

# Δομημένος Προγραμματισμός

Τμήμα Επιχειρηματικού Σχεδιασμού και  
Πληροφοριακών Συστημάτων

**[www.bpis.teicrete.gr](http://www.bpis.teicrete.gr)**



# Γενικές Πληροφορίες

- Σελίδα μαθήματος <http://eclass.teicrete.gr/courses/DBI101/>
- Βιβλίο  
**C Προγραμματισμός, Εισαγωγή στη C++ και τη Java , P. J. Deitel**
- Διδασκαλία  
4 ώρες εργαστήριο  
**Ομάδα Α**  
Δευτέρα 15:00-19:00  
**Ομάδα Β**  
Πέμπτη 11:00-13:00 και Παρασκευή 11:00-13:00

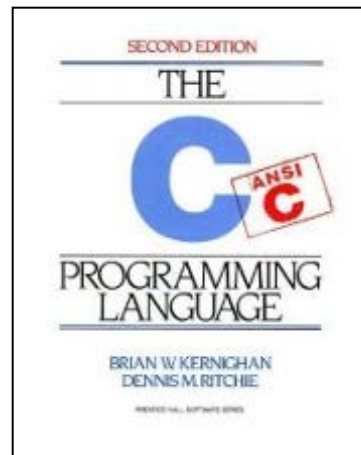
# Βιβλιογραφία

- Αναζητήστε στο Google ([www.google.com](http://www.google.com))

C Programming Language (2nd Edition) **filetype:pdf**

## C Programming Language (2nd Edition)

**Brian W. Kernighan και Dennis M. Ritchie**



# Αξιολόγηση

- Σε κάθε εργαστήριο θα παραδίδετε τον κώδικα που έχετε αναπτύξει στα πλαίσια των ασκήσεων που θα σας δοθούν

Δημιουργία φακέλου (σε κάθε εργαστήριο):

`Domhmenos_AMxxx\05_03_2010\`

π.χ. αν το Α.Μ. σας είναι 50 θα αποθηκεύσετε τις σημερινές ασκήσεις στον φάκελο `Domhmenos_AM50\05_10_2010\`

- Επίσης σε καθορισμένες ημερομηνίες θα αποστέλλετε μέσω `eclass` τις ασκήσεις που σας έχουν δοθεί για το σπίτι

# Αξιολόγηση (Εργαστήριο)

## Αποστολή κώδικα

- Στο email που θα στείλετε ορίζετε σαν **Θέμα** το ΑΜ σας και τον τίτλο του μαθήματος

π.χ. 50 Δομημένος Προγραμματισμός

# Αξιολόγηση

- Ο τελικός βαθμός του εργαστηρίου προκύπτει κατά:

**60%** από την τελική εξέταση του εργαστηρίου

**30%** από τη βαθμολογία των ασκήσεων για το σπίτι

**10%** από τη συμμετοχής σας στο εργαστήριο και βαθμολογία των ασκήσεων του εργαστηρίου

## Κανόνες Ομαλής Διεξαγωγής Διαλέξεων

- Ερχόμαστε στην αίθουσα στην ώρα μας
- Δεν επιτρέπονται το νερό, τα αναψυκτικά, ο καφές, τρόφιμα κτλ. Πετάμε τα σκουπίδια στα καλάθια κατά την αποχώρηση μας από την αίθουσα.
- Τα κινητά τηλέφωνα θα πρέπει να είναι σε **αθόρυβη** λειτουργία και μπορούν να χρησιμοποιηθούν **μόνον σε περίπτωση άμεσης ανάγκης**.

# Οδηγός Επιτυχίας

- Στην αίθουσα θα αναπτυχθούν και θα αναλυθούν πολλά θέματα προγραμματισμού τα οποία είναι πολύ δύσκολα στην κατανόηση τους με ατομική μελέτη.
- Η εκμάθηση προγραμματισμού Η/Υ απαιτεί συνεχή μάθηση και πειραματισμό. Μαθαίνουμε από την συνεχή επίλυση προβλημάτων με την βοήθεια των καθηγητών.
- Είμαστε εδώ για να βοηθήσουμε. Συνεργαστείτε μαζί μας μελετώντας συνεχώς την καινούργια γνώση στον προγραμματισμό.

