

# Flex - Bison

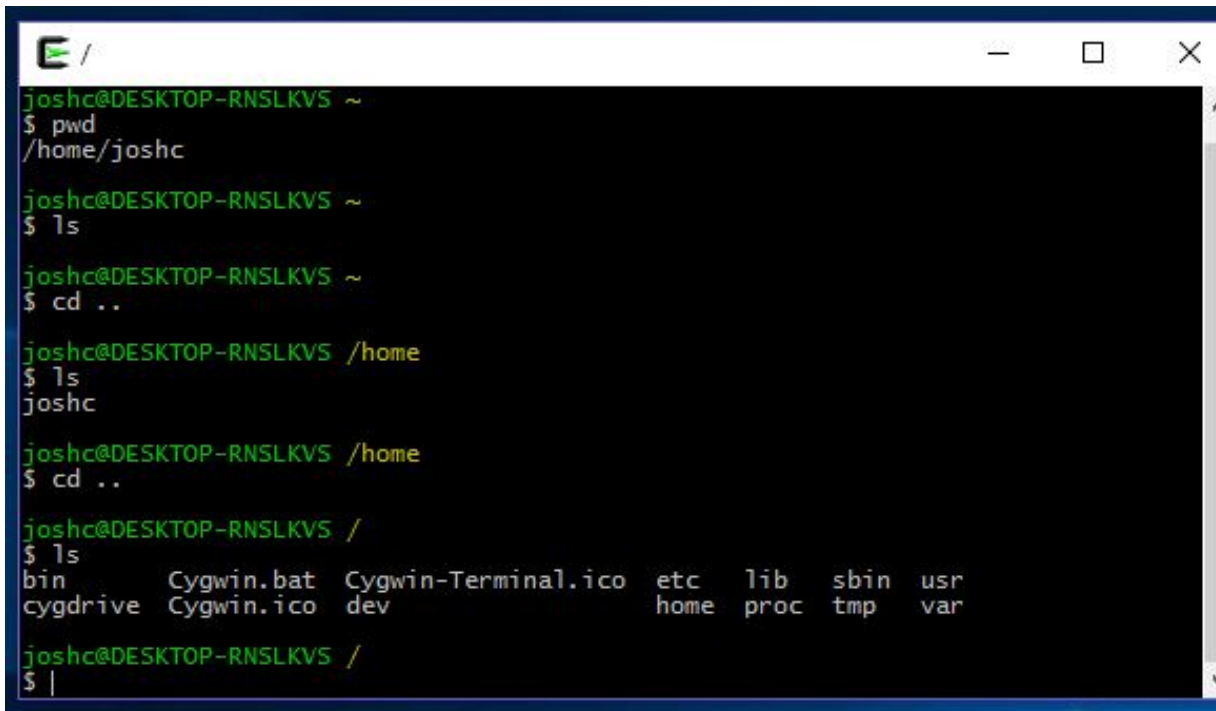
## *Αρχές Γλωσσών Προγραμματισμού και Μεταφραστών*

Εγκατάσταση και Χρήση  
των Flex, Bison

# Χρήσιμοι σύνδεσμοι

- <http://flex.sourceforge.net/>
- <http://www.gnu.org/software/bison/>

Για χρήστες με Windows - <http://www.cygwin.com/>



```

joshc@DESKTOP-RNSLKVS ~
$ pwd
/home/joshc

joshc@DESKTOP-RNSLKVS ~
$ ls

joshc@DESKTOP-RNSLKVS ~
$ cd ..

joshc@DESKTOP-RNSLKVS /home
$ ls
joshc

joshc@DESKTOP-RNSLKVS /home
$ cd ..

joshc@DESKTOP-RNSLKVS /
$ ls
bin          Cygwin.bat  Cygwin-Terminal.ico  etc  lib  sbin  usr
cygdrive     Cygwin.ico  dev                  home proc tmp  var

joshc@DESKTOP-RNSLKVS /
$ |

```

# Εγκατάσταση Flex Bison σε Windows (1/6)

`>=bison 2.3-1, flex>= 2.5.4a-3, >=gcc-core 3.4.4-1, make >=3.81-1`



The screenshot shows the Cygwin website at `cygwin.com/install.html`. The page features a navigation menu on the left with categories like 'Cygwin', 'Cygwin/X', 'Community', 'Documentation', 'Contributing', and 'Related Sites'. The main content area has a large 'Cygwin' logo with the tagline 'Get that *Linux* feeling - on Windows'. Below the logo is a section titled 'Installing and Updating Cygwin Packages' with several sub-sections: 'Installing and Updating Cygwin for 64-bit versions of Windows', 'Signing key transition', and 'General installation notes'. The 'General installation notes' section contains several paragraphs of text explaining the installation process, including the use of the `setup-x86_64.exe` binary and the `--no-admin` option. A 'Q: How do I add a package to my existing Cygwin installation?' section is also visible, followed by an 'A:' response and a 'Tip'.

