



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

PROJECT ΑΡΧΕΣ ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ
ΚΑΙ ΜΕΤΑΦΡΑΣΤΩΝ 2020-2021

ΣΤΟΙΧΕΙΑ ΜΕΛΩΝ ΟΜΑΔΩΝ

ΑΘΑΝΑΣΟΠΟΥΛΟΣ ΠΑΝΑΓΙΩΤΗΣ 1058112 5ο ΕΤΟΣ pathanas@ceid.upatras.gr
ΣΙΝΑΣ ΡΙΓΚΕΡΣ 1054345 5ο ΕΤΟΣ sinas@ceid.upatras.gr
ΚΟΣΜΙΔΗΣ ΚΩΣΤΑΝΤΙΝΟΣ 1054431 5ο ΕΤΟΣ kosmidis@ceid.upatras.gr

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΓΡΑΜΜΑΤΙΚΗΣ ΤΗΣ ΓΛΩΣΣΑ ΣΕ BNF	2
ΑΡΧΕΙΟ ΠΕΡΙΓΡΑΦΗΣ ΤΗΣ ΓΛΩΣΣΑΣ ΩΣ ΕΙΣΟΔΟΣ ΣΤΟ FLEX	3
ΑΡΧΕΙΟ ΠΕΡΙΓΡΑΦΗΣ ΤΗΣ ΓΛΩΣΣΑΣ ΩΣ ΕΙΣΟΔΟΣ ΣΤΟ BISON	5
SCREENSHOTS	9
ΣΧΟΛΙΑ - ΠΑΡΑΔΟΧΕΣ	16

ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΓΡΑΜΜΑΤΙΚΗΣ ΤΗΣ ΓΛΩΣΣΑ ΣΕ BNF

```
<program> :: <declList>
<declList> :: = <declList decl> | <decl>
<decl> :: = varDecl | <funDecl>
<funDecl> :: = <typeSpec> ID (parms) <stmt> | ID (parms) <stmt>

<stmtList>:: = <stmtList> <stmt>
<stmt> :: = <expStmt> | <iterStmt> | <selectStmt> | <switch> |
<print> | <breakStmt> | TYPE_COMMENT
<iterStmt> :: = while <simpleExp> do <stmt> | for ID=iterRange do
<stmt>
<selectStmt> :: =if <simpleExp> then <stmt> | if simpleExp then <stmt>
else <stmt>

<expStmt> :: = <exp> ; | ;
<exp> :: = <mutable> '=' <exp> | <mutable> '+=' <exp> | <mutable> '-='
<exp> | <mutable. '*=' <exp> | <mutable> '/=' <exp> | <mutable> ++|
<mutable> -- | <SimpleExp>

<mutable> :: = ID | ID [ exp ]
<call> :: = ID <args>
<args> :: = <argList>
<argList> :: =<argList> , <exp> | <exp>

<parms> :: =<parmList>
<parmList> :: = <parmList> ; <parmTypeList> | <parmTypeList>
<parmTypeList> :: = <typeSpec> parmIdList
<parmIdList> :: = <parmIdList> , <parmId> | <parmId>
<parmId> :: = ID | ID []

<varDecl> :: =<typeSpec> <varDeclList>
```

```

<typeSpec> :: = int | char

<varDeclList> :: = <varDeclId>,<varDeclInit> | <varDeclInit>
<varDeclInit> :: = <varDeclId> | <varDeclId> : simpleExp
<varDeclId> :: = ID | ID [NUMCONST]

<returnStmt> :: = <return> ; | <return> exp;
<breakStmt> :: = <break> ;

<simpleExp> :: = <simpleExp> or <andExp> | <andExp>
<andExp> :: = <andExp> and <unaryRelExp> | <unaryRelExp>
<unaryRelExp> :: = not <unaryRelExp> | <relExp>

<mulExp> :: = <mulExp> <mulop> <unaryExp> | <unaryExp>
<mulop> :: = * | / | %
<unaryExp> :: = <unaryop> <unaryExp> | <factor>
<unaryop> :: = - | * | ?

<factor> :: = <immutable> | <mutable>
<immutable> :: = (exp) | <call>

```

ΑΡΧΕΙΟ ΠΕΡΙΓΡΑΦΗΣ ΤΗΣ ΓΛΩΣΣΑΣ ΩΣ ΕΙΣΟΔΟΣ ΣΤΟ FLEX

```

%{
#include <stdio.h>
#include <string.h>
#include "y.tab.h"

int lineno = 1;

void showError();

}%

%option noyywrap yylineno

%x comments

```

```

%x MULTILINE_COMMENT

NUMBERS ([0-9])+
NAME ([a-zA-Z0-9_])+
COMMENT "%" .*
SUMOP [-+]
MULOP [/^*]
MARKS ["]

%%

"=" { return (ASSIGNOP); }
"INTEGER" { return (INTEGER); }
"VARS" { return (VARS); }
"PROGRAM" { return (PROGRAM); }
"FUNCTION" { return (FUNCTION); }
"END_FUNCTION" { return (END_FUNCTION); }
"STARTMAIN" { return (STARTMAIN); }
"ENDMAIN" { return (ENDMAIN); }
"PRINT" { return (PRINT); }
"RETURN" { return (RETURN); }
"STRUCT" { return (STRUCT); }
"ENDSTRUCT" { return (ENDSTRUCT); }
"TYPEDEF" { return (TYPEDEF); }
" ( ) " { return (ARGS); }
" : " { return (DELI); }
" . " { return (DOT); }
"IF" { return (IF); }
"ELSEIF" { return (ELSEIF); }
"ELSE" { return (ELSE); }
"ENDIF" { return (ENDIF); }
"THEN" { return (THEN); }
"SWITCH" { return (SWITCH); }
"CASE" { return (CASE); }
"DEFAULT" { return (DEFAULT); }
"ENDSWITCH" { return (ENDSWITCH); }
"<" | ">" | "==" | "!=" { return (TESTOP); }
"FOR" { return (FOR); }
"ENDFOR" { return (ENDFOR); }
"STEP" { return (STEP); }
"TO" { return (TO); }
"WHILE" { return (WHILE); }

```

```

"ENDWHILE"                { return (ENDWHILE); }
"|"                        { return (OR); }
"&&"                      { return (AND); }
"("                        { return (LEFT_PAR); }
")"                        { return (RIGHT_PAR); }
"{"                        { return (LEFT_C_BRA); }
"}"                        { return (RIGHT_C_BRA); }
"["                        { return (LEFT_BRA); }
"]"                        { return (RIGHT_BRA); }
";"                        { return (SEM); }
","                        { return (COMMA); }
"CHAR"                     { return (CHAR); }
"BREAK"                    { return (BREAK); }
{NUMBERS}                  { yylval.i = atoi(yytext);
return INT; }
{COMMENT}                  { return (TYPE_COMMENT); }
{NAME}                     { yylval.a = yytext; return
NAME; }
{SUMOP}                    { return (SUMOP); }
{MULOP}                    { return (MULOP); }
{MARKS}                    { return (MARKS); }
"\n"|"\\r\\n"              { return (NEWLINE); }
"\\t"|" "|" " " " "        { return (INDENT); }
.                           { showError(); }
"/*"                       {BEGIN(MULTILINE_COMMENT);}
<MULTILINE_COMMENT>\\n|\\r\\n|\\r {lineno++;}
<MULTILINE_COMMENT>.        {}
<MULTILINE_COMMENT>"*/"     {BEGIN(INITIAL);}

%%

void showError() {
    // printf("Unrecognized token\\n");
}