**Διατυπώνετε ελεύθερα τις απορίες σας. Μια απορία δική σας ίσως να είναι κομμάτι γνώσης για τον συνάδελφο σας.**

# Αντικείμενο Μαθήματος

- Εισαγωγή στην λογική του προγραμματισμού ηλεκτρονικών υπολογιστών.
- Κατανόηση της φιλοσοφίας και των αρχών που διέπουν τον προγραμματισμό
- Εκμάθηση των βασικών μηχανισμών γραφής προγραμμάτων
- Εφαρμογή και πειραματισμό της γνώσης σε μια γλώσσα προγραμματισμού υψηλού επιπέδου

## Θεματικές Ενότητες Μαθήματος

- Εισαγωγή στον προγραμματισμό
  - Ιστορική αναδρομή
  - Από το πρόβλημα στο πρόγραμμα
  - Βασικά στοιχεία της C (Λεξιλόγιο, Σύνταξη κτλ.)
- Μεταβλητές και συναρτήσεις της C
  - Η έννοια της μεταβλητής και οι τύποι της
  - Συναρτήσεις εισόδου εξόδου και αλληλεπίδραση με τον χρήστη
- Τελεστές, διατύπωση εκφράσεων στη C, και έλεγχος ροής
  - Κατηγορίες εκφράσεων και τελεστών.
  - Μετατροπές τύπων
  - Έλεγχος ροής με `if ... else` και υποθετικός τελεστής, `Switch`  
...

## Θεματικές Ενότητες Μαθήματος

- Επαναλήψεις και βρόχοι
  - βρόχος while-do, do-while
  - βρόχος for ...
  - μετασχηματισμός while – for
- Πίνακες και αλφαριθμητικά
  - Μονοδιάστατοι, πολυδιάστατοι πίνακες
  - Αλφαριθμητικά και παρουσίαση σχετικών συναρτήσεων της C
- Απαριθμητικοί τύποι και δομές
  - Απαριθμητικοί τύποι
  - Δομές (structures) και χρήση αυτών
- Συναρτήσεις
  - Δήλωση, ορισμός και κλήση συναρτήσεων
  - Εμβέλεια μεταβλητών

## Θεματικές Ενότητες Μαθήματος

- Δείκτες στη C
  - Η έννοια του δείκτη (δήλωση και χρήση αυτού)
  - Δείκτες σε συναρτήσεις και δομές
- Διαχείριση αρχείων
  - Άνοιγμα και κλείσιμο αρχείων
  - Εγγραφή και ανάγνωση αρχείων
  - Συναρτήσεις της C για είσοδο/έξοδο σε αρχεία
  - Ανάγνωση, εγγραφή σε δυαδικά αρχεία
- Δυναμική διαχείριση μνήμης
  - Οι συναρτήσεις malloc, calloc και free
  - Δείκτες σε δείκτες
  - Δείκτες και συναρτήσεις σε αλφαριθμητικά

# Η γλώσσα προγραμματισμού C

- Δημιουργήθηκε από τον Dennis Ritchie στα Bell Labs USA το 1972.
- Δημιουργήθηκε για τη σχεδίαση του λειτουργικού συστήματος UNIX.
- Είχε γρήγορη εξάπλωση.

## Η γλώσσα προγραμματισμού C

- Η C είναι ισχυρή και ευέλικτη για πολλές εφαρμογές.
- Είναι δημοφιλής άρα υπάρχουν ποικιλία μεταγλωττιστών και χρήσιμων βοηθημάτων.
- Προγράμματα της C μπορούν να εκτελεστούν σε διαφορετικά συστήματα υπολογιστών.

# Η γλώσσα προγραμματισμού C

- Η C είναι μία γλώσσα λίγων λέξεων.  
Μία γλώσσα με περισσότερες λέξεις κλειδιά **ΔΕΝ** είναι πιο ισχυρή.
- Ο κώδικας της C μπορεί (και πρέπει) να γραφεί σε ρουτίνες που λέγονται *συναρτήσεις*.
- Η C++ (αντικειμενοστρεφής ή object oriented) είναι υπερσύνολο της C. Όσα ισχύουν στη C, ισχύουν και στη C++.