Cygwin  
Install Cygwin  
Update Cygwin  
Search Packages  
Licensing Terms

Cygwin/X

Community  
Reporting Problems  
Mailing Lists  
Newsgroups  
IRC channels  
Gold Stars  
Mirror Sites  
Donations

Documentation  
FAQ  
User's Guide  
API Reference  
Acronyms

Contributing  
Snapshots  
Source in Git  
Cygwin Packages

Related Sites

## Cygwin

Get that *Linux* feeling - on Windows

### Installing and Updating Cygwin Packages

#### Installing and Updating Cygwin for 64-bit versions of Windows

Run [setup-x86\\_64.exe](#) any time you want to update or install a Cygwin package for 64-bit windows. The [signature](#) for [setup-x86\\_64.exe](#) can be used to verify the validity of this binary.

#### Signing key transition

The key used to sign setup binaries has been updated. During the transition period, signatures are made using both old (676041BA) and new (1A698DE9E2E56300) public keys [here](#).

See [this mail](#) for more details.

#### General installation notes

When installing packages for the first time, the setup program *does not install every package*. Only the **minimal base packages** from the Cygwin distribution are installed by default, which takes up about 100 MB.

Clicking on categories and packages in the setup program package installation screen allows you to select what is installed or updated.

Individual packages like *bash*, *gcc*, *less*, etc. are released independently of the Cygwin DLL, so the Cygwin DLL version is not useful as a general program release number. The setup program tracks the versions of all installed components and provides the mechanism for **installing or updating** everything available from this site for Cygwin.

Once you've installed your desired subset of the Cygwin distribution, the setup program will remember what you selected, so re-running it will update your system with any new package releases.

On Windows Vista and later, the setup program will check by default if it runs with administrative privileges and, if not, will try to elevate the process. If you want to avoid this behaviour and install under an unprivileged account just for your own usage, run setup with the `--no-admin` option.

#### Q: How do I add a package to my existing Cygwin installation?

A: Run the setup program and select the package you want to add.

Tip: if you don't want to also upgrade existing packages, select 'Keep' at the top-right of the package chooser page.







