

Project Αρχές Γλωσσών

2020-2021

Ομάδα:

| | | | | |
|---------------|--------------------------|--------------------|--------------------------|--------------------|
| Όνοματεπώνυμο | Καπελετιώτης Χαρίλαος | Κοκκάλα Βασιλική | Κυριακόπουλος Ιωάννης | Τσαβέα Μαρία |
| ΑΜ | 1057772 | 1067519 | 1070930 | 1067534 |
| Έτος | 5 _ο | 3 _ο | 3 _ο | 3 _ο |
| Mail | up1057772@upnet.gr | up1067519@upnet.gr | up1070930@upnet.gr | up1067534@upnet.gr |

BNF

Program:

PROGRAM

WORD BLOCK NEWLINE

Commands

MainFunction

Comments

ENDBLOCK

;

String:

EAR InsideString EAR

;

InsideString:

WORD InsideString

|TEXT InsideString

|NUMBER InsideString

|/*empty*/

;

ArrStr:

WORD LBRCK NUMBER RBRCK

;

Assignment:

IntAssign SEMICOLON

;

Array:

WORD LBRCK NUMBER RBRCK

;

Commands:

Comments Commands

|WORD LPAR CPar RPAR SEMICOLON Commands

|ProgramVariables Commands

|Structs Commands

|Functions Commands

|LINE_COMMENT Commands

/* empty */

;

Case:

CASE UP_AND_DOWN_DOT WORD IntoTheLoops

| CASE UP_AND_DOWN_DOT WORD IntoTheLoops Case

| CASE UP_AND_DOWN_DOT NUMBER IntoTheLoops

| CASE UP_AND_DOWN_DOT NUMBER IntoTheLoops Case

| CASE UP_AND_DOWN_DOT String IntoTheLoops

| CASE UP_AND_DOWN_DOT String IntoTheLoops Case

;

Condition:

LPAR Comparisons RPAR

;

Comparisons:

WORD EQUALS WORD

| WORD NOT_EQUAL WORD

| WORD LESS WORD

| WORD MORE WORD

| WORD AND WORD

| WORD OR WORD

| WORD EQUALS NUMBER
| WORD NOT_EQUAL NUMBER
| WORD LESS NUMBER
| WORD MORE NUMBER
| WORD AND NUMBER
| WORD OR NUMBER
| NUMBER EQUALS WORD
| NUMBER NOT_EQUAL WORD
| NUMBER LESS WORD
| NUMBER MORE WORD
| NUMBER AND WORD
| NUMBER OR WORD
;

Comments:

COMMENTS_START IntoTheComments COMMENTS_END
|/*empty*/
;

IntoTheComments:

WORD IntoTheComments
|TEXT IntoTheComments
|NEWLINE IntoTheComments
|NUMBER IntoTheComments
|CHAR IntoTheComments
|VARS IntoTheComments
|PROGRAM IntoTheComments

```
|INTEGER IntoTheComments
|FUNCTION IntoTheComments
|STRUCT IntoTheComments
|ENDSTRUCT IntoTheComments
|PRINT IntoTheComments
|ENDBLOCK IntoTheComments
|SEMICOLON IntoTheComments
|Print IntoTheComments
|LPAR IntoTheComments
|RPAR IntoTheComments
|BLOCK IntoTheComments
|TYPEDEF IntoTheComments
|Loops IntoTheComments
|Variables IntoTheComments
|/* empty */
;
```

CPar:

```
/*empty*/
|COMMA CPar
|NUMBER CPar
|WORD CPar
|WORD LBRCK RBRCK CPar
;
```

CreateStruct:

```
STRUCT WORD NEWLINE WORD SET BLOCK CreatStructParams ENDBLOCK
```

```
|STRUCT WORD NEWLINE WORD
|WORD DOT WORD SET ToBeSet SEMICOLON
|WORD DOT WORD LBRCK NUMBER RBRCK SET ToBeSet
;
```

ToBeSet:

```
WORD
| NUMBER
| String
|WORD DOT WORD
|WORD DOT WORD LBRCK NUMBER
;
```

CreatStructParams:

```
WORD CreatStructParams
|WORD LBRCK NUMBER RBRCK CreatStructParams
| NUMBER CreatStructParams
| COMMA CreatStructParams
| String CreatStructParams
| /* empty */
;
```

Declaration:

```
VARS INTEGER IDec SEMICOLON
|VARS CHAR StrDec SEMICOLON
;
```

Expression:

Paragontas Operator Paragontas Expression

|Operator Paragontas

|Operator Expression

|LPAR Expression RPAR

/*empty*/

;

Paragontas:

NUMBER

|WORD

;

Operator:

ADD

|MIN

|MUL

|POW

|AND

|OR

|DIV

;

For:

FOR WORD UP_AND_DOWN_DOT SET NUMBER TO NUMBER STEP NUMBER

IntoTheLoops ENDFOR

;

Functions:

FUNCTION WORD LPAR Parenthesis RPAR NEWLINE

ProgramInstructions

RETURN

ReturnTypes

SEMICOLON

END_FUNCTION

;

FunctionCall:

WORD

LPAR CPar RPAR

;

IDec:

|WORD IDec

|Array IDec

|COMMA IDec

;

IntAssign:

WORD SET NUMBER

| WORD SET WORD

| WORD LBRCK NUMBER RBRCK SET

| WORD SET Expression

| WORD SET FunctionCall


```
| FunctionCall  
| WORD SET String  
;
```

IntoTheLoops:

```
Variables IntoTheLoops  
| Loops IntoTheLoops  
| Print IntoTheLoops  
| BREAK SEMICOLON IntoTheLoops  
| /* empty */  
| LINE_COMMENT IntoTheLoops  
;
```

