Π Α Ν Ε Π Ι Σ Τ Η Μ Ι Ο Π Α Τ Ρ Ω Ν

Πολυτεχνικη Σχολη

Τμημα Μηχανικων Ηλεκτρονικων Υπολογιστων & Πληροφορικης

ΕΡΓΑΣΙΑ FLEX & BISON ΣΤΟ ΜΑΘΗΜΑ ΑΡΧΕΣ ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΚΑΙ ΜΕΤΑΦΡΑΣΤΩΝ

Πατρα 2020

Μέλη ομάδας:

- Αβραμόπουλος Γεώργιος - 1070772 – Έτος 3ο – up1070772@upnet.gr
- Κοντογιώργης Νικόλαος – 1070922 – Έτος 3ο – up1070922@upnet.gr
- Κωνσταντοπούλου Νεφέλη – 1059628 – Έτος 4ο – up1059628@upnet.gr
- Μαραζιώτης Ορέστης – 1064028 – Έτος 4ο – up1064028@upnet.com

Backus-Naur Form notation (BNF):

<A_PARENTHESI> ::= “(”
<D_PARENTHESI> ::= “)”
<KOMMA> ::= “,”
<Q_MARK> :: “;”
<A_BRACKET> ::= “[”
<D_BRACKET> ::= “]”
<ADD> ::= “+”
<SUBTRACT> ::= “-”
<POWER_OF> ::= “^”
<MULTIPLY> ::= “*”
<DIVIDE> ::= “/”
<EQUALS> ::= “=”
<BIGGER_THAN> ::= “<”
<SMALLER_THAN> ::= “>”
<LOG_EQUALS> ::= “==”
<NOT_EQUAL> ::= “!=”
<COLON> ::= “:”
<ARIST> ::= “’”

<WHILE> ::= “while”
<ENDWHILE> ::= “end_while”
<AND> ::= “and”
<OR> ::= “or”
<FOR> ::= “for”
<COUNTER> ::= “counter”
<TO> ::= “to”
<STEP> ::= “step”
<ENDFOR> ::= “end_for”
<IF> ::= “if”
<THEN> ::= “then”
<ELSE> ::= “else”
<ELSEIF> ::= “else_if”
<ENDIF> ::= “end_if”
<SWITCH> ::= “switch”
<CASE> ::= “case”
<DEFAULT> ::= “default”
<END_SWITCH> ::= “end_switch”
<PRINT> ::= “print”
<BREAK> ::= “break”
<PROGRAM> ::= “program”
<RETURN> ::= “return”

<THETIKOS_AKER> ::= “thetikos_aker”
<FLOAT> ::= “myfloat”
<STARTMAIN> ::= “start_main”
<ENDMAIN> ::= “end_main”
<FUNCTION> ::= “function”
<VARS> ::= “vars”
<END_FUNCTION> ::= “end_function”
<STRUCT> ::= “struct”
<ENDSTRUCT> ::= “endstruct”
<TYPEDEF> ::= “typedef”
<CHAR> ::= “mychar”
<INTEGER> ::= “myinteger”

<LETTER> ::= a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z
<DIGIT> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<STRING> ::= <LETTER> |
<LETTER><STRING> |
<DIGIT><STRING>

<main> ::= <program><list_structs><list_function><function><startmain>
| <program><list_structs><startmain>
| <program><startmain>
| <program>

<program> ::= <PROGRAM><STRING>

<list_structs> ::= <list_structs><structs>
| <structs>

<structs> ::= <STRUCT><STRING><vars><ENDSTRUCT>
| <STRUCT><STRING><vars><STRING><ENDSTRUCT>

<vars> ::= <VARS><vars_list>

<vars_list> ::= <CHAR><string_list><Q_MARK><vars_list>
| <INTEGER><string_list><Q_MARK><vars_list>
| <CHAR><string_list><Q_MARK>
| <INTEGER><string_list><Q_MARK>

<string_list> ::=<STRING><A_BRACKET><THETIKOS_AKER><D_BRACKET><KOMMA>
<string_list>
| <STRING><KOMMA><string_list>
| <STRING><A_BRACKET><THETIKOS_AKER><D_BRACKET>
| <STRING>

<list_function> ::= <list_function><function>
| <function>

<function> ::= <FUNCTION><STRING><A_PARENTHESI><string_list><D_PARENTHESI><vars>
<entoles><return><END_FUNCTION>
| <FUNCTION><STRING><A_PARENTHESI><D_PARENTHESI><vars><entoles><return>
<END_FUNCTION>
| <FUNCTION><STRING><A_PARENTHESI><string_list><D_PARENTHESI><entoles>
<return><END_FUNCTION>
| <FUNCTION><STRING><A_PARENTHESI><D_PARENTHESI><entoles><return>
<END_FUNCTION>