# Εγκατάσταση Flex Bison σε Windows (5/6)

Cywin Setup - Select Packages

Select Packages  
Select packages to install

View Full Search make Clear  Keep  Best  Sync  Test

Package	Current	New	Src?	Categories	Size	Description
WindowMaker		Skip	<input type="checkbox"/>	X11	1,373k	GNUstep window manager
WindowMaker-debuginfo		Skip	<input type="checkbox"/>	Debug	1,983k	Debug info for WindowMaker
automake		Skip	<input type="checkbox"/>	Devel	3k	Wrapper for multiple versions of Automake
automake1.10		Skip	<input type="checkbox"/>	Devel	689k	(1.10) a tool for generating GNU-compliant Makefiles
automake1.11		Skip	<input type="checkbox"/>	Devel	836k	(1.11) a tool for generating GNU-compliant Makefiles
automake1.12		Skip	<input type="checkbox"/>	Devel	706k	(1.12) a tool for generating GNU-compliant Makefiles
automake1.13		Skip	<input type="checkbox"/>	Devel	749k	(1.13) a tool for generating GNU-compliant Makefiles
automake1.14		Skip	<input type="checkbox"/>	Devel	773k	(1.14) a tool for generating GNU-compliant Makefiles
automake1.15		Skip	<input type="checkbox"/>	Devel	598k	(1.15) a tool for generating GNU-compliant Makefiles
automake1.16		Skip	<input type="checkbox"/>	Devel	844k	(1.16) a tool for generating GNU-compliant Makefiles
automake1.4		Skip	<input type="checkbox"/>	Devel	248k	(1.4) a tool for generating GNU-compliant Makefiles
automake1.5		Skip	<input type="checkbox"/>	Devel	332k	(1.5) a tool for generating GNU-compliant Makefiles
automake1.6		Skip	<input type="checkbox"/>	Devel	365k	(1.6) a tool for generating GNU-compliant Makefiles
automake1.7		Skip	<input type="checkbox"/>	Devel	426k	(1.7) a tool for generating GNU-compliant Makefiles
automake1.8		Skip	<input type="checkbox"/>	Devel	499k	(1.8) a tool for generating GNU-compliant Makefiles
automake1.9		Skip	<input type="checkbox"/>	Devel	557k	(1.9) a tool for generating GNU-compliant Makefiles
bash-completion-cmake		Skip	<input type="checkbox"/>	Shells	3k	Cross-platform makefile generation system (bash-completion)
cmake		Skip	<input type="checkbox"/>	Devel	6,297k	Cross-platform makefile generation system
cmake-debuginfo		Skip	<input type="checkbox"/>	Debug	326,610k	Debug info for cmake
cmake-doc		Skip	<input type="checkbox"/>	Devel	1,907k	Cross-platform makefile generation system (documentation)
cmake-gui		Skip	<input type="checkbox"/>	Devel	2,082k	Cross-platform makefile generation system (GUI)
emacs-cmake		Skip	<input type="checkbox"/>	Editors	5k	Cross-platform makefile generation system (Emacs mode)
extra-cmake-modules		Skip	<input type="checkbox"/>	Devel	281k	Extra CMake Modules for KDE
gcc-tools-epoch1-automake		Skip	<input type="checkbox"/>	Devel	419k	(gcc-special) a tool for generating GNU-compliant Makefiles
gcc-tools-epoch2-automake		Skip	<input type="checkbox"/>	Devel	589k	(gcc-special) a tool for generating GNU-compliant Makefiles
gccmakedep		Skip	<input type="checkbox"/>	Devel	6k	X Makefile dependency tool for GCC
imake		Skip	<input type="checkbox"/>	Devel	35k	X Imake legacy build system
imake-debuginfo		Skip	<input type="checkbox"/>	Debug	64k	Debug info for imake
libWMaker-devel		Skip	<input type="checkbox"/>	Libs	3k	Window Maker interface library (development)
libWMaker1		Skip	<input type="checkbox"/>	Libs	4k	Window Maker interface library (runtime)
libpagemaker-tools		Skip	<input type="checkbox"/>	Graphics	5k	MS Publisher file converters
libpagemaker0.0-debuginfo		Skip	<input type="checkbox"/>	Debug	591k	Debug info for libpagemaker0.0
libpagemaker0.0-devel		Skip	<input type="checkbox"/>	Libs	2k	Adobe PageMaker import filter library (development)
libpagemaker0.0-doc		Skip	<input type="checkbox"/>	Libs	109k	Adobe PageMaker import filter library (API documentation)
libpagemaker0.0_0		Skip	<input type="checkbox"/>	Libs	65k	Adobe PageMaker import filter library (runtime)
make		4.3-1	<input checked="" type="checkbox"/>	Devel	503k	The GNU version of the 'make' utility
make-debuginfo		Skip	<input type="checkbox"/>	Debug	441k	Debug info for make
makedepend		Skip	<input type="checkbox"/>	Devel	29k	X Makefile dependency tool
makedepend-debuginfo		Skip	<input type="checkbox"/>	Debug	75k	Debug info for makedepend
makepasswd		Skip	<input type="checkbox"/>	Perl, Utils	14k	Generate and encrypt passwords
makeself		Skip	<input type="checkbox"/>	Archive	27k	Utility to generate self-extractable archives
mingw64-686-libpagemaker0.0		Skip	<input type="checkbox"/>	Libs	76k	Adobe PageMaker import filter library for Win32 toolchain
mingw64-686-qt4-qmake		Skip	<input type="checkbox"/>	Devel	7,326k	Qt4 development tools for Win32 toolchain
mingw64-686-qt4-qmake-debuginfo		Skip	<input type="checkbox"/>	Debug	79,179k	Debug info for mingw64-686-qt4-qmake