If:

```
IF Condition THEN IntoTheLoops ENDIF  
| IF Condition THEN IntoTheLoops MoreThanOneElse ELSE IntoTheLoops ENDIF  
| IF Condition THEN IntoTheLoops ELSE IntoTheLoops ENDIF  
;
```

Loops:

```
If | While | Switch | For  
;
```

MainFunction:

```
STARTMAIN  
ProgramInstructions
```

ENDMAIN

;

MoreThanOneElseIf:

ELSEIF Condition THEN IntoTheLoops

|ELSEIF Condition THEN IntoTheLoops MoreThanOneElseIf

;

ProgramInstructions:

ProgramVariables ProgramInstructions

|CreateStruct ProgramInstructions

|Loops ProgramInstructions

|LINE_COMMENT ProgramInstructions

|Print ProgramInstructions

/* empty */

;

Print:

PRINT LPAR Inside_Print RPAR SEMICOLON

;

Inside_Print:

|String Inside_Print

|WORD Inside_Print

|NUMBER Inside_Print

|COMMA Inside_Print

;

Parenthesis:

/* empty */

| COMMA Parenthesis

| WORD Parenthesis

| WORD LBRCK RBRCK Parenthesis

;

ProgramVariables:

Variables ProgramVariables

| /* empty */

;

ReturnTypes:

WORD

| NUMBER

| String

;

Structs:

AStruct

| TStruct

;

AStruct:

STRUCT WORD NEWLINE StructVariables ENDSTRUCT

;

TStruct:

TYPEDEF STRUCT WORD NEWLINE StructVariables WORD ENDSTRUCT

;

StructVariables:

StructDecsl StructVariables

| StructDecsC StructVariables

| /* empty */

;

StructDecsC:

VARs

CHAR `

Types SEMICOLON

;

StructDecsl:

VARs

INTEGER

Types SEMICOLON

;

Types:

WORD Types

| WORD LBRCK NUMBER RBRCK Types

| COMMA Types

```
| /* empty */
```

```
;
```

StrDec:

```
ArrStr
```

```
| ArrStr COMMA StrDec
```

```
| WORD
```

```
| WORD COMMA StrDec
```

```
| StrDec COMMA StrDec
```

```
;
```

Switch:

```
SWITCH LPAR WORD RPAR Case DEFAULT UP_AND_DOWN_DOT IntoTheLoops ENDSWITCH
```

```
;
```

Variables:

```
Declaration
```

```
| Assignment
```

```
;
```

While:

```
WHILE Condition IntoTheLoops ENDWHILE
```

```
;
```

ΠΑΡΑΔΟΧΕΣ :

- Argument counter για τα ορίσματα των συναρτήσεων κατά το κάλεσμα τους στην main.

- Type check, δηλαδή έλεγχος του type κατά την ανάθεση τιμών σε μεταβλητές.
- Έλεγχος ύπαρξης των structs.
- Έλεγχος ύπαρξης των αντικειμένων των structs.
- Έλεγχος ύπαρξης μεταβλητών.
- Έλεγχος ύπαρξης συνάρτησης.
- Απαγόρευση κλήσης συνάρτησης σε global field.
- Έλεγχος υπέρβασης ορίου ενός πίνακα.
- Έλεγχος δήλωσης ονομάτων στα structs κατά την χρήση του typedef.

SCREENSHOTS ΠΑΡΑΔΕΙΓΜΑΤΩΝ ΕΦΑΡΜΟΓΗΣ (ΕΠΙΤΥΧΗΜΕΝΗΣ ΚΑΙ ΜΗ ΑΝΑΛΥΣΗΣ)

Παράδειγμα 1^ο: Σωστή σύνταξη της PROGRAM.

The image shows a side-by-side comparison of a program and its compilation output. On the left, a code editor window titled 'sample.txt' displays a program with the following structure: a 'PROGRAM ANAFORA' block containing a comment '/* eimai sxolio agnohse me */', followed by a 'STARTMAIN' block, then a 'VARS' block with two integer variables 'var1' and 'var2', and finally an 'ENDMAIN' block. On the right, the compilation output is shown, indicating 'The Program Produced 0 Errors' and 'The program compiled Successfully!'. The output also lists the variables and their types: 'Variable => var2 Type => INTEGER' and 'Variable => var1 Type => INTEGER'. The output is formatted with dashed lines separating sections for 'VARS', 'FUNCS', and 'STRUCTS'.

```

sample.txt
1  PROGRAM ANAFORA
2
3      /* eimai sxolio
4      agnohse me
5      */
6
7  STARTMAIN
8
9
10     VARS INTEGER var1;
11     VARS INTEGER var2;
12
13
14 ENDMAIN
15

The Program Produced 0 Errors.
The program compiled Successfully!

-----VARS-----
Variable => var2      Type => INTEGER
Variable => var1      Type => INTEGER

-----FUNCS-----
-----STRUCTS-----

```

```

sample.txt
1 PROGRAM
2 ANAFORA{
3
4     /* eimai sxolio
5     agnohse me
6     */
7
8 STARTMAIN
9
10
11     VARS INTEGER var1;
12     VARS INTEGER var2;
13
14
15 ENDMAIN
16 }

```

At line: 2 error0: Unrecognized reserved word or character sequence

Εδώ έχουμε πρόβλημα αφού το όνομα του προγράμματος δεν βρίσκεται δίπλα από τη δεσμευμένη λέξη PROGRAM, αλλά στην επόμενη γραμμή.

```

sample.txt
1 PROGRAM ANAFORA{ /* eimai sxolio
2     agnohse me
3     */
4
5 STARTMAIN
6
7
8     VARS INTEGER var1;
9     VARS INTEGER var2;
10
11
12 ENDMAIN
13 }

```

At line: 1 error0: Unrecognized reserved word or character sequence

Εδώ έχουμε πρόβλημα αφού η αγκύλη { δεν ακολουθείται από αλλαγή γραμμής.