```

ΑΡΧΕΙΟ ΠΕΡΙΓΡΑΦΗΣ ΤΗΣ ΓΛΩΣΣΑΣ ΩΣ ΕΙΣΟΔΟΣ ΣΤΟ BISON

```

%{
#include <stdio.h>

```

```

int yylex();
int yyerror(char *s);

extern FILE *yyin;
extern FILE **yyout;
extern char* yytext;
extern int yylineno;
extern int lineno;

%}

%token VARS PROGRAM FUNCTION END_FUNCTION STARTMAIN ENDMAIN RETURN ARGS
DELI DOT TEST
%token TESTOP FOR WHILE OR AND LEFT_PAR RIGHT_PAR LEFT_C_BRA
RIGHT_C_BRA LEFT_BRA RIGHT_BRA
%token THEN TEST2 INT NAME TYPE_COMMENT ASSIGNOP SEM COMMA CHAR BREAK
OP NEWLINE INDENT
%token INTEGER IF ELSE ENDIF ENDWHILE CHARACTER STRING_LITERAL SUMOP
MULOP TO ENDFOR STEP ELSEIF
%token SWITCH CASE DEFAULT ENDSWITCH PRINT MARKS STRUCT ENDSTRUCT
TYPEDEF

%type <a> NAME
%type <i> INT

%union {
    char *a;
    int i;
}

%%

program:      PROGRAM NAME newline declList spaces mainDecl;
declList:     structDecl spaces decl | declList spaces decl | decl |
structDecl | /* empty */;
decl:         funDecl;
funDecl:      FUNCTION NAME LEFT_PAR parms RIGHT_PAR NEWLINE stmtList
spaces RETURN complExp SEM NEWLINE END_FUNCTION { printf("Function
definition\n"); };
mainDecl:     STARTMAIN spaces stmtList spaces ENDMAIN { printf("Main
definition\n"); };
structDecl:   structDecl spaces struct | struct;

```

```

struct:          STRUCT NAME NEWLINE varRepeat spaces ENDSTRUCT {
printf("Struct definition\n"); }
                | TYPEDEF STRUCT NAME NEWLINE varRepeat spaces NAME
ENDSTRUCT { printf("Struct definition\n"); }
                ;
stmtList:        stmtList spaces stmt | stmt | /* empty */;
stmt:            varDecl | expStmt | iterStmt | selectStmt | switch |
print | break | TYPE_COMMENT { printf("Comment\n"); };
iterStmt:        WHILE LEFT_PAR NAME condition complExp RIGHT_PAR
NEWLINE stmtList NEWLINE ENDWHILE { printf("While statement\n"); }
                | FOR NAME ASSIGNOP INT TO INT STEP INT NEWLINE stmtList
NEWLINE ENDFOR { printf("For statement\n"); }
                ;
selectStmt:      IF LEFT_PAR NAME condition complExp RIGHT_PAR THEN
NEWLINE stmtList NEWLINE ENDIF { printf("If statement\n"); }
                | IF LEFT_PAR NAME condition complExp RIGHT_PAR THEN
NEWLINE stmtList NEWLINE elseif NEWLINE ELSE NEWLINE stmtList NEWLINE
ENDIF { printf("If statement\n"); }
                | IF LEFT_PAR NAME condition complExp RIGHT_PAR THEN
NEWLINE stmtList NEWLINE ELSE stmtList NEWLINE ENDIF { printf("If
statement\n"); }
                ;
elseif:          elseif NEWLINE ELSEIF NEWLINE stmt
                | ELSEIF NEWLINE stmt
                ;
switch:          SWITCH LEFT_PAR NAME RIGHT_PAR NEWLINE case NEWLINE
stmtList ENDSWITCH { printf("Switch statement\n"); }
                | SWITCH LEFT_PAR NAME RIGHT_PAR NEWLINE case NEWLINE
DEFAULT DELI NEWLINE stmtList NEWLINE ENDSWITCH { printf("Switch
statement\n"); }
                ;
case:            case NEWLINE CASE LEFT_PAR complExp RIGHT_PAR DELI
NEWLINE stmt
                | CASE LEFT_PAR complExp RIGHT_PAR DELI NEWLINE stmt
                ;
print:           PRINT LEFT_PAR MARKS text MARKS RIGHT_PAR SEM {
printf("Print expression\n"); }
                | PRINT LEFT_PAR MARKS text MARKS LEFT_BRA COMMA NAME
RIGHT_BRA RIGHT_PAR SEM { printf("Print expression\n"); }
                ;
varRepeat:       varRepeat varDecl | varDecl | /* empty */;
spaces:          spaces newline | spaces space | newline | space;
newline:         newline NEWLINE | NEWLINE;

```

```

space:      space spac | spac;
spac:      /* empty */;
text:      text NAME | NAME;
break:     BREAK SEM { printf("Break statement\n"); };
condition: TESTOP | AND | OR;
expStmt:   exp;
exp:       resExp | mutable ASSIGNOP exp;
resExp:    call | complExp;
complExp:  INT | NAME | SUMOP | MULOP | LEFT_PAR | RIGHT_PAR |
complExp SEM;
mutable:   NAME | NAME LEFT_PAR INT RIGHT_PAR;
call:      NAME LEFT_PAR args RIGHT_PAR SEM { printf("Calling
%s\n", $1); };
args:      argList;
argList:   argList COMMA NAME | NAME;
parms:     parmList;
parmList:  parmList COMMA parmTypeList | parmTypeList;
parmTypeList: typeSpec NAME;
varDecl:   VARS typeSpec varDeclList SEM { printf("Var
definition\n"); };
typeSpec:  INTEGER | CHAR;
varDeclList: varDeclList COMMA varDeclInit | varDeclInit;
varDeclInit: varDeclId;
varDeclId: NAME | NAME LEFT_BRA INT RIGHT_BRA;

%%

int yyerror(char *s) {
    fprintf(stderr, "%s in line %d\n", s, yylineno);
    return 0;
}

int main(int argc, char **argv) {
    printf("C Set Parser\n\n");
    if (argc > 1) {
        yyin = fopen(argv[1], "r");
    } else {
        yyin = stdin;
    }
    yyparse();
}