Hide obsolete packages

< Back Next > Cancel

# Εγκατάσταση Flex Bison σε Windows (6/6)

Cygwin Setup - Select Packages

— □ ×

Select Packages  
Select packages to install



View Full

Search gcc-core

Clear

Keep  Best  Sync  Test

Package	Current	New	Src?	Categories	Size	Description
cygwin32-gcc-core		6.4.0-1	<input type="checkbox"/>	Devel	16.464k	GCC for Cygwin 32bit toolchain (C, OpenMP)
djgpp-gcc-core		5.4.0-1	<input type="checkbox"/>	Devel	7.926k	GCC for DJGPP toolchain (C)
gcc-core		11.2.0-1	<input type="checkbox"/>	Devel	30.745k	GNU Compiler Collection (C, OpenMP)
mingw64-i686-gcc-core		11.2.0-1	<input checked="" type="checkbox"/>	Devel	28.172k	GCC for Win32 (i686-w64-mingw32) toolchain (C, OpenMP)
mingw64-x86_64-gcc-core		11.2.0-1	<input type="checkbox"/>	Devel	28.972k	GCC for Win64 toolchain (C, OpenMP)

Hide obsolete packages

< Back Next > Cancel



# Εγκατάσταση Flex Bison σε Ubuntu

1. `sudo apt-get update`
2. `sudo apt-get upgrade`
3. `sudo apt install make`
4. `sudo apt install gcc`
5. `sudo apt-get install flex bison`

# Ενδεικτικό Παράδειγμα - Project 2021

## Βασική Ιδέα / Υλοποίηση

- Σκοπός της εργαστηριακής άσκησης είναι να εξοικειωθείτε με την περιγραφή μιας γλώσσας σε μορφή BNF, βασικές έννοιες των μεταγλωττιστών και τελικά να υλοποιήσετε δύο από τα τμήματα ενός μεταγλωττιστή, έναν συντακτικό και ένα λεκτικό αναλυτή με τη χρήση των εργαλείων Bison και Flex. Η γλώσσα που θα αναλυθεί είναι μια ψευδογλώσσα, η οποία ακολουθεί τη λογική της γλώσσας C, και περιγράφεται αμέσως μετά.

- ✓ Κάθε πρόγραμμα ξεκινά με τη δεσμευμένη λέξη **PROGRAM** η οποία ακολουθείται από μια ακολουθία χαρακτήρων που υποδηλώνουν το όνομα του προγράμματος. Μετά το όνομα του προγράμματος πρέπει υποχρεωτικά να υπάρχει αλλαγή γραμμής.
- ✓ Στη συνέχεια, υπάρχει (προαιρετικός) ορισμός συναρτήσεων. Οι συναρτήσεις ορίζονται ως εξής:
  - Ξεκινούν με τη δεσμευμένη λέξη **FUNCTION** η οποία ακολουθείται από το όνομα της συνάρτησης και στη συνέχεια, εντός παρενθέσεων, από λίστα των παραμέτρων χωρισμένων με κόμμα.
  - Έπειτα, μετά από αλλαγή γραμμής, υπάρχει το σώμα της συνάρτησης. Στο σώμα της συνάρτησης αρχικά έχουμε την προαιρετική δήλωση μεταβλητών. Η δήλωση γίνεται με τη δεσμευμένη λέξη **VARS**. Κατόπιν, πρώτα ορίζεται ο τύπος δεδομένων των μεταβλητών (**CHAR** και **INTEGER**) ακολουθούμενος από λίστα με τα ονόματα των μεταβλητών (περισσότερες από μια μεταβλητές χωρίζονται με “,”). Στο τέλος πρέπει να υπάρχει “;”.
  - Πρέπει να υπάρχει η δυνατότητα ορισμού πινάκων με τη δήλωση του μήκους (υποχρεωτικά θετικός ακέραιος) εντός [ ].
    - Π.χ. CHAR var1,var2,var3[10],var4;
  - Στη συνέχεια, ακολουθούν οι εντολές του προγράμματος, που αναλύονται στη συνέχεια.
  - Το τέλος της συνάρτησης ορίζεται ρητά με τη δεσμευμένη λέξη **END\_FUNCTION**. Πριν κλείσει η συνάρτηση, θα πρέπει υποχρεωτικά να επιστρέφεται μια τιμή με τη δεσμευμένη λέξη **RETURN** ακολουθούμενη από την τιμή, η οποία μπορεί να είναι είτε όνομα μεταβλητής, είτε κυριολεκτικό (literal).
- ✓ Αμέσως μετά τον ορισμό των συναρτήσεων, ακολουθεί το κύριο μέρος του προγράμματος το οποίο περικλείεται από τις δεσμευμένες λέξεις **STARTMAIN** και **ENDMAIN**. Στο κύριο μέρος του προγράμματος, αρχικά έχουμε την προαιρετική δήλωση μεταβλητών (ακριβώς όπως περιγράφηκε παραπάνω για τις συναρτήσεις), ενώ στη συνέχεια μπορούν να εμφανίζονται με οποιαδήποτε σειρά οποιεσδήποτε από τις εντολές του προγράμματος που περιγράφονται παρακάτω.