# Η γλώσσα προγραμματισμού C

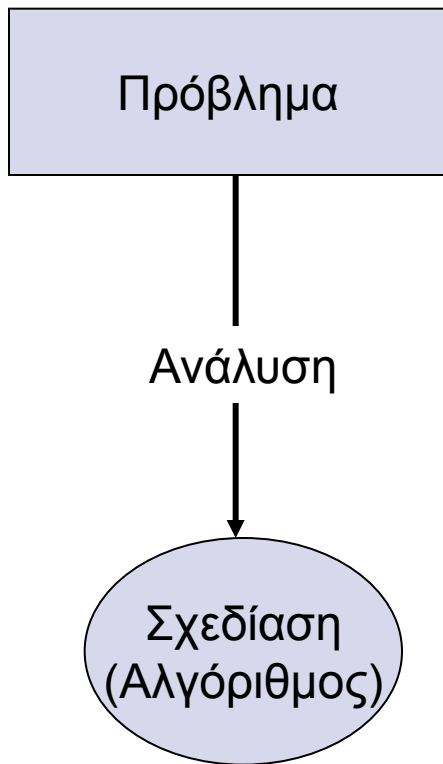
- Συναρτήσεις σε C μπορούν να χρησιμοποιηθούν ξανά σε άλλες εφαρμογές.

**Με εφόδιο τη C αποκτάτε ισχυρά θεμέλια για εκμάθηση άλλων γλωσσών προγραμματισμού όπως Java, C++, κτλ.**

## Από το Πρόβλημα στο Πρόγραμμα

- Ανάλυση του προβλήματος
- Επιινόηση & Σχεδιασμός λύσης (Αλγόριθμο)
- Μεταφορά Αλγορίθμου σε Κώδικα (πηγαίο)
- Μεταγλώττιση πηγαίου κώδικα
- Παραγωγή του αντικείμενου κώδικα
- Σύνθεση του αντικείμενου κώδικα
- Εκτέλεση προγράμματος