Παράδειγμα 2^ο: Έλεγχος σωστής λειτουργίας της FUNCTION.

```
sample.txt
1 PROGRAM ANAFORA{
2
3
4 FUNCTION ADD(a,b)
5     VARS INTEGER sum;
6     VARS CHAR OKAY,HI;
7     sum = a+b;
8     RETURN sum;
9 END_FUNCTION
10
11 STARTMAIN
12
13
14     VARS INTEGER var1;
15     VARS INTEGER var2;
16
17
18 ENDMAIN
19 }
```

The Program Produced 0 Errors.
The program compiled Successfully!

-----VARS-----

Variable => var2 Type => INTEGER
Variable => var1 Type => INTEGER

Variable => OKAY Type => CHAR
Variable => HI Type => CHAR

Variable => sum Type => INTEGER

-----FUNCS-----

FUNCTION => ADD
Arguments=> b
Arguments=> a

-----STRUCTS-----

```
sample.txt
1 PROGRAM ANAFORA{
2
3
4 FUNCTION ADD(a,b)
5     VARS INTEGER sum
6     VARS CHAR OKAY,HI;
7     sum = a+b;
8     RETURN sum;
9 END_FUNCTION
10
11 STARTMAIN
12
13
14     VARS INTEGER var1;
15     VARS INTEGER var2;
16
17
18 ENDMAIN
19 }
```

At line: 6error0: Unrecognized reserved word or character sequence

Στη γραμμή 5 λείπει το (;) επομένως εμφανίζεται error.


```
sample.txt
1 PROGRAM ANAFORA{
2
3
4 FUNCTION ADD(a,b)
5     VARS INTEGER sum;
6     VARS CHAR OKAY,HI;
7     VARS CHAR var1,var3[3];
8     sum = a+b;
9     RETURN sum;
10 END_FUNCTION
11
12 STARTMAIN
13
14
15     VARS INTEGER var4;
16     VARS INTEGER var5;
17
18
19 ENDMAIN
20 }
```

```
The Program Produced 0 Errors.
The program compiled Successfully!
-----VARS-----
Variable => var5      Type => INTEGER
Variable => var4      Type => INTEGER
Variable => var1      Type => CHAR
Variable => var3      Type => CHAR
Variable => OKAY      Type => CHAR
Variable => HI        Type => CHAR
Variable => sum       Type => INTEGER
-----
-----FUNCS-----
FUNCTION => ADD
Arguments=> b
Arguments=> a
-----
-----STRUCTS-----
-----
```

Στην γραμμή 7 υπάρχει η δυνατότητα ορισμού πινάκων με τη δήλωση του μήκους εντός [].

```
sample.txt
1 PROGRAM ANAFORA{
2
3
4 FUNCTION ADD(a,b)
5     VARS INTEGER sum;
6     VARS CHAR OKAY,HI;
7     VARS CHAR var1,var3[3];
8     sum = a+b;
9     RETURN sum;
10
11
12 STARTMAIN
13
14
15     VARS INTEGER var4;
16     VARS INTEGER var5;
17
18
19 ENDMAIN
20 }
```

```
At line: 12error0: Unrecognized reserved word or
character sequence
```

Λείπει η δεσμευμένη λέξη END_FUNCTION οπότε προκύπτει error.

```

sample.txt
1  PROGRAM ANAFORA{
2
3
4  ✓ FUNCTION ADD(a,b)
5      VARS INTEGER sum;
6      VARS CHAR OKAY,HI;
7      VARS CHAR var1,var3[3];
8      sum = a+b;
9
10
11  END_FUNCTION
12
13  ✓ STARTMAIN
14
15
16      VARS INTEGER var4;
17      VARS INTEGER var5;
18
19
20  ENDMAIN
21  }

```

At line: 11 error0: Unrecognized reserved word or character sequence

Λείπει η δεσμευμένη λέξη RETURN οπότε προκύπτει error.

Παράδειγμα 3^ο: Έλεγχος σωστής λειτουργίας της STRUCT.

```
sample.txt
1  PROGRAM ANAFORA{
2
3  STRUCT BASE
4      VARS INTEGER OK,var22,var22e[15];
5      VARS CHAR ANTIK;
6      VARS CHAR IM_OKAY[10];
7  ENDSTRUCT
8
9
10 STARTMAIN
11
12 VARS INTEGER var1;
13 VARS INTEGER var2;
14
15 ENDMAIN
16
17 }
18
```


```
The Program Produced 0 Errors.
The program compiled Successfully!
-----VARS-----
Variable => var2          Type => INTEGER
Variable => var1          Type => INTEGER
-----
-----FUNCS-----
-----STRUCTS-----
Struct => BASE, _INSTANCE: (null)
Variable => IM_OKAY
Variable => ANTIK
Variable => OK
Variable => var22
Variable => var22e
-----
```

```
sample.txt
1  PROGRAM ANAFORA{
2
3  STRUCT BASE
4      VARS INTEGER OK,var22,var22e[15];
5      VARS CHAR ANTIK;
6      VARS CHAR IM_OKAY[10];
7
8
9
10 STARTMAIN
11
12 VARS INTEGER var1;
13 VARS INTEGER var2;
14
15 ENDMAIN
16
17 }
18
```

At line: 10error0: Unrecognized reserved word or character sequence

Λείπει η δεσμευμένη λέξη ENDSTRUCT οπότε προκύπτει error.

