                        )
                    )
                )
                ( RET 0 )
            )
        )
        (FUNC
            foo
            ARGS NONE
            ( RET INT )
            (BLOCK
                ( BLOCK BLOCK NONE )
                ( RET 0 )
            )
        )
    )
    (MAIN
        (BLOCK
            (=
                a
                ( FUNC CALL foo )
            )
        )
    )
)
```

קלט:

```
1 func foo() return int
2 {
3  var x : int;
4  {
5  x = 1;
6  var y : int;
7 /% must declare all variables before any statement %/
8  }
9  return 0;
0 }
1 proc Main()
2 {
3  a = foo();
4 }
```

פלט:

```
syntax error , line number: 6
parser caused by: 'var'
```

קלט:

```
1 func foo() return int { return 0; }
2 func foo_2() return int { var a: int; a = 2; return a; }
3 func foo_3() return int { if (true) { return foo(); } return 0; }
4 proc Main()
5 {
6  a = foo();
7 }
```

פלט:

```
(CODE
    (
        (
            (FUNC
                foo
                ARGS NONE
                ( RET INT )
                (BLOCK
                    ( RET 0 )
                )
            )
            (FUNC
                foo_2
                ARGS NONE
                ( RET INT )
                (BLOCK
                    (
                        ( INT a )
                        ( = a 2 )
                    )
                    ( RET a )
                )
            )
        )
        (FUNC
            foo_3
            ARGS NONE
            ( RET INT )
            (BLOCK
                (IF
                    BOOL
                    (BLOCK
                        (RET
                            ( FUNC CALL foo )
                        )
                    )
                )
                ( RET 0 )
            )
        )
    )
    (MAIN
        (BLOCK
            (=
                a
                ( FUNC CALL foo )
            )
        )
    )
)
```

קלט:

```
1 func foo_3() return int {return true; }
2 func foo_3() return int {if (true) {return foo(); } }
3 proc Main()
4 {
5  a = foo();
6 }
```

פלט:

```
syntax error , line number: 2
parser caused by: '}'
```

קלט:

```
1 proc Main()
2 {
3  a = foo();
4 }
5 proc Main()
6 {
7  var a : int;
8  a = foo();
9 }
```

פלט:

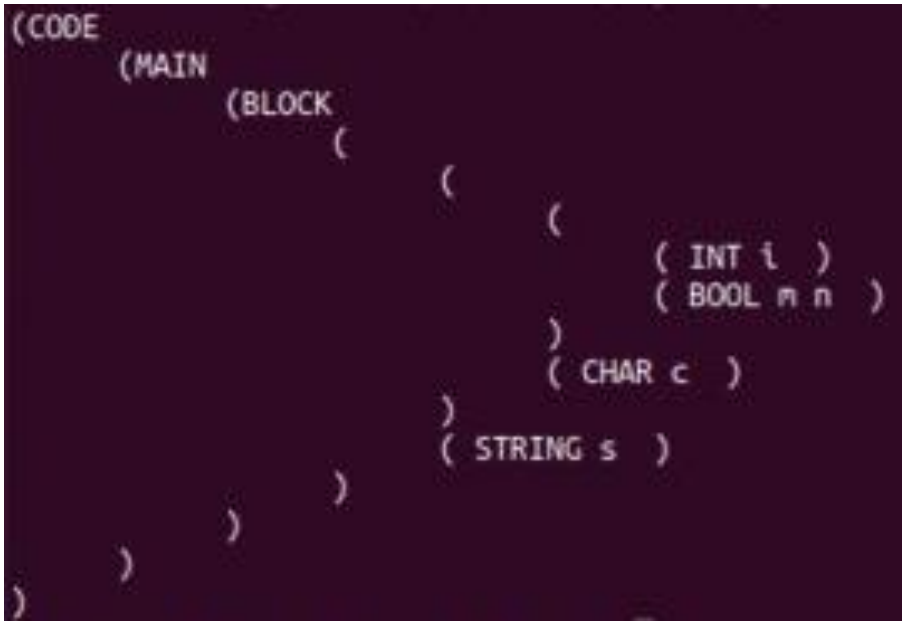
```
(CODE
      (MAIN
            (BLOCK
                  (=
                        a
                        ( FUNC CALL foo )
                  )
            )
      )
)
syntax error , line number: 5
parser caused by: 'proc'
```

קלט:

```
proc Main()
{
var i : int;
var m, n : bool;
var c : char;
var s : string[20];

}
```

פלט:



קלט:

```
1 proc Main()
2 {
3 var i = 5 : int;
4 var i : int;
5 var m, n : bool;
6 var c : char;
7 var s : string[20];
8 }
```

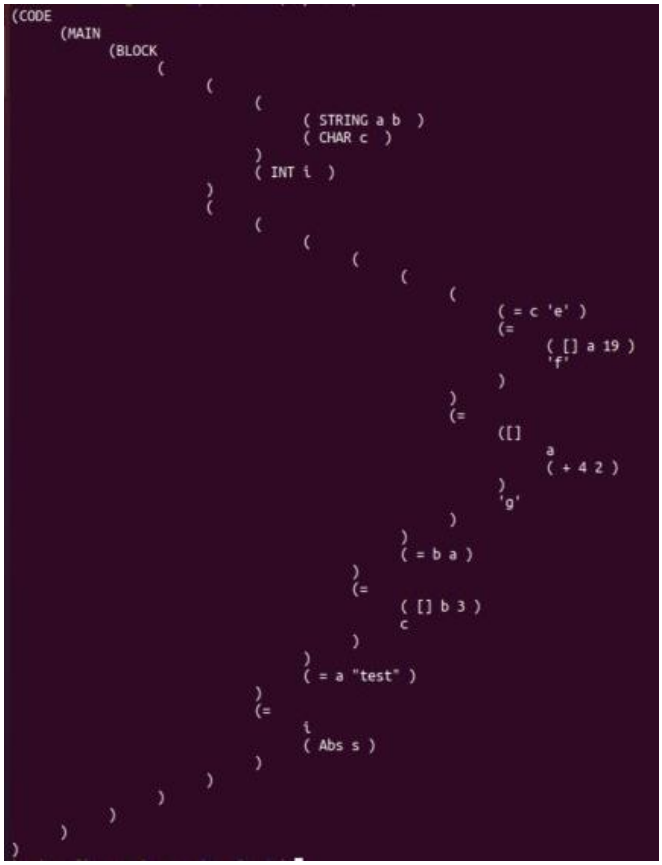
פלט:

```
syntax error , line number: 3
parser caused by: '='
```

קלט:

```
proc Main()
{
var a, b : string[100];
var c : char;
var i: int;
c = 'e';
a[19] = 'f';
a[4+2] = 'g';
b = a;
b[3] = c;
a = "test"; /% basically equivalent to a[0] = 't'; a[1] = 'e';
a[2] = 's'; a[3] = 't'; a[4] = '\0'; %/
i = |s|; /% this assigns 100 to variable i, since the length
operator returns the size of the character array %/
}
```

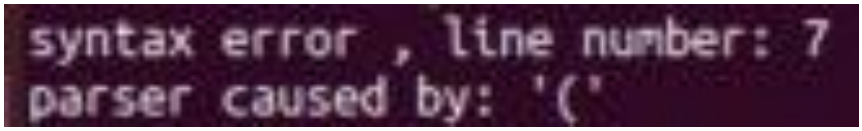
פלט:



קלט:

```
1 proc Main()
2 {
3 var a, b : string[100];
4 var c : char;
5 c = 'e'; /% everything up to this is OK %/
6 c = a; /% type mismatch, can't assign string type to character type %/
7 (a + 4)[0] = 'e';
8  /% cannot add anything to array elements - they are not pointers %/
9 }
```

פלט:



קלט:

```
proc Main()
{
if(3 > 2)
{
 /%...statements...%/
 i = 5; /% i has been declared above%/
}
if(true) { j = 3; } else { k = 4; }
while(true) { l = 2; k = l + j; }
if(true) i = 5;
if(true) { j = 3; } else x = x -1;
while(false) x = x + 1;
}
```

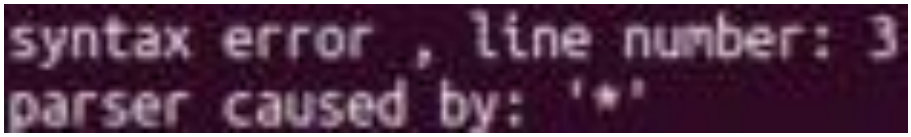
פלט:

```
(CODE
    (MAIN
        (BLOCK
            (
                (
                    (
                        (
                            (
                                (IF
                                    ( > 3 2 )
                                    (BLOCK
                                        ( = i 5 )
                                    )
                                )
                                (IF-ELSE
                                    BOOL
                                    (BLOCK
                                        ( = j 3 )
                                    )
                                    (BLOCK
                                        ( = k 4 )
                                    )
                                )
                            )
                            (WHILE
                                BOOL
                                (BLOCK
                                    (
                                        ( = l 2 )
                                        (=
                                            k
                                            ( + l j )
                                        )
                                    )
                                )
                            )
                        )
                        (IF
                            BOOL
                            ( = i 5 )
                        )
                    )
                    (IF-ELSE
                        BOOL
                        (BLOCK
                            ( = j 3 )
                        )
                        (=
                            x
                            ( - x 1 )
                        )
                    )
                )
                (WHILE
                    BOOL
                    (=
                        x
                        ( + x 1 )
                    )
                )
            )
        )
    )
)
```

קלט:

```
proc Main()
{
var x : bool*; /% no such pointer type %/
x = &(1+3);
var x : char;
var y : int*;
y = &x; /% address of x is of type char* %/
var x : char*;
var y : char;
x = &(&y);
/% can only take the address of variable or array element, and (&y) is
an expression %/
}
```

פלט:



קלט:

```
1 proc Main()
2 {
3 while(x>10){
4 while(x<10){
5 if(x){
6 x=20;
7 }
8 else{
9 x=30;}
10 }
11 }
12 }
```

פלט:

```
(CODE
    (MAIN
        (BLOCK
            (WHILE
                ( > x 10 )
                (BLOCK
                    (WHILE
                        ( < x 10 )
                        (BLOCK
                            (IF-ELSE
                                x
                                (BLOCK
                                    ( = x 20 )
                                )
                                (BLOCK
                                    ( = x 30 )
                                )
                            )
                        )
                    )
                )
            )
        )
    )
)
```

קלט:

```
proc Main()
{
if (y){
if(x){
x=20;
}
else{
x=30;}
}
}
```

פלט:

```
(CODE
      (MAIN
            (BLOCK
                  (IF
                        y
                        (BLOCK
                              (IF-ELSE
                                    x
                                    (BLOCK
                                          ( = x 20 )
                                    )
                                    (BLOCK
                                          ( = x 30 )
                                    )
                              )
                        )
                  )
            )
      )
)
```

קלט:

```
1 func foo() return int
2 {
3  var x : int;
4 func foo() return int
5 {
6  {
7  {} /% empty code blocks are okay, although not very useful %/
8  while(x>10){
9 while(x<10){
10 if(x){
11 x=20;
12 }
13 else{
14 x=30;}
15 }
16 }
17  }
18  return 0;
19 }
20  return 0;
21 }
22
23 proc Main()
24 {
25
26 }
```

פלט:

```
(CODE
    (FUNC
        foo
        ARGS NONE
        ( RET INT )
        (BLOCK
            (
                ( INT x  )
                (FUNC
                    foo
                    ARGS NONE
                    ( RET INT )
                    (BLOCK
                        (BLOCK
                            (
                                BLOCK NONE
                                (WHILE
                                    ( > x 10 )
                                    (BLOCK
                                        (WHILE
                                            ( < x 10 )
                                            (BLOCK
                                                (IF-ELSE
                                                    x
                                                    (BLOCK
                                                        ( = x 20 )
                                                    )
                                                    (BLOCK
                                                        ( = x 30 )
                                                    )
                                                )
                                            )
                                        )
                                    )
                                )
                            )
                        )
                        ( RET 0 )
                    )
                )
            )
            ( RET 0 )
        )
    )
    ( MAIN BLOCK NONE )
)
```