```



```

return 0;
}

```

SCREENSHOTS

ΕΡΩΤΗΜΑ 1 ΣΩΣΤΟ ΠΡΟΓΡΑΜΜΑ

The screenshot shows the Visual Studio Code editor with a C program. The Explorer sidebar on the left shows the project structure. The main editor displays the following code:

```

1  PROGRAM program
2
3  FUNCTION function(INTEGER a, INTEGER b)
4  VARS INTEGER var1, var2, var45[10], var46;
5  VARS CHAR var24, var[20];
6
7  %sxolia
8
9  IF(a<1) THEN
10 B=2;
11 ELSEIF
12 B=3;
13 ELSEIF
14 B=4;
15 ELSE
16 B=5;
17 ENDIF
18
19 SWITCH(b)
20 CASE (1):
21 C=1;
22 CASE (2):
23 C=2;
24 CASE (3):
25 C=3;
26 DEFAULT:
27 C=4;
28 END SWITCH
29
30 WHILE (var10<30)
31 var5 = function1(var1, var2);
32 PRINT("print execution",[var5]);
33
34 %sxolia
35
36 PRINT("HELLO WORLD",[var5]);
37 END WHILE
38
39 FOR counter=1 TO 200 STEP 5

```

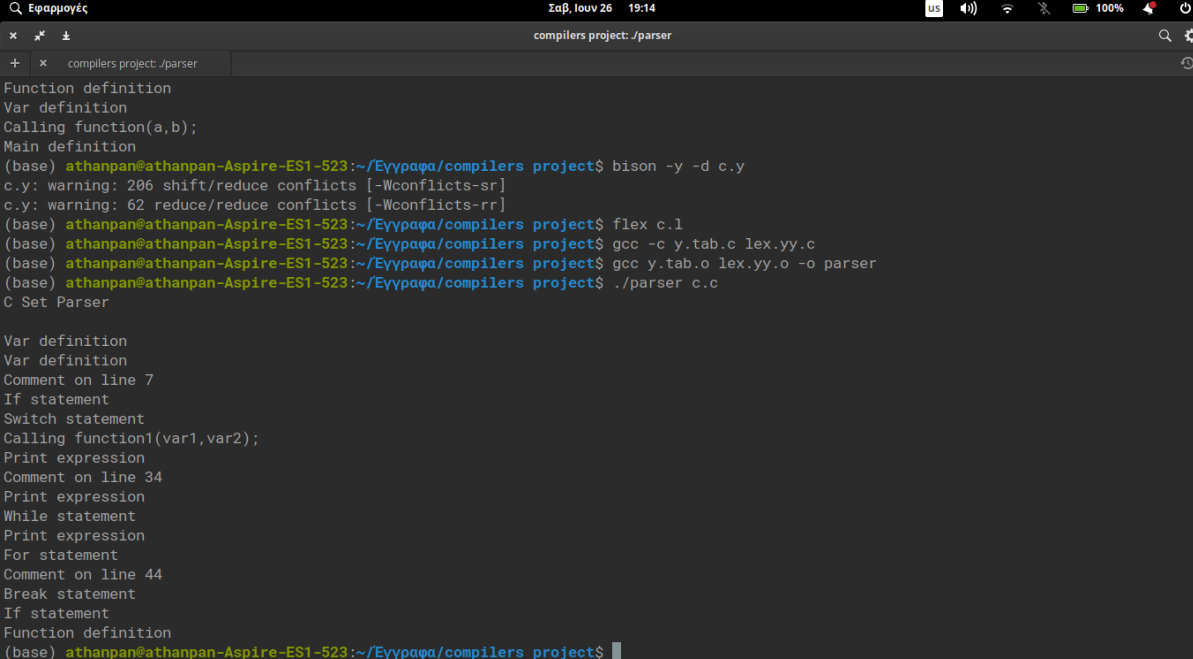
The screenshot shows the continuation of the C program in the Visual Studio Code editor. The code continues from the previous screenshot:

```

40 var25=var4;
41 PRINT("for executing",[var25]);
42 ENDFOR
43
44 %sxolia
45
46 IF(a<1) THEN
47 BREAK;
48 ELSE
49 VAR67=30;
50 ENDIF
51
52 RETURN b;
53 END_FUNCTION
54
55 STARTMAIN
56 VARS INTEGER var100;
57 var100=20;
58 function(a,b);
59 ENDMAIN

```

ΕΡΩΤΗΜΑ 1 ΣΩΣΤΗ ΕΚΤΕΛΕΣΗ TERMINAL

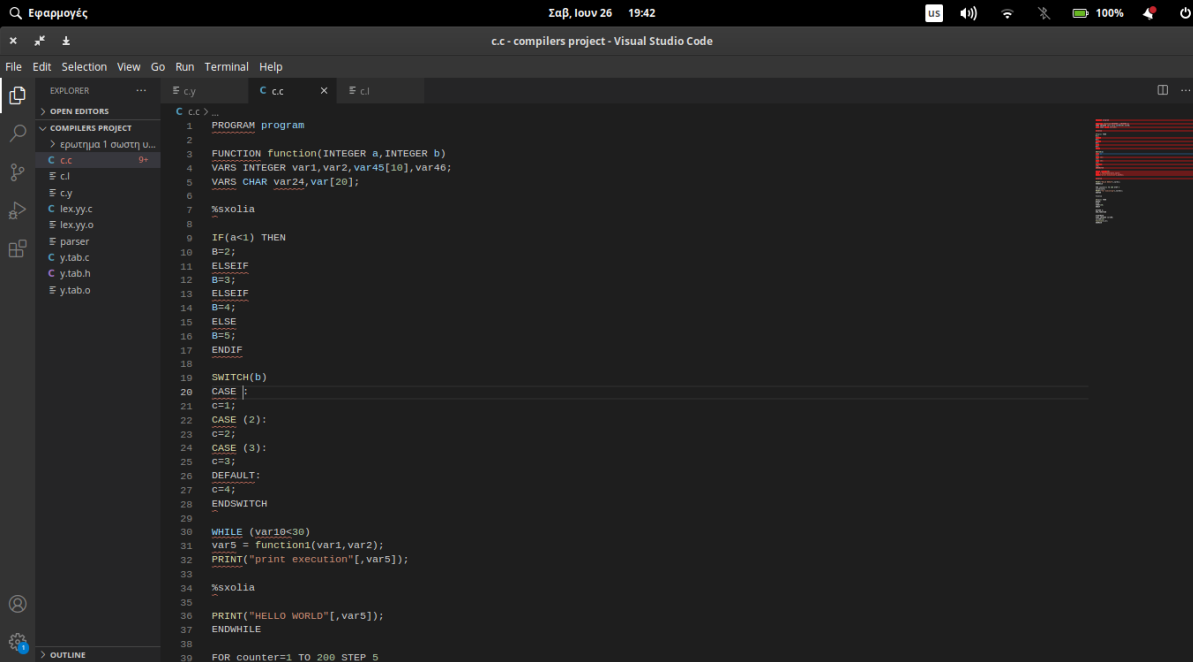


```
Function definition
Var definition
Calling function(a,b);
Main definition
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ bison -y -d c.y
c.y: warning: 206 shift/reduce conflicts [-Wconflicts-sr]
c.y: warning: 62 reduce/reduce conflicts [-Wconflicts-rr]
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ flex c.l
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ gcc -c y.tab.c lex.yy.c
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ gcc y.tab.o lex.yy.o -o parser
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ ./parser c.c
C Set Parser

Var definition
Var definition
Comment on line 7
If statement
Switch statement
Calling function1(var1,var2);
Print expression
Comment on line 34
Print expression
While statement
Print expression
For statement
Comment on line 44
Break statement
If statement
Function definition
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$
```