Ας δοκιμάσουμε τώρα και τον δεύτερο τρόπο με την δεσμευμένη λέξη TYPEDEF.


sample.txt

```

1 PROGRAM ANAFORA{
2
3     TYPEDEF STRUCT BASE2
4         VARS INTEGER OK1;
5         VARS CHAR ANTIK2;
6         VARS CHAR IM_OKA4Y[10];
7     BASE2 ENDSTRUCT
8
9     STARTMAIN
10
11     VARS INTEGER var1;
12     VARS INTEGER var2;
13
14     ENDMAIN
15
16 }
```

```

The Program Produced 0 Errors.
The program compiled Successfully!

-----VARS-----
Variable => var2          Type => INTEGER
Variable => var1          Type => INTEGER

-----FUNCS-----

-----STRUCTS-----

Struct => BASE2,          _INSTANCE: (null)

Variable => IM_OKA4Y
Variable => ANTIK2
Variable => OK1

```

Δουλεύει σωστά.

```

%tupou antikeimeno
STRUCT BASE
    PINAKAS = {var,4, 5,"STRINGFOUND"};

STRUCT BASE
    ANTIKEIMENO;
STRUCT BASE2
    PERSON;
ANTIKEIMENO.IM_OKAY[10]= PERSON.IM_OKA4Y[9];

ANTIKEIMENO.OK = 12;

ENDMAIN
}
```

```

The Program Produced 0 Errors.
The program compiled Successfully!

-----VARS-----
Variable => okay          Type => CHAR
Variable => var99          Type => CHAR

Variable => var5          Type => CHAR
Variable => var4          Type => INTEGER

Variable => var3          Type => INTEGER
Variable => var2          Type => INTEGER

Variable => var1          Type => INTEGER
Variable => sum           Type => INTEGER

-----FUNCS-----
FUNCTION => ADD
Arguments=> b
Arguments=> a

-----STRUCTS-----

Struct => BASE2,          _INSTANCE: PERSON

Variable => IM_OKA4Y
Variable => ANTIK2
Variable => OK1

Struct => BASE, _INSTANCE: ANTIKEIMENO

Variable => IM_OKAY
Variable => ANTIK
Variable => OK
Variable => var22
Variable => var22e

```

Και σε αυτή τη περίπτωση το πρόγραμμα τρέχει σωστά.

Παράδειγμα 4^ο: Έλεγχος σωστής λειτουργίας των comment.

```
sample.txt
1 PROGRAM ANAFORA{
2
3     /* eimai sxolio agnohse me*/
4     TYPEDEF STRUCT BASE2
5     VARS INTEGER OK1;
6     VARS CHAR ANTIK2;
7     VARS CHAR IM_OKA4Y[10];
8     BASE2 ENDSTRUCT
9
10    /* agnohse me
11    ksana
12    VARS INTEGER 100;
13    PRINT("IN A COMMENT");
14    PRINT(1
15    TYPEDEF VARS
16    */
17
18    %STRUCTS BELOW
19    TYPEDEF STRUCT BASE2
20    VARS INTEGER OK1;
21    VARS CHAR ANTIK2;
22    VARS CHAR IM_OKA4Y[10];
23    BASE2 ENDSTRUCT
24
25    STARTMAIN
26
27    VARS INTEGER var1;
28    VARS INTEGER var2;
29
30    %while reserved follows
31    WHILE ( A != var3)
32        VARS INTEGER var898;
33        BREAK ;
34    ENDWHILE
35
36
37    ENDMAIN
38 }
```

```
The Program Produced 0 Errors.
The program compiled Successfully!
-----VARS-----
Variable => var898      Type => INTEGER
Variable => var2        Type => INTEGER
Variable => var1        Type => INTEGER
-----
-----FUNCS-----
-----STRUCTS-----
Struct => BASE2,      _INSTANCE: (null)
Variable => IM_OKA4Y
Variable => ANTIK2
Variable => OK1
Struct => BASE2,      _INSTANCE: (null)
Variable => IM_OKA4Y
Variable => ANTIK2
Variable => OK1
-----
```

Παρατηρούμε πως τα σχόλια δουλεύουν σωστά, είτε βρίσκονται εκτός είτε εντός της main, είτε περιέχουν δεσμευμένες λέξεις είτε όχι. Να σημειωθεί πως το (%) αντιστοιχεί σε σχόλια μιας γραμμής ενώ τα (/*σχόλια*/) πολλαπλών γραμμών.

Παράδειγμα 5^ο: Έλεγχος σωστής λειτουργίας της STARTMAIN.

```
38
39 STARTMAIN
40
41
42 VARS INTEGER var1;
43 VARS INTEGER var2;
44 VARS INTEGER var3;
45 VARS INTEGER var4;
46 VARS CHAR var5;
47 VARS CHAR var99;
48 var99=var5+5*(var2+var1);
49 var3=var1;
50 var2=3;
51
52 var1 = ADD(var2,var3);
53 VARS CHAR okay;
54
55 okay = "ok";
56
57
58 ENDMAIN
59 }
```

```
The Program Produced 0 Errors.
The program compiled Successfully!
-----VARS-----
Variable => okay          Type => CHAR
Variable => var99          Type => CHAR
Variable => var5           Type => CHAR
Variable => var4           Type => INTEGER
Variable => var3           Type => INTEGER
Variable => var2           Type => INTEGER
Variable => var1           Type => INTEGER
Variable => sum            Type => INTEGER
-----
-----FUNCS-----
FUNCTION => ADD
Arguments=> b
Arguments=> a
-----
-----STRUCTS-----
Struct => BASE2,          _INSTANCE: (null)
Variable => IM_OKA4Y
Variable => ANTIK2
Variable => OK1
Struct => BASE, _INSTANCE: (null)
Variable => IM_OKAY
Variable => ANTIK
Variable => OK
Variable => var22
Variable => var22e
```