# Εντολές Ανάθεσης

Έχουν τη μορφή `<μεταβλητή>=<έκφραση>;`, όπου η έκφραση μπορεί να είναι ένα κυριολεκτικό ή μια σύνθετη παράσταση από συνδυασμό κυριολεκτικών, μεταβλητών και παρενθέσεων ή κλήση συνάρτησης. Επιπλέον, η σύνθετη παράσταση μπορεί να είναι οποιαδήποτε αριθμητική παράσταση που περιλαμβάνει τις πράξεις `+`, `-`, `^`, `*`, `/`.

Π.χ.:

```
var1 = 2;
```

```
var2 = var3 + 5 * 2^(3-7/(var1+var3));
```

```
var4 = var3;
```

```
var5 = function1(var1,var2);
```

Οι μεταβλητές μπορούν να εμφανίζονται σε αυτό το τμήμα του προγράμματος ανεξάρτητα εάν έχουν δηλωθεί στο προηγούμενο τμήμα δήλωσης μεταβλητών ή όχι. Ομοίως και για τις συναρτήσεις, δεν είναι υποχρεωτικό να έχει προηγηθεί δήλωσή τους.

# Εντολές Βρόχου

Υπάρχουν 2 είδη εντολών βρόχου. Το πρώτο είδος έχει την ακόλουθη μορφή:

**WHILE** (συνθήκη)

... εντολές προγράμματος ...

**ENDWHILE**

Όπου **συνθήκη** είναι οποιαδήποτε λογική έκφραση που περιέχει τους συγκριτικούς τελεστές **>**, **<**, **==**, **!=** και τους λογικούς **AND** και **OR**.

Το δεύτερο είδος έχει την ακόλουθη μορφή:

**FOR** counter:=1 **TO** 100 **STEP** 2 (τα νούμερα είναι ενδεικτικά)

... εντολές προγράμματος ...

**ENDFOR**

# Εντολές Ελέγχου

**IF (συνθήκη) THEN**

... εντολές προγράμματος ...

**ELSEIF**

... εντολές προγράμματος ...

**ELSEIF**

... εντολές προγράμματος ...

...

**ELSE**

... εντολές προγράμματος ...

**ENDIF**

**SWITCH(<έκφραση>)**

**CASE(<έκφραση>):**

... εντολές προγράμματος...

**CASE(<έκφραση>):**

... εντολές προγράμματος...

...

**DEFAULT:**

... εντολές προγράμματος...

**ENDSWITCH**

Η εμφάνιση των ELSEIF και ELSE είναι προαιρετική. Όπου **συνθήκη** είναι οποιαδήποτε λογική έκφραση που περιέχει τους συγκριτικούς τελεστές >, <, ==, != και τους λογικούς **AND** και **OR**.

# Εντολές εκτύπωσης - τερματισμού

- **PRINT**("κείμενο "[,var1]);  
(Δηλαδή κείμενο εντός "" κι αμέσως μετά προαιρετικά όνομα μεταβλητής)
- **BREAK**;
- Σχόλια γραμμής μετά από τον χαρακτήρα "%"

# Δομή Προγράμματος Flex

1. Κώδικας C (Δήλωση βιβλιοθηκών/συναρτήσεων)
2. Δήλωση κανονικών εκφράσεων
3. Ορισμός κανόνων για την παραγωγή των tokens
4. Πρόσθετες συναρτήσεις εάν χρειαστούν καλό είναι να δηλώνονται στο flex.