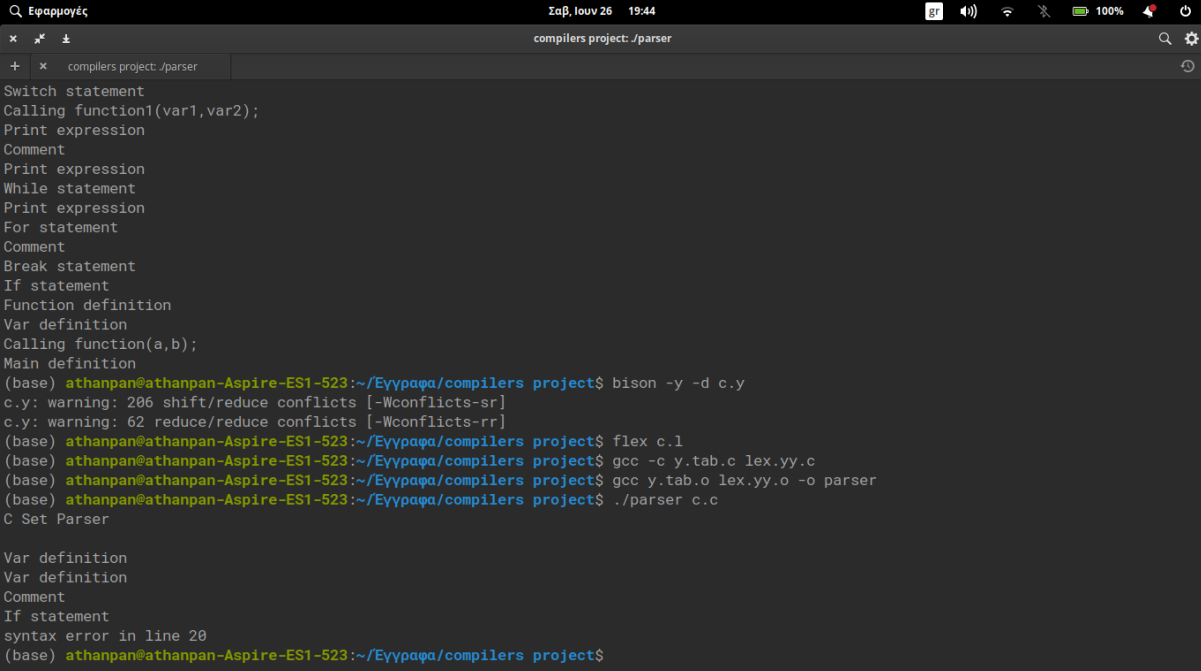
ΕΡΩΤΗΜΑ 1 ΛΑΘΟΣ ΠΡΟΓΡΑΜΜΑ

Παραλείποντας την <έκφραση> στο CASE στην line 20 έχουμε συντακτικό λάθος.



```
1 PROGRAM program
2
3 FUNCTION function(INTEGER a, INTEGER b)
4 VAR: INTEGER var1, var2, var45[10], var46;
5 VAR: CHAR var24, var[20];
6
7 %sxolia
8
9 IF(a<1) THEN
10 B=2;
11 ELSEIF
12 B=3;
13 ELSEIF
14 B=4;
15 ELSE
16 B=5;
17 ENDIF
18
19 SWITCH(b)
20 CASE {
21 C=1;
22 CASE (2):
23 C=2;
24 CASE (3):
25 C=3;
26 DEFAULT:
27 C=4;
28 END SWITCH
29
30 WHILE (var10<30)
31 var5 = function1(var1,var2);
32 PRINT("print execution",[var5]);
33
34 %sxolia
35
36 PRINT("HELLO WORLD",[var5]);
37 END WHILE
38
39 FOR counter=1 TO 200 STEP 5
```