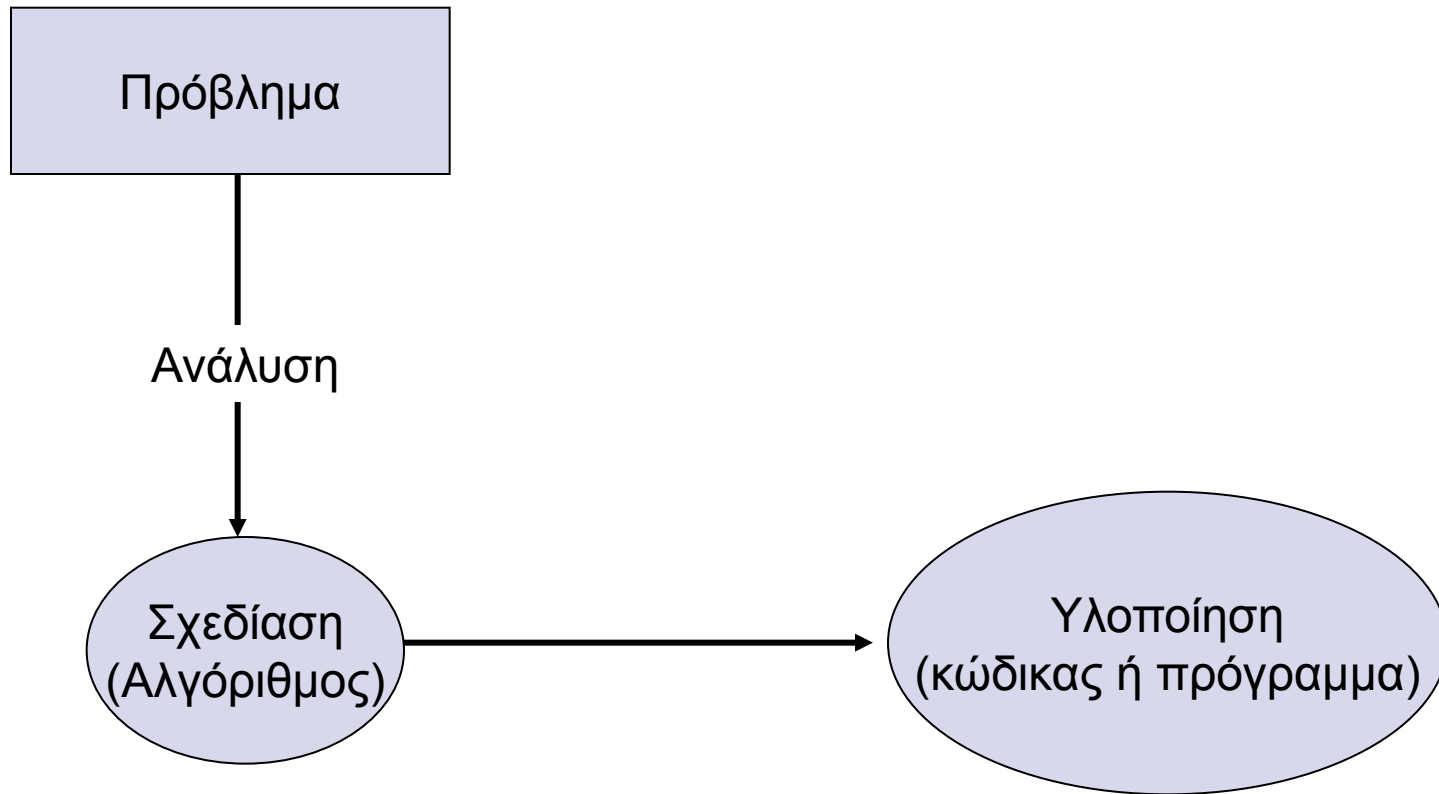
# Από το Πρόβλημα στο Πρόγραμμα



# Αλγόριθμος

- Μία **υπολογιστική διαδικασία** που αποτελείται από μία πεπερασμένη σειρά αυστηρώς καθορισμένων υπολογιστικών βημάτων με σκοπό τη λύση ενός προβλήματος
- Κάθε αλγόριθμος δέχεται κάποια δεδομένα σαν **είσοδο** και παράγει κάποια αποτελέσματα (τη λύση του προβλήματος) σαν **έξοδο**.
- Τρόποι παρουσίασης ενός αλγόριθμου:
  - φυσική γλώσσα
  - διάγραμμα ροής (Flow chart)
  - Ψευδοκώδικας (pseudocode)

# Από το Πρόβλημα στο Πρόγραμμα



## Πρόγραμμα – Γλώσσα Προγραμματισμού

- **Πρόγραμμα** είναι η έκφραση ενός αλγορίθμου (η μεταφορά του σε εντολές) σε μια γλώσσα προγραμματισμού
- **Γλώσσα προγραμματισμού** είναι μια συστηματική σημειογραφία με την οποία περιγράφουμε με ακρίβεια υπολογιστικές διεργασίες ή διαδικασίες

## Πρόγραμμα – Γλώσσα Προγραμματισμού

- **Υπολογιστική διεργασία ή διαδικασία** είναι ένα σύνολο από υπολογιστικά βήματα που μπορεί να εκτελέσει ένας υπολογιστής προκειμένου να διεκπεραιώσει ένα συγκεκριμένο έργο
- Γλώσσα C ως επιλογή για την υλοποίηση προγραμμάτων στον μάθημα
  - Υποστηρίζει δομημένο προγραμματισμό
  - Αποτελεσματική σε παραγωγή αξιόπιστων και γρήγορων σε εκτέλεση προγράμματα
  - Ευρέως διαδομένη

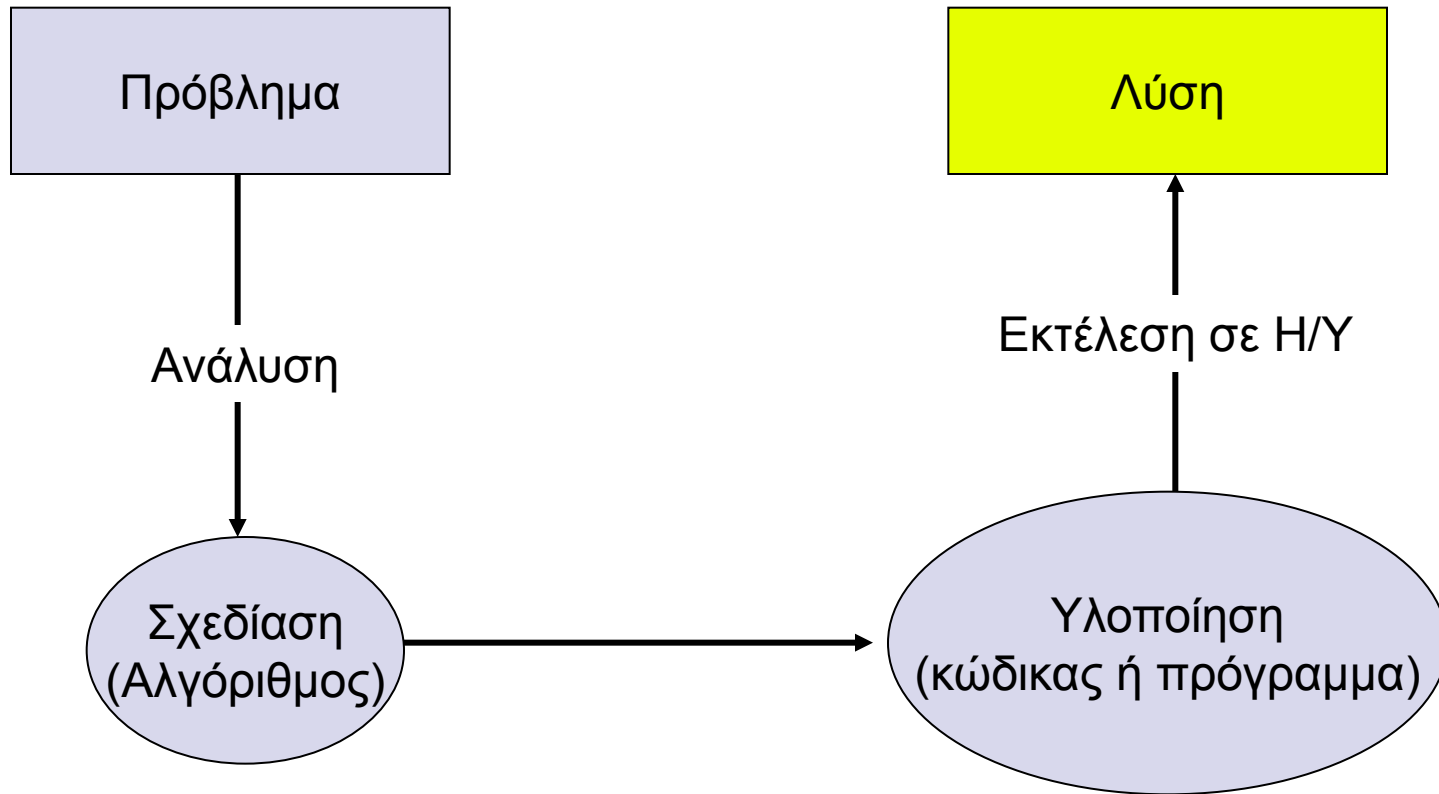
# Παράδειγμα

- Πρόβλημα  
Να βρεθεί ο μέσος όρος  $N$  πραγματικών αριθμών
- Ανάλυση
  - Δεδομένα: οι αριθμοί  $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_N$
  - Ζητούμενο: ο μέσος όρος
- Σχεδίαση – Αλγόριθμος
  - Διάβασε τα δεδομένα
  - Υπολόγισε το μέσο όρο
  - Εκτύπωσε το αποτέλεσμα στην οθόνη

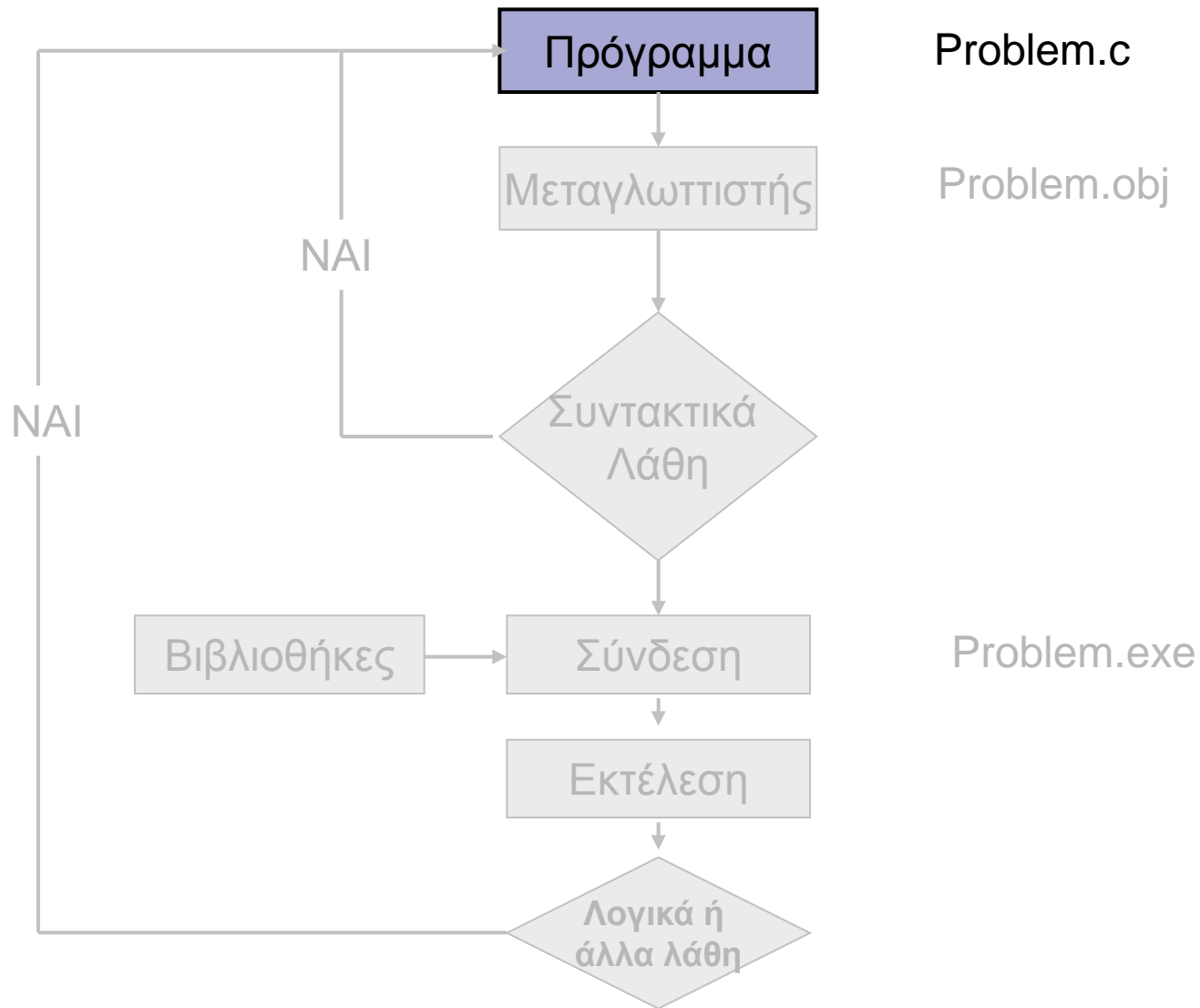
# Παράδειγμα

- Σχεδίαση – Αλγόριθμος
  - Θέσε τη τιμή μηδέν στο  $\Sigma$
  - Επανάλαβε το παρακάτω  $N$  φορές
    - διάβασε ένα αριθμό από τα δεδομένα και πρόσθεσε τον στο άθροισμα  $\Sigma$
  - Διαίρεσε το άθροισμα  $\Sigma$  με το πλήθος των τιμών  $N$  και καταχώρησε το αποτέλεσμα στο  $MO$   
( $MO = \Sigma / N$ )
  - Εκτύπωσε τη τιμή  $MO$

# Από το Πρόβλημα στο Πρόγραμμα



# Κύκλος Υλοποίησης



# Πηγαίο Πρόγραμμα

- Είναι το αρχικό πρόγραμμα που γράφεται από τον προγραμματιστή (source code)
- Για τη σύνταξή του απαιτείται απλά ένας κειμενογράφος (editor)
- Στη γλώσσα προγραμματισμού C αποθηκεύεται σαν αρχείο με κατάληξη **.c**

# Παράδειγμα

- Υλοποίηση – Πρόγραμμα

```
/******
```

```
Eyresh mesou orou 5 arithmwn
```

```
*****/
```

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    float s=0, a, mo;    // Orismos metablhtwn mo: mesos oros, s: a8roisma
```

```
    int i,N=5;          // Plh8os ari8mwn
```

```
    for (i=1;i<=N;i++)
```

```
    {
```

```
        printf("Type a number: ");
```

```
        scanf("%f",&a);    // Eisodos ari8mwn
```

```
        s += a;            //Pros8esh ari8mou sto a8roisma
```