<return> ::= <RETURN><THETIKOS_AKER><Q_MARK>
| <RETURN><STRING><Q_MARK>

<list_entoles> ::= <list_entoles><entoles>
| <entoles>

<entoles> ::= <anathesi>
| <while>
| <for>
| <if>
| <switch>
| <print>
| <break>

<print> ::= <PRINT><A_PARENTHESI><ARIST><STRING><ARIST><D_PARENTHESI><Q_MARK>
| <PRINT><A_PARENTHESI><ARIST><STRING><ARIST><KOMMA><string_list>
<D_PARENTHESI><Q_MARK>

<break> ::= <BREAK><Q_MARK>

<anathesi> ::= <STRING><EQUALS><prajeis><Q_MARK>

<prajeis> ::= <prajeis><ADD><prajeis>
| <prajeis><SUBTRACT><prajeis>
| <prajeis><MULTIPLY><prajeis>
| <prajeis><DIVIDE><prajeis>
| <prajeis><POWER_OF><prajeis>
| <A_PARENTHESI><prajeis><D_PARENTHESI>
| <STRING><ADD><prajeis>
| <STRING><A_PARENTHESI><string_list><D_PARENTHESI><ADD><prajeis>
| <THETIKOS_AKER><ADD><prajeis>
| <STRING><SUBTRACT><prajeis>
| <THETIKOS_AKER><SUBTRACT><prajeis>
| <STRING><MULTIPLY><prajeis>
| <STRING><A_PARENTHESI><string_list><D_PARENTHESI><MULTIPLY><prajeis>
| <THETIKOS_AKER><MULTIPLY><prajeis>
| <STRING><DIVIDE><prajeis>
| <STRING><A_PARENTHESI><string_list><D_PARENTHESI><DIVIDE><prajeis>
| <THETIKOS_AKER><DIVIDE><prajeis>
| <STRING><POWER_OF><prajeis>
| <STRING><A_PARENTHESI><string_list><D_PARENTHESI><POWER_OF><prajeis>
| <THETIKOS_AKER><POWER_OF><prajeis>
| <STRING><A_PARENTHESI><string_list><D_PARENTHESI>

| <STRING><A_PARENTHESI><D_PARENTHESI>
| <THETIKOS_AKER>
<while> ::= <WHILE><condition><list_entoles><ENDWHILE>

<for> ::= <FOR><STRING><COLON><EQUALS><THETIKOS_AKER><TO><THETIKOS_AKER>
<STEP><THETIKOS_AKER><list_entoles><ENDFOR>

<list_elseif> ::= <list_elseif><elseif>
| <elseif>

<elseif> ::= <ELSEIF><condition><list_entoles>

<if> ::= <IF><condition><THEN><list_entoles><ENDIF>
| <IF><condition><THEN><list_entoles><ELSE><list_entoles><ENDIF>
| <IF><condition><THEN><list_entoles><list_elseif><ENDIF>
| <IF><condition><THEN><list_entoles><list_elseif><ELSE><list_entoles><ENDIF>

<list_case> ::= <list_case><case>
| <case>

<case> ::= <CASE><prajeis><COLON><list_entoles>

<switch> ::= <SWITCH><prajeis><list_case><DEFAULT><COLON><list_entoles><ENDSWITCH>
| <SWITCH><prajeis><list_case><ENDSWITCH>