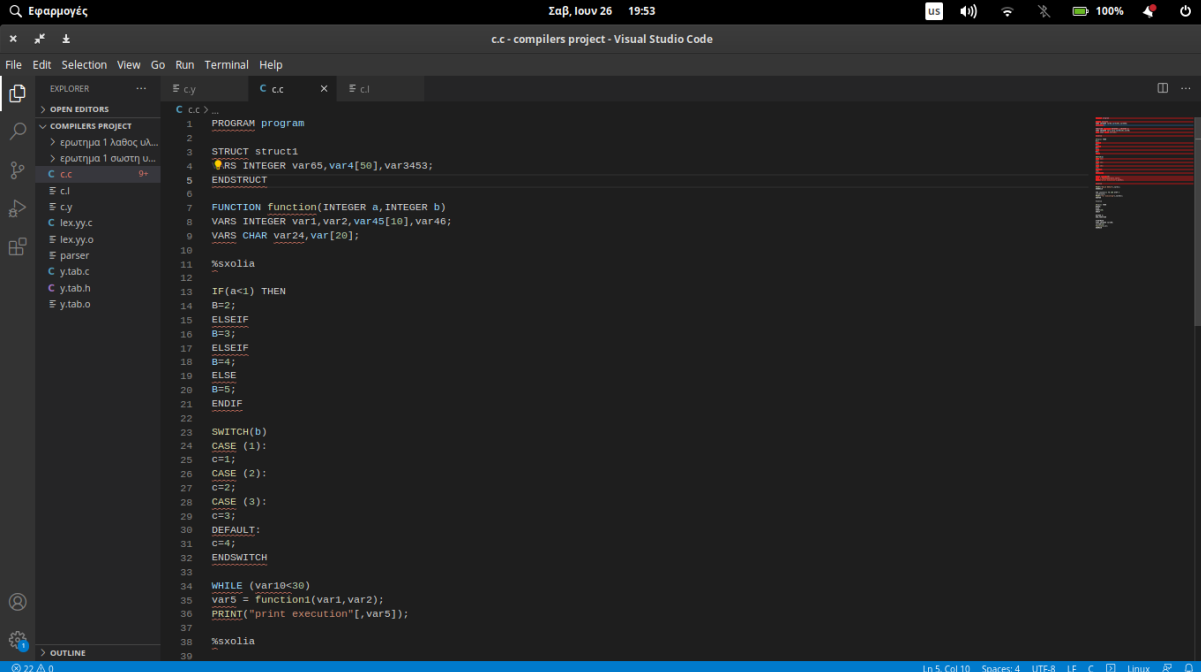
ΕΡΩΤΗΜΑ 1 ΛΑΘΟΣ ΕΚΤΕΛΕΣΗ TERMINAL



```
Switch statement
Calling function1(var1,var2);
Print expression
Comment
Print expression
While statement
Print expression
For statement
Comment
Break statement
If statement
Function definition
Var definition
Calling function(a,b);
Main definition
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ bison -y -d c.y
c.y: warning: 206 shift/reduce conflicts [-Wconflicts-sr]
c.y: warning: 62 reduce/reduce conflicts [-Wconflicts-rr]
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ flex c.l
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ gcc -c y.tab.c lex.yy.c
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ gcc y.tab.o lex.yy.o -o parser
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ ./parser c.c
C Set Parser

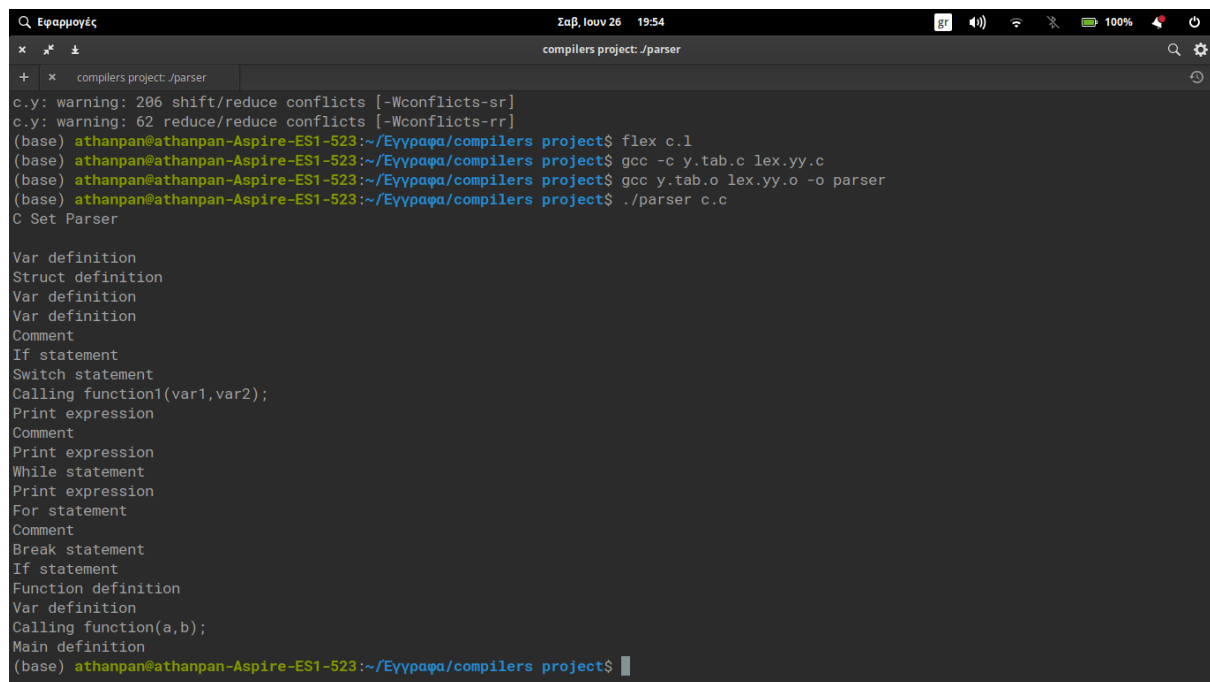
Var definition
Var definition
Comment
If statement
syntax error in line 20
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$
```

ΕΡΩΤΗΜΑ 2 ΣΩΣΤΟ ΠΡΟΓΡΑΜΜΑ



```
1 PROGRAM program
2
3 STRUCT struct1
4     RS INTEGER var65,var4[50],var3453;
5 ENDSTRUCT
6
7 FUNCTION function(INTEGER a,INTEGER b)
8     VARS INTEGER var1,var2,var45[10],var46;
9     VARS CHAR var24,var[20];
10
11     %xollia
12
13 IF(a<1) THEN
14     B=2;
15 ELSEIF
16     B=3;
17 ELSEIF
18     B=4;
19 ELSE
20     B=5;
21 ENDIF
22
23 SWITCH(b)
24 CASE (1):
25     C=1;
26 CASE (2):
27     C=2;
28 CASE (3):
29     C=3;
30 DEFAULT:
31     C=4;
32 END SWITCH
33
34 WHILE (var10<30)
35     var5 = function1(var1,var2);
36     PRINT("print execution",[var5]);
37
38     %xollia
39
```

ΕΡΩΤΗΜΑ 2 ΣΩΣΤΗ ΕΚΤΕΛΕΣΗ TERMINAL

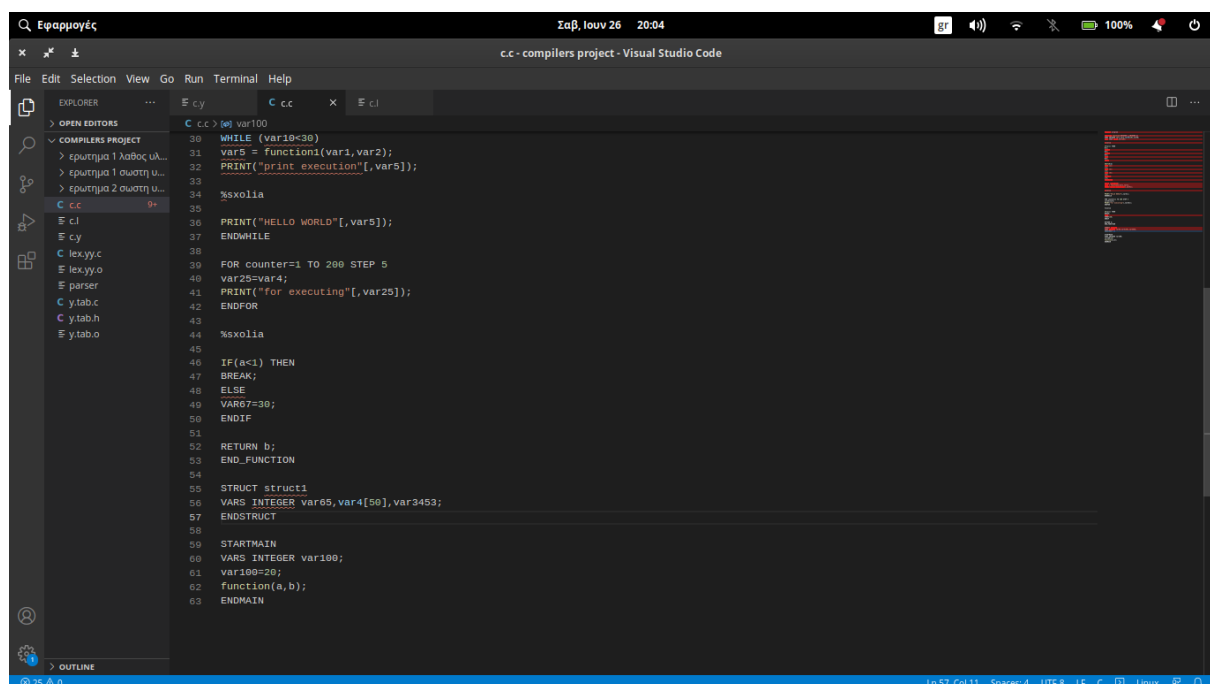


```
c.y: warning: 206 shift/reduce conflicts [-Wconflicts-sr]
c.y: warning: 62 reduce/reduce conflicts [-Wconflicts-rr]
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ flex c.l
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ gcc -c y.tab.c lex.yy.c
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ gcc y.tab.o lex.yy.o -o parser
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ ./parser c.c
C Set Parser

Var definition
Struct definition
Var definition
Var definition
Comment
If statement
Switch statement
Calling function1(var1,var2);
Print expression
Comment
Print expression
While statement
Print expression
For statement
Comment
Break statement
If statement
Function definition
Var definition
Calling function(a,b);
Main definition
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$
```