Στις εντολές προγράμματος οι εντολές ανάθεσης δουλεύουν σωστά με τους τρόπους που ζητήθηκαν.

```
38
39 STARTMAIN
40
41
42 VARS INTEGER var1;
43 VARS INTEGER var2;
44 VARS INTEGER var3;
45 VARS INTEGER var4;
46 VARS CHAR var5;
47 VARS CHAR var99;
48 var99=var5+5*(var2+var1);
49 var3=var1;
50 var2=3;
51
52 var1 = ADD(var2,var3);
53 VARS CHAR okay;
54
55 okay = "ok";
56
57 %while reserved follows
58
59 WHILE ( A != var3)
60
61 VARS INTEGER var898;
62
63 BREAK ;
64
65 ENDWHILE
66 ENDMAIN
67 }
```

```
The Program Produced 0 Errors.
The program compiled Successfully!
-----VARS-----
variable => var898        Type => INTEGER
variable => okay          Type => CHAR
variable => var99         Type => CHAR
variable => var5          Type => CHAR
variable => var4          Type => INTEGER
variable => var3          Type => INTEGER
variable => var2          Type => INTEGER
variable => var1          Type => INTEGER
variable => sum           Type => INTEGER
-----
-----FUNCS-----
FUNCTION => ADD
Arguments=> b
Arguments=> a
-----
-----STRUCTS-----
Struct => BASE2,          _INSTANCE: (null)
variable => IM_OKAY
variable => ANTIK2
variable => OK1
Struct => BASE, _INSTANCE: (null)
variable => IM_OKAY
variable => ANTIK
variable => OK
variable => var22
```

Στις εντολές βρόχου, στην πρώτη κατηγορία που ανήκει η WHILE, τρέχει όπως ζητήθηκε.

```

56
57     %while reserved follows
58
59     WHILE ( A != var3)
60
61         VARS INTEGER var898;
62
63         BREAK ;
64
65 ENDMAIN
66 }

```

At line: 65error0: Unrecognized reserved word or character sequence

Ενώ τώρα που λείπει η δεσμευμένη λέξη ENDWHILE, εμφανίζεται error.

```

56
57
58     FOR alfa :=3 TO 180 STEP 10
59         PRINT("The var is", var3);
60         %DOES IT WORK;??
61     ENDFOR
62
63 ENDMAIN
64 }

```

The Program Produced 0 Errors.
The program compiled Successfully!

-----VARS-----

Variable => okay Type => CHAR
Variable => var99 Type => CHAR

Στις εντολές βρόχου, στην δεύτερη κατηγορία που ανήκει η FOR, τρέχει όπως ζητήθηκε.

```

57
58  ✓   FOR alfa :=3 TO 180 STEP 10
59       PRINT("The var is", var3);
60       %DOES IT WORK;??
61
62
63 ENDMAIN
64 }

```

At line: 63error0: Unrecognized reserved word or character sequence

Ενώ τώρα που λείπει η δεσμευμένη λέξη ENDFOR, εμφανίζεται error.

```
56
57 IF (a < 0) THEN
58
59 PRINT("The var is", okay);
60
61 ENDIF
62
63 IF (a < 0) THEN
64
65 BREAK;
66
67 PRINT("The var is", var1);
68
69 ELSEIF (a >0) THEN PRINT("The VAR that found is", var2);
70
71 ELSE PRINT("Else has", var3);
72
73 ENDIF
74
75 ENDMAIN
76
```

```
The Program Produced 0 Errors.
The program compiled Successfully!
-----VARS-----
Variable => okay           Type => CHAR
Variable => var99          Type => CHAR
Variable => var5           Type => CHAR
Variable => var4           Type => INTEGER
```

Στις εντολές ελέγχου, στην πρώτη κατηγορία που ανήκει η IF, τρέχει όπως ζητήθηκε.

```
56
57 IF (a < 0) THEN
58
59 PRINT("The var is", okay);
60
61 ENDIF
62
63 IF (a < 0) THEN
64
65 BREAK;
66
67 PRINT("The var is", var1);
68
69 ELSEIF (a >0) THEN PRINT("The VAR that found is", var2);
70
71 ELSE PRINT("Else has", var3);
72
73 ENDIF
74
75 ENDMAIN
76
```

```
56
57 IF (a < 0) THEN
58
59 PRINT("The var is", okay);
60
61 ENDIF
62
63 IF (a < 0) THEN
64
65 BREAK;
66
67 PRINT("The var is", var1);
68
69 ELSEIF (a >0) THEN PRINT("The VAR that found is", var2);
70
71 ELSEIF PRINT("Else has", var3);
72
73 ENDIF
74
75 ENDMAIN
76
```

```
56
57 IF (a < 0) THEN
58
59 PRINT("The var is", okay);
60
61 ENDIF
62
63 IF (a < 0) THEN
64
65 BREAK;
66
67 PRINT("The var is", var1);
68
69 ELSE (a >0) THEN PRINT("The VAR that found is", var2);
70
71 ELSE PRINT("Else has", var3);
72
73 ENDIF
74
75 ENDMAIN
76
```

```
At line: 69error0: Unrecognized reserved word or
character sequence
```