# Απόσπασμα flex ανάλυσης

## 1. Κώδικας C - Εισαγωγή βιβλιοθηκών

```
%{  
#include "myparser.tab.h"  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
%}
```

# Δήλωση κανονικών εκφράσεων

## 2. Κανονικές Εκφράσεις

underscore     " \_ "

digit     [0-9]

letter   [a-zA-Z]

chartype     "CHAR"

inttype "INTEGER"

userdata type "Node"

comparative ">"|"<"|"=="|"!="

datatype {chartype}{inttype}

# Flex – Αναγνώριση tokens

**3. Ορισμός κανόνων για την παραγωγή tokens**, αναγνωρίζοντας συγκεκριμένες λέξεις – κλειδιά ή κανονικές εκφράσεις

```
"*"      {printf("%s",yytext); return '*'; }
"/"      {printf("%s",yytext); return '/'; }
"="      {printf("%s",yytext); return '='; }
"AND"    {printf("%s",yytext); return AND; }
"OR"     {printf("%s",yytext); return OR; }
":="     {printf("%s",yytext); return ASSIGN_OPERATOR; }
"PROGRAM" {printf("%s",yytext); return PROGRAM; }
"VARS"   {printf("%s",yytext); return VARS; }
"TYPEDEF" {printf("%s",yytext); return TYPEDEF; }
"STRUCT" {printf("%s",yytext); return STRUCT; }
"ENDSTRUCT" {printf("%s",yytext); return ENDSTRUCT; }
"FUNCTION" {printf("%s",yytext); return FUNCTION; }
"END_FUNCTION" {printf("%s",yytext); return END_FUNCTION; }
"RETURN"{printf("%s",yytext); return RETURN; }

{strliteral} {printf("%s",yytext); return STRLITERAL; }
{charliteral} {printf("%s",yytext); return CHARLITERAL; }
{linecomment} {printf("%s",yytext); }
{userdatatype} {printf("%s",yytext); return USERDATATYPE; }
{datatype} {printf("%s",yytext); return DATATYPE; }
{name} {printf("%s",yytext); return NAME; }
{number} {printf("%s",yytext); return NUM; }
{comparative} {printf("%s",yytext); return COMP_OPERATOR; }
```

# Δομή Προγράμματος Bison

%{

Κώδικας C

(μακροεντολές, τύποι δεδομένων, δηλώσεις μεταβλητών και συναρτήσεων)

%}

Δηλώσεις Bison

%%

Κανόνες παραγωγής γραμματικής

%%

Κώδικας C

(υλοποίηση συναρτήσεων, main() )

# Απόσπασμα bison ανάλυσης – Κώδικας C

## 1. Κώδικας C – Βιβλιοθήκες / Μεταβλητές

```
%{  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
void yyerror(char *);  
extern FILE *yyin;  
extern FILE *yyout;  
extern int yylineno;  
int line = 0;
```

# Απόσπασμα bison ανάλυσης - Δηλώσεις Bison

## 2. Δηλώσεις Bison – Καταγραφή των tokens που λαμβάνονται σε συνεργασία με τον λεκτικό αναλυτή FLEX.

```
%token PROGRAM NAME ARRAY NUM COMP_OPERATOR AND OR STRLITERAL  
CHARLITERAL
```

```
%token TYPEDEF STRUCT ENDSTRUCT
```

```
%token WHILE ENDWHILE
```

```
%token FOR ASSIGN_OPERATOR TO STEP ENDFOR
```

```
%token IF THEN ELSEIF ELSE ENDIF
```

```
%token SWITCH CASE DEFAULT ENDSWITCH
```

```
%token PRINT
```

```
%token BREAK
```

```
%token NEWLINE
```

# Απόσπασμα bison ανάλυσης - Κανόνες Παραγωγής Γραμματικής

## 3. Κανόνες Παραγωγής Γραμματικής – Όλοι οι επιτρεπτοί κανόνες που αποδέχεται ο bison με την λογική του parse tree.

- ```
%%
```
- `program: program_declaration main_statement newline { printf("Code Parsed successfully!\n");}`  
| `program_declaration function main_statement newline { printf("Code Parsed successfully!\n");}`  
| `program_declaration struct_statement main_statement newline { printf("Code Parsed successfully!\n");}`  
| `program_declaration struct_statement function main_statement newline { printf("Code Parsed successfully!\n");}`;
  - `program_declaration: PROGRAM NAME newline ;`
  - `variable_declaration: VARS datatype declared_variables ';' newline`  
| `variable_declaration VARS datatype declared_variables ';' newline;`
  - `declared_variables: NAME | NAME ARRAY | declared_variables ',' declared_variables;`

## Διαδικασία Εκτέλεσης

- `bison -d parser.y`
- `flex flex.l`
- `gcc -o run lex.yy.c parser.tab.c -lfl`
- `./run <testFile>`

(όπου `testFile` το αρχείο προς εκτέλεση)