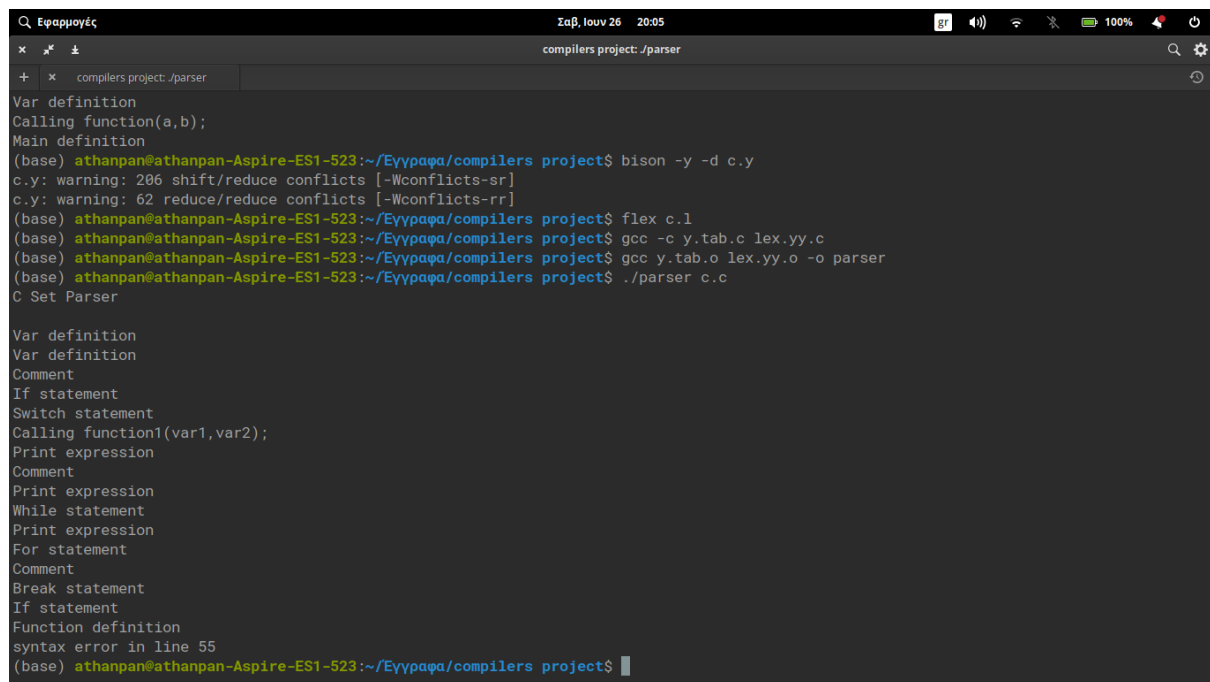
ΕΡΩΤΗΜΑ 2 ΛΑΘΟΣ ΠΡΟΓΡΑΜΜΑ

Τοποθετώντας την struct κατω απο την function έχουμε συντακτικό erro στην line 55



```
1  #include <stdio.h>
2
3  int main()
4  {
5      printf("HELLO WORLD\n");
6
7      struct t1
8      {
9          int a;
10         int b;
11     };
12
13     int var100;
14     var100=30;
15     function1(var1,var2);
16     PRINT("print execution",var5);
17     %sxolia
18
19     PRINT("HELLO WORLD",var5);
20     ENDMILE
21
22     FOR counter=1 TO 200 STEP 5
23     var25=var4;
24     PRINT("for executing",var25);
25     ENDFOR
26
27     %sxolia
28
29     IF(a<1) THEN
30     BREAK;
31     ELSE
32     VAR67=30;
33     ENDIF
34
35     RETURN b;
36     END_FUNCTION
37
38     STRUCT struct1
39     VARS INTEGER var65,var4[50],var3453;
40     ENDSTRUCT
41
42     STARTMAIN
43     VARS INTEGER var100;
44     var100=30;
45     function(a,b);
46     ENDMAIN
47
48     }
```