Στις τρεις παραπάνω περιπτώσεις, εμφανίζεται error, αφού στην πρώτη λείπει η ENDIF, στην δεύτερη είναι λάθος η δομή αφού στο τέλος αντί για ELSE έχουμε ELSEIF, ενώ στη τελευταία έχουμε ELSE αντί για ELSEIF.

```
93
94     var898 = ADD(var3, var6);
95
96     SWITCH (var3)
97
98     CASE : OKAY
99         BREAK;
100    CASE : 1
101        PRINT("The var is", var3);
102        VARS CHAR A[4];
103    CASE : "foundString"
104        PRINT("The Var is", var3);
105
106    CASE : "ONE STRING"
107        PRINT("The var is", var3);
108        VARS CHAR A[4];
109
110    DEFAULT:
111        PRINT("\", var3);
112
113    VARS CHAR ARR[100];
114
115    ENDSWITCH
116
117    ENDMAIN
118 }
```

The Program Produced 0 Errors.
The program compiled Successfully!

-----VARS-----

| | |
|--------------------|-----------------|
| Variable => ARR | Type => CHAR |
| Variable => A | Type => CHAR |
| Variable => A | Type => CHAR |
| Variable => var898 | Type => INTEGER |
| Variable => okay | Type => CHAR |
| Variable => var5 | Type => CHAR |
| Variable => var4 | Type => INTEGER |
| Variable => var3 | Type => INTEGER |
| Variable => var2 | Type => INTEGER |
| Variable => var1 | Type => INTEGER |
| Variable => sum | Type => INTEGER |

-----FUNCS-----

FUNCTION => ADD
Arguments=> b
Arguments=> a

-----STRUCTS-----

Struct => BASE2, _INSTANCE: PERSON

Variable => IM_OKA4Y
Variable => ANTIK2
Variable => OK1

Struct => BASE, _INSTANCE: ANTIKEIMENO

Variable => IM_OKAY

Στις εντολές ελέγχου, στην δεύτερη κατηγορία που ανήκει η SWITCH, τρέχει όπως ζητήθηκε.

```
95
96     SWITCH (var3)
97
98     CASE : OKAY
99         BREAK;
100    CASE : 1
101        PRINT("The var is", var3);
102        VARS CHAR A[4];
103    CASE : "foundString"
104        PRINT("The Var is", var3);
105
106    CASE : "ONE STRING"
107        PRINT("The var is", var3);
108        VARS CHAR A[4];
109
110    DEFAULT:
111        PRINT("\", var3);
112
113    VARS CHAR ARR[100];
114
115    ENDMAIN
116 }
```

At line: 115error0: Unrecognized reserved word or character sequence

Ενώ εδώ που λείπει η ENDSWITCH, έχουμε error.

Οι εντολές PRINT και BREAK υπήρχαν μέσα στα παραδείγματα και αποδεικνύεται παραπάνω η σωστή λειτουργία τους.

```
At line: 1error0: Unrecognized reserved word or character sequence
```

Όταν οι δεσμευμένες λέξεις δεν δηλώνονται σωστά παράγεται το αντίστοιχο μήνυμα της πάνω εικόνας. (πχ. λάθος δήλωση PROGRAM, FUNCTION, LOOPS, VARS κτλπ.).

```
At Line: 87 error9: Array out of range.
```

```
At line: 89error5: Invalid type assignment. You have INTEGER <= CHAR
```

```
The Program Produced 2 Errors.
```

Κάποια επιπλέον λάθη τα οποία αναφέρονται στις παραδοχές.

Αποθήκευση μεταβλητών

Για την ορθή λειτουργία της συντακτικής ανάλυσης, αλλά και την εμφάνιση του symbol table έπρεπε οι μεταβλητές να αποθηκευτούν. Υπάρχουν διάφορες δομές δεδομένων, που μπορούν να εξυπηρετήσουν αυτό τον σκοπό όπως Hash Tables, BSTs, linkedList etc. Στην δεδομένη περίπτωση χρησιμοποιήθηκαν linked lists. Για κάθε διαφορετικό είδος δεδομένου χρησιμοποιήθηκε και ένα διαφορετικό linked list (εικόνα 1).

Γιατί linked Lists ;

Τα linked Lists είναι εύκολα στην υλοποίηση, αλλά και ικανοποιητικά αξιόπιστα στην αποθήκευση των συγκεκριμένων δεδομένων. Επίσης εξυπηρετεί σε μεγάλο βαθμό το searching, διότι το πρώτο στοιχείο που εισήχθη στην λίστα είναι στην κορυφή, οπότε διαισθητικά έχουμε μερική γνώση της θέσης του κάθε δεδομένου.

Γιατί 4 structs ;

Κατασκευάστηκαν 4 structs για να μην υπάρξει σύγχυση κατά την εισαγωγή και επεξεργασία των στοιχείων, αλλά να είναι και κατανοητά προς τον αναγνώστη του κώδικα.

(εικόνα 1)

4 Διαφορετικοί τύποι struct. Το struct List για την αποθήκευση μεταβλητών, struct Struct για την αποθήκευση Δομών, struct Functions για την αποθήκευση Συναρτήσεων και struct Arglist για την αποθήκευση ορισμάτων

```
struct List{
    char * type;
    char * data;
    char *val;
    int idata;
    int N;
    struct List * next;
};

struct Structs{
    char *name;
    char *Instance_name;
    struct List * arguments;
    struct Structs *next;
};

struct Functions{
    char * name;
    struct ArgList *arguments;
    int count;
    struct Functions *next;
};

struct ArgList{
    char * varName;
    int count ;
    struct ArgList * next;
};
```

των συναρτήσεων

Παραδοχές :

Πολλές παραδοχές στον κώδικα μπορούν από τα μηνύματα λάθους του συντακτικού αναλυτή (εικόνα 2).

```

int error(int type, char * word1, char * word2) {
    switch(type){
        case 1:
            printf("\nAt Line: %d \033[0;31merror1: In function \033[0;32m%s \033[0;31m arguments out of scope. \n", lineCounter, word1);
            printf("\033[0;35m-----\033[0;37m\n");
            counter ++;
            break;
        case 2:
            printf("\nAt Line: %d \033[0;31merror2: The Struct \033[0;32m%s \033[0;31m is not declared. \n", lineCounter, word1);
            printf("\033[0;35m-----\033[0;37m\n");
            counter ++;
            break;
        case 3:
            printf("\nAt Line: %d \033[0;31merror3: Struct \033[0;32m%s \033[0;31m has no member: %s. \n", lineCounter, word1, word2);
            printf("\033[0;35m-----\033[0;37m\n");
            counter ++;
            break;
        case 4:
            printf("\033[0;31merror4:At Line: %d The value of assignment \033[0;32m%s \033[0;31mis not declared\033[0;37m\n", lineCounter, word1);
            counter ++;
            break;
        case 5:
            printf("\nAt line: %d\033[0;31merror5: Ivalid type assignment. You have \033[0;32m%s \033[0;31m<= \033[0;32m%s \033[0;37m \n", lineCounter, word1, word2);
            printf("\033[0;35m-----\033[0;37m\n");
            counter ++;
            break;
        case 6:
            printf("\nAt line: %d\033[0;31merror6: The function \033[0;32m%s \033[0;31mdoes not exist\033[0;37m \n", lineCounter, word1);
            printf("\033[0;35m-----\033[0;37m\n");
            counter ++;
            break;
        case 7:
            printf("\nAt line: %d\033[0;31merror7: You can not call a function in global field. \n", lineCounter);
            printf("\033[0;35m-----\033[0;37m\n");
            counter ++;
            break;
        case 8:
            printf("\nAt Line: %d \033[0;31merror8: The function \033[0;32m%s \033[0;31m is undeclared. \n", lineCounter, word1);
            printf("\033[0;35m-----\033[0;37m\n");
            counter ++;
            break;
        case 9:
            printf("\nAt Line: %d \033[0;31merror9: Array out of scope. \n", lineCounter);
            printf("\033[0;35m-----\033[0;37m\n");
            counter ++;
            break;
        case 10:
            printf("\nAt Line: %d \033[0;31merror10: The variable \033[0;32m%s \033[0;31m is undeclared. \n", lineCounter, word1);
            printf("\033[0;35m-----\033[0;37m\n");
            counter ++;
            break;
        case 11:
            printf("\nAt Line: %d \033[0;31merror11: Struct names should much \033[0;32m%s \033[0;31m is undeclared. \n", lineCounter);
            printf("\033[0;35m-----\033[0;37m\n");
            counter ++;
            break;
        case 12:
            printf("\n \033[0;31mThe Program Produced\033[0;32m %d \033[0;31m Errors. \033[0;37m \n", counter);
            break;
        case 13:
            printf("\nAt line: %d\033[0;31merror13:Struct _Instance \033[0;32m%s \033[0;31m has no previous declaration.\033[0;37m \n", lineCounter, word1);
            printf("\033[0;35m-----\033[0;37m\n");
            counter ++;
            break;
    }
}

```

(εικόνα 2)

Τα error messages που παράγονται.
Θα αναλυθούν παρακάτω

- **Error message 1:** Παράγεται όταν τα ορίσματα μέσα σε μια συνάρτηση (κατά το κάλεσμα της) ξεπεράσουν ή είναι λιγότερα από το όριο της δήλωσης της.(εικόνα 3)

(εικόνα 3)

```

At Line: 51 error1: In function ADD arguments out of scope.
-----
The Program Produced 1 Errors.

```

- **Error message 2:** Το μήνυμα λάθους 2 προκύπτει όταν καλείς ένα τυχαίο όνομα struct το οποίο δεν υπάρχει, δηλαδή δεν έχει δηλωθεί εκτός της main σαν struct. (εικόνα 4)

(εικόνα 4)

```

At Line: 102 error2: The Struct NotDeclaredClass is not declared.
-----

```

- **Error message 3:** Εμφανίζεται όταν καλείται ένα μέλος ενός struct το οποίο δεν υπάρχει.(εικόνα 5)

```

At Line: 99 error3: Struct ANTIKEIMENO has no member: notExistingMember.
-----

```

(εικόνα 5)

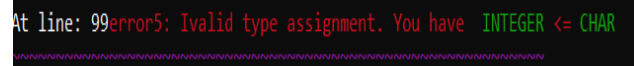
- **Error message 4:** Όταν θέτουμε σε μια ορισμένη μεταβλητή μια τιμή, η οποία δεν υπάρχει, παράγεται το αντίστοιχο μήνυμα λάθους.(εικόνα 6) .



```
error4:At Line: 49 The value of assignment at var1 is not declared
```

(εικόνα 6)

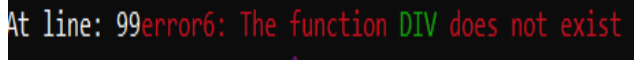
- **Error message 5:** Κατά την εισαγωγή μια μεταβλητής σε μια άλλη μεταβλητή, εάν ο τύπος τους δεν είναι συμβατός παράγεται μήνυμα λάθους.(εικόνα 7)



```
At line: 99error5: Invalid type assignment. You have INTEGER <= CHAR
```

(εικόνα 7)

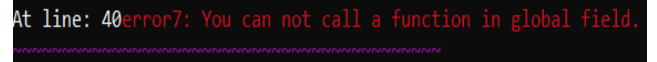
- **Error message 6:** Παράγεται σε περίπτωση που καλέσουμε μια λέξη σε μορφή συνάρτησης (π.χ. DIV(a,b)).(εικόνα 8)



```
At line: 99error6: The function DIV does not exist
```

(εικόνα 8)

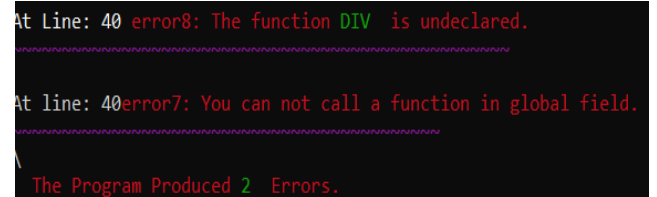
- **Error message 7:** Ο Συντακτικός αναλυτής δεν επιτρέπει το κάλεσμα συνάρτησης σε global field. (εικόνα 9)



```
At line: 40error7: You can not call a function in global field.
```

(εικόνα 9)

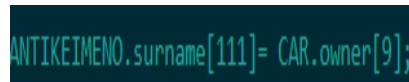
- **Error message 8:** Παράγεται όταν καλείται και δεν υπάρχει μια συνάρτηση. Συνοδεύεται από το error message 7. (εικόνα 10)



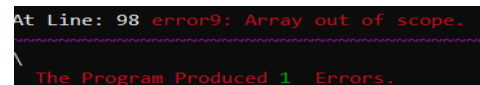
```
At Line: 40 error8: The function DIV is undeclared.
At line: 40error7: You can not call a function in global field.
The Program Produced 2 Errors.
```

(εικόνα 10)

- **Error message 9:** Κατά το κάλεσμα ενός πίνακα, αν ξεπεράσουμε τα όρια του, εμφανίζεται το αντίστοιχο μήνυμα λάθους. (εικόνα 11)



```
ANTIKEIMENO.surname[111]= CAR.owner[9];
```

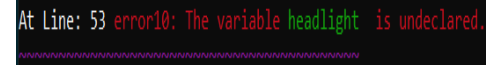


```
At Line: 98 error9: Array out of scope.
The Program Produced 1 Errors.
```

(εικόνα 11)

- **Error message 10:** Όταν μια μεταβλητή δεν έχει δηλωθεί και καλείται, παράγεται το αντίστοιχο μήνυμα. (εικόνα 12)

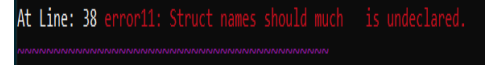
(εικόνα 12)



```
At Line: 53 error10: The variable headlight is undeclared.
```

- **Error message 11:** Κατά την χρήση του typedef (στην περίπτωση δήλωσης μιας δομής), όταν τα ονόματα δεν ταιριάζουν, εκτυπώνεται το error11. (εικόνα 13)

(εικόνα 13)



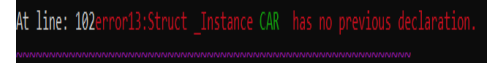
```
At Line: 38 error11: Struct names should much is undeclared.
```

- **Error message 13:** Στην περίπτωση αυτή, κατά την δήλωση αντικειμένου μια δομής όταν η δομή δεν υπάρχει, κείται μήνυμα λάθους. (εικόνα 14)

(εικόνα 14)

Τα error 13 & 11 συνδυάζονται όταν δεν έχει γίνει σωστή δήλωση σε δομή με typedef και όταν κείται δήλωση αντικειμένου με την λάθος δηλωμένη δομή.

Για την περίπτωση 12 δεν έγινε αναφορά. Η περίπτωση 12 ουσιαστικά μετράει τα πόσα λάθη έγιναν κατά την συντακτική ανάλυση και βγάζει τον αριθμό τους. Σε περίπτωση που δεν υπάρχουν λάθη βγάζει μήνυμα 0.



```
At Line: 102error13:Struct _Instance CAR has no previous declaration.
```

Αν ο κώδικας είναι σωστός παράγεται κάτι αντίστοιχο της (εικόνας 15).

```

The Program Produced 0 Errors.
The program compiled Successfully!
-----VARS-----
Variable => ARR          Type => CHAR
Variable => A             Type => CHAR
Variable => A             Type => CHAR
Variable => var898        Type => INTEGER
Variable => okay          Type => CHAR
Variable => var5           Type => CHAR
Variable => var11         Type => INTEGER
Variable => var4           Type => INTEGER
Variable => var3           Type => INTEGER
Variable => var2           Type => INTEGER
Variable => var1           Type => INTEGER
Variable => sum            Type => INTEGER
-----
-----FUNCS-----
FUNCTION => ADD
Arguments=> b
Arguments=> a
-----
-----STRUCTS-----
Struct => Car_Owner,      _INSTANCE: CAR
Variable => owner
Variable => licenceE
Variable => licencep
Struct => Person,         _INSTANCE: ANTIKEIMENO
Variable => surname
Variable => FirstNam
Variable => height
Variable => weight
Variable => licencep

```

(εικόνα 15)

Όταν ο κώδικας δεν έχει λάθη παράγεται το symbol table που περιέχει όλα τα αποθηκευμένα δεδομένα.

(εικόνα 16)

```

At line: 1error0: Unrecognized reserved word or character sequence

```

Όταν οι δεσμευμένες λέξεις δεν δηλώνονται σωστά παράγεται το αντίστοιχο μήνυμα της αριστερής εικόνας.
πχ.(λάθος δήλωση PROGRAM, FUNCTION, LOOPS, VARS κτλπ.)