<and_or> ::= <AND>
| <AND><and_or>
| <OR>
| <OR><and_or>

<condition> ::= <A_PARENTHESI><STRING><BIGGER_THAN><THETIKOS_AKER>
<D_PARENTHESI>
| <A_PARENTHESI><STRING><BIGGER_THAN><THETIKOS_AKER>
<D_PARENTHESI><and_or><condition>
| <A_PARENTHESI><STRING><BIGGER_THAN><STRING><D_PARENTHESI>
| <A_PARENTHESI><STRING><BIGGER_THAN><STRING><D_PARENTHESI>
<and_or><condition>
| <A_PARENTHESI><STRING><SMALLER_THAN><THETIKOS_AKER>
<D_PARENTHESI>
| <A_PARENTHESI><STRING><SMALLER_THAN><THETIKOS_AKER>
<D_PARENTHESI><and_or><condition>
| <A_PARENTHESI><STRING><SMALLER_THAN><STRING><D_PARENTHESI>
| <A_PARENTHESI><STRING><SMALLER_THAN><STRING><D_PARENTHESI>
<and_or><condition>
| <A_PARENTHESI><STRING><LOG_EQUALS><THETIKOS_AKER><D_PARENTHESI>
| <A_PARENTHESI><STRING><LOG_EQUALS><THETIKOS_AKER><D_PARENTHESI>
<and_or><condition>
| <A_PARENTHESI><STRING><LOG_EQUALS><STRING><D_PARENTHESI>
| <A_PARENTHESI><STRING><LOG_EQUALS><STRING><D_PARENTHESI>
<and_or><condition>
| <A_PARENTHESI><STRING><NOT_EQUAL><THETIKOS_AKER><D_PARENTHESI>
| <A_PARENTHESI><STRING><NOT_EQUAL><THETIKOS_AKER><D_PARENTHESI>
<and_or><condition>
| <A_PARENTHESI><STRING><NOT_EQUAL><STRING><D_PARENTHESI>

```
        | <A_PARENTHESI><STRING><NOT_EQUAL><STRING><D_PARENTHESI>
        <and_or><condition>
        <A_PARENTHESI><condition><D_PARENTHESI>

<startmain> ::= <STARTMAIN><vars><list_entoles><ENDMAIN>
        | <STARTMAIN><list_entoles><ENDMAIN>
```

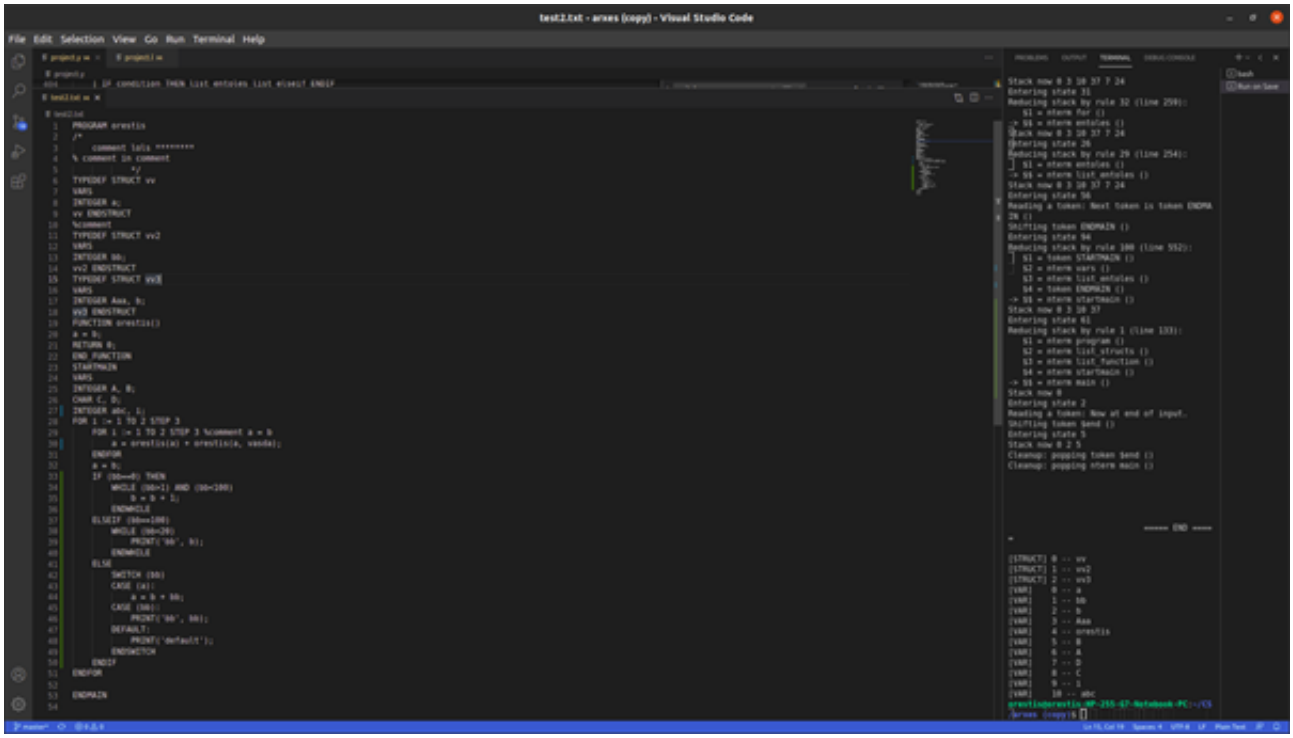
Περιεχόμενα αρχείου εισόδου .txt (ορθής λειτουργίας):

```
PROGRAM orestis
TYPEDEF STRUCT vv
VARS
INTEGER a;
vv ENDSTRUCT
%comment
TYPEDEF STRUCT vv2
VARS
INTEGER bb;
vv2 ENDSTRUCT
TYPEDEF STRUCT vv3
VARS
INTEGER Aaa, b, pinakas[3];
vv3 ENDSTRUCT
FUNCTION orestis()
RETURN 0;
END_FUNCTION
/*
  comment lols ********
% comment in comment
        */
STARTMAIN
VARS
INTEGER A, B;
CHAR C, D;
INTEGER abc, i;
FOR i := 1 TO 2 STEP 3
  FOR i := 1 TO 2 STEP 3 %comment a = b
    a = orestis(a) + orestis(a, vasda);
  ENDFOR
  a = b;
  IF (bb==0) THEN
    WHILE (bb>1) AND (bb<100)
      b = b + 1;
    ENDWHILE
  ELSEIF (orestis(bb)==100)
    WHILE (bb<20)
      PRINT('bb', b);
    ENDWHILE
  ELSE
    SWITCH (bb)
    CASE (a):
      a = b + bb;
    CASE (bb):
```

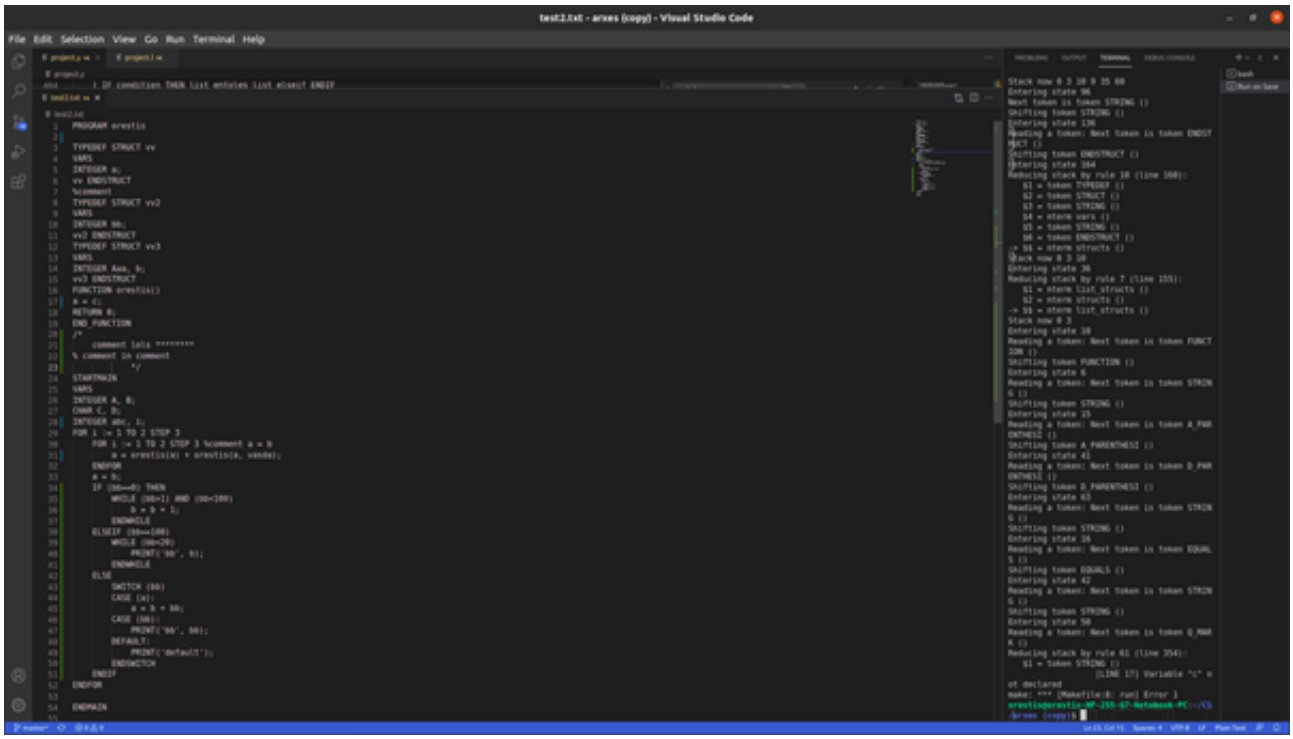
```
            PRINT('bb', bb);
        DEFAULT:
            PRINT('default');
        ENDSWITCH
    ENDIF
ENDFOR

ENDMAIN
```

Screenshots λειτουργίας (στο αριστερό μέρος βλέπουμε το txt αρχείο με το τρέχον παράδειγμα ενώ στο δεξί μέρος βλέπουμε το terminal με το output του bison):

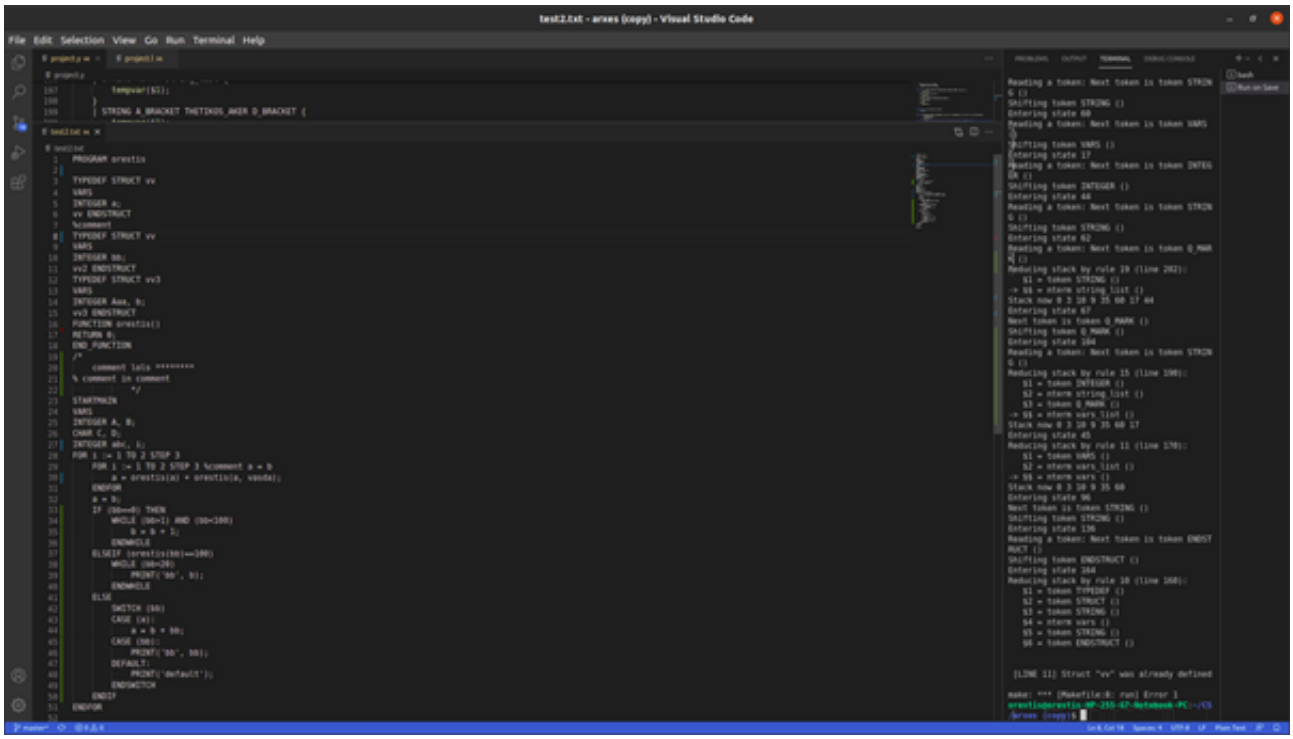
- Γενική περίπτωση ορθής δήλωσης μεταβλητών, structs και functions.



- Παράδειγμα όπου η μεταβλητή c δεν έχει δηλωθεί και σε ποιό line υπάρχει.



- Παράδειγμα όπου η struct vv υπάρχει ήδη και σε ποιό line.



Multiline comments:

Τα σχόλια πολλαπλών γραμμών υποστηρίζονται με το παρακάτω κομμάτι κώδικα στο flex αρχείο:

```
55  %%
56  <INITIAL>{
57    "/*"              BEGIN(IN_COMMENT);
58  }
59  <IN_COMMENT>{
60    "*/"      BEGIN(INITIAL);
61    [^*\n]+   {};
62    "*"       {};
63  }
```

Παραδοχές:

- Οι τύποι μεταβλητών είναι οι integer και char.
- Οι έλεγχοι των δηλώσεων των functions είναι ίδια με των variables καθώς και στην περίπτωση όπου υπάρχει κάποια ίδια ή δεν έχει δηλωθεί κάποια.

Σχόλια:

- Όνομα αρχείου flex, project.l
- Όνομα αρχείου bison, project.y
- Όνομα αρχείου input για έλεγχο, test2.txt
- Compile & Execute: cd PROJECTDIR && make clean && make && make run