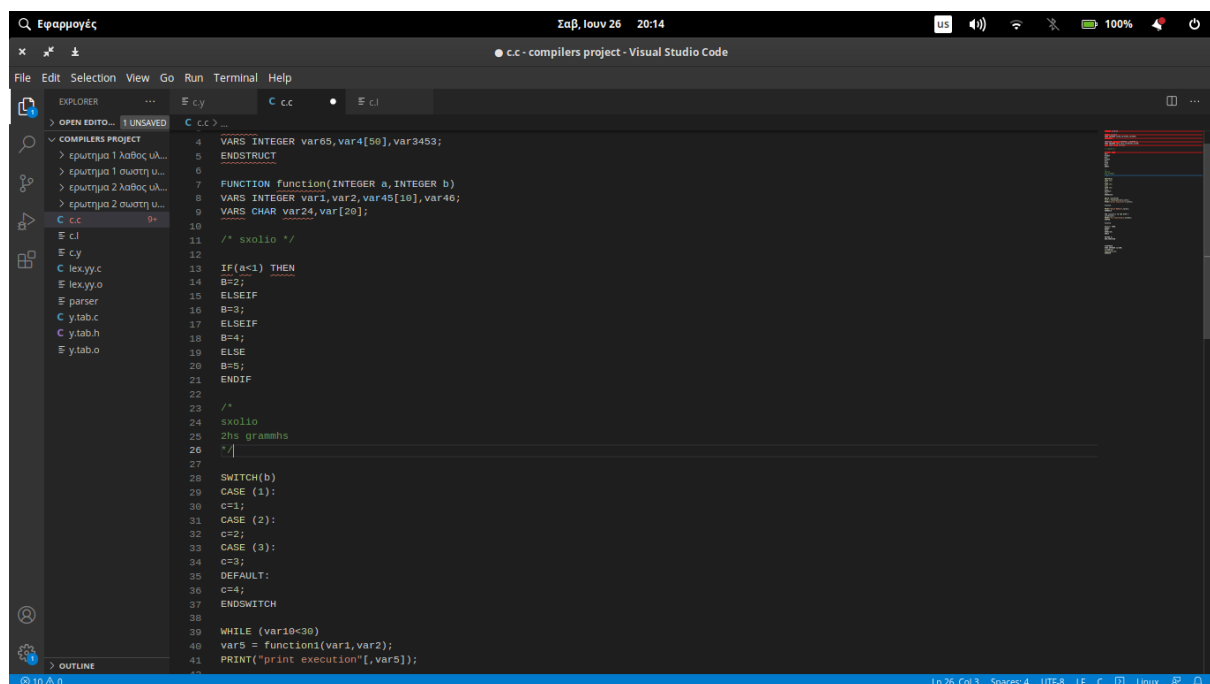
ΕΡΩΤΗΜΑ 2 ΛΑΘΟΣ ΕΚΤΕΛΕΣΗ TERMINAL



```
Var definition
Calling function(a,b);
Main definition
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ bison -y -d c.y
c.y: warning: 206 shift/reduce conflicts [-Wconflicts-sr]
c.y: warning: 62 reduce/reduce conflicts [-Wconflicts-rr]
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ flex c.l
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ gcc -c y.tab.c lex.yy.c
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ gcc y.tab.o lex.yy.o -o parser
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ ./parser c.c
C Set Parser

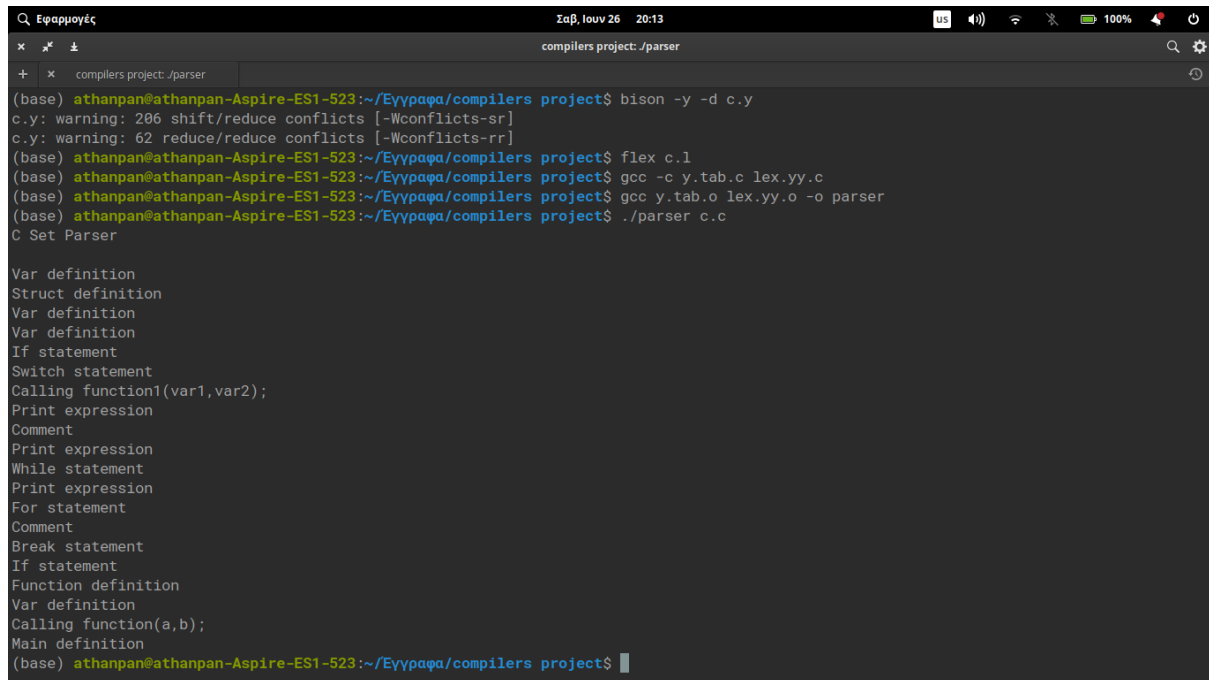
Var definition
Var definition
Comment
If statement
Switch statement
Calling function1(var1,var2);
Print expression
Comment
Print expression
While statement
Print expression
For statement
Comment
Break statement
If statement
Function definition
syntax error in line 55
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$
```

ΕΡΩΤΗΜΑ 4 ΣΩΣΤΟ ΠΡΟΓΡΑΜΜΑ



```
4  VARS INTEGER var65,var4[50],var3453;
5  ENDSTRUCT
6
7  FUNCTION function(INTEGER a,INTEGER b)
8  VARS INTEGER var1,var2,var45[10],var46;
9  VARS CHAR var24,var[20];
10
11  /* sxolio */
12
13  IF(a<1) THEN
14  B=2;
15  ELSEIF
16  B=3;
17  ELSEIF
18  B=4;
19  ELSE
20  B=5;
21  ENDIF
22
23  /*
24  sxolio
25  zhs grammhs
26  */
27
28  SWITCH(b)
29  CASE (1):
30  c=1;
31  CASE (2):
32  c=2;
33  CASE (3):
34  c=3;
35  DEFAULT:
36  c=4;
37  END SWITCH
38
39  WHILE (var10<30)
40  var5 = function1(var1,var2);
41  PRINT("print execution",var5));
```

ΕΡΩΤΗΜΑ 4 ΣΩΣΤΗ ΕΚΤΕΛΕΣΗ TERMINAL

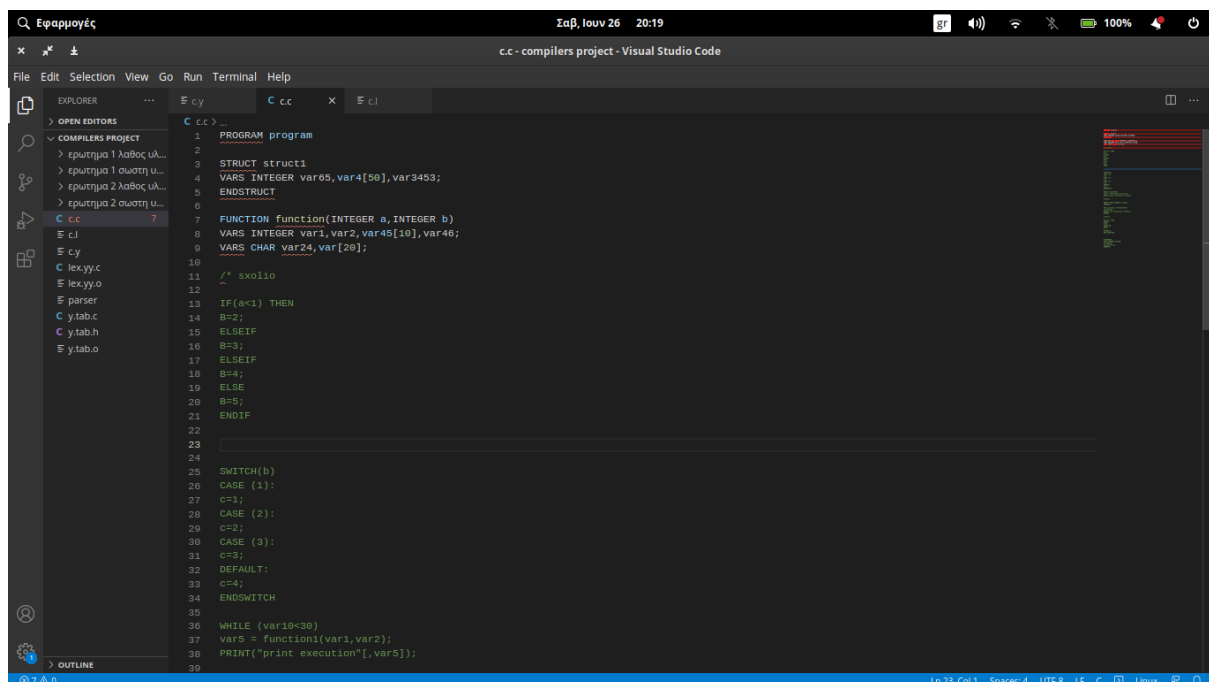


```
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ bison -y -d c.y
c.y: warning: 206 shift/reduce conflicts [-Wconflicts-sr]
c.y: warning: 62 reduce/reduce conflicts [-Wconflicts-rr]
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ flex c.l
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ gcc -c y.tab.c lex.yy.c
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ gcc y.tab.o lex.yy.o -o parser
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ ./parser c.c
C Set Parser

Var definition
Struct definition
Var definition
Var definition
If statement
Switch statement
Calling function1(var1,var2);
Print expression
Comment
Print expression
While statement
Print expression
For statement
Comment
Break statement
If statement
Function definition
Var definition
Calling function(a,b);
Main definition
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$
```

ΕΡΩΤΗΜΑ 4 ΛΑΘΟΣ ΠΡΟΓΡΑΜΜΑ

Μην κλείνοντας το σχόλιο έχουμε error στην line 67



```
1 PROGRAM program
2
3 STRUCT struct1
4 VARS INTEGER var65,var4[50],var3453;
5 ENDSTRUCT
6
7 FUNCTION function(INTEGER a,INTEGER b)
8 VARS INTEGER var1,var2,var45[10],var46;
9 VARS CHAR var24,var[20];
10
11 /* σχολιο
12 ~
13 IF(a<1) THEN
14 B=2;
15 ELSEIF
16 B=3;
17 ELSEIF
18 B=4;
19 ELSE
20 B=5;
21 ENDIF
22
23
24
25 SWITCH(b)
26 CASE (1):
27 c=1;
28 CASE (2):
29 c=2;
30 CASE (3):
31 c=3;
32 DEFAULT:
33 c=4;
34 END SWITCH
35
36 WHILE (var10<30)
37 var5 = function1(var1,var2);
38 PRINT("print execution",[var5]);
39
```

The screenshot shows the Visual Studio Code editor with a C program open. The Explorer sidebar on the left shows a project named 'compilers project' with files 'c.c', 'c.l', 'c.y', 'lex.yy.c', 'lex.yy.o', 'parser', 'y.tab.c', 'y.tab.h', and 'y.tab.o'. The main editor window displays the code for 'c.c', which includes a switch statement, a function call, a print statement, a while loop, a for loop, and a main function. The status bar at the bottom indicates 'Ln 23, Col 1', 'Spaces: 4', 'UTF-8', 'LF', 'C', 'Linux', and '7.0.0'.

```
30 CASE (3):
31     c=3;
32 DEFAULT:
33     c=4;
34 END SWITCH
35
36 WHILE (var10<30)
37     var5 = function1(var1,var2);
38     PRINT("print execution",[var5]);
39
40     %sxolia
41
42     PRINT("HELLO WORLD",[var5]);
43 END WHILE
44
45 FOR counter=1 TO 200 STEP 5
46     var25=var4;
47     PRINT("for executing",[var25]);
48 END FOR
49
50     %sxolia
51
52 IF (a<1) THEN
53     BREAK;
54 ELSE
55     VAR67=30;
56 END IF
57
58 RETURN b;
59 END_FUNCTION
60
61
62
63 STARTMAIN
64 VARS INTEGER var100;
65 var100=20;
66 function(a,b);
67 ENDMAIN
```

ΕΡΩΤΗΜΑ 4 ΛΑΘΟΣ ΕΚΤΕΛΕΣΗ TERMINAL

The screenshot shows a terminal window with the following commands and output:

```
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ bison -y -d c.y
c.y: warning: 206 shift/reduce conflicts [-Wconflicts-sr]
c.y: warning: 62 reduce/reduce conflicts [-Wconflicts-rr]
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ flex c.l
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ gcc -c y.tab.c lex.yy.c
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ gcc y.tab.o lex.yy.o -o parser
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$ ./parser c.c
C Set Parser

Var definition
Struct definition
Var definition
Var definition
syntax error in line 67
(base) athanpan@athanpan-Aspire-ES1-523:~/Εγγραφα/compilers project$
```

ΣΧΟΛΙΑ - ΠΑΡΑΔΟΧΕΣ

- Μετά την λέξη PROGRAM και το όνομα του προγράμματος που είναι μια ακολουθία χαρακτήρων υπάρχει υποχρεωτική αλλαγή γραμμής.
- Μετά την λέξη FUNCTION και το όνομα της συνάρτησης, το οποίο είναι ακολουθία χαρακτήρων και την παρένθεση μέσα στην οποία βρίσκεται η λίστα των παραμέτρων χωρισμένη με κόμμα, υπάρχει υποχρεωτική αλλαγή γραμμής.
- Μετά την δεσμευμένη λέξη VARS ακολουθεί χωρίς αλλαγή γραμμής ο τύπος μεταβλητών (CHAR, INTEGER) και στην συνέχεια γίνεται η δήλωση των ονομάτων των μεταβλητών. Π.χ. VARS INTEGER var1, var2, var3[10]. την
- Μετά το END_FUNCTION για να δηλώσουμε άλλη συνάρτηση ή την main υπάρχει υποχρεωτική αλλαγή γραμμής.
- Μετά το ENDSTRUCT για να δηλώσουμε άλλη συνάρτηση ή την main ή άλλο struct υπάρχει υποχρεωτική αλλαγή γραμμής.
- Μετά την δήλωση σχολίων μια γραμμής υπάρχει υποχρεωτική αλλαγή γραμμής
- Για την δήλωση των εντολών προγράμματος του βρόχου WHILE απαιτείται αλλαγή γραμμής από την δεσμευμένη λέξη while και την συνθήκη
- Απαιτείται αλλαγή γραμμής για την δήλωση των εντολών προγράμματος , του βρόχου IF , από την δεσμευμένη λέξη IF και την συνθήκη. Το ίδιο ισχύει και για τις δεσμευμένες λέξεις ELSE και ELSE IF.
- Απαιτείται αλλαγή γραμμής από την SWITCH(έκφραση) για την δήλωση της CASE(έκφραση) και αλλαγή γραμμής για την δήλωση των εντολών προγράμματος
- Απαιτείται αλλαγή γραμμής για την δήλωση των εντολών προγράμματος , του βρόχου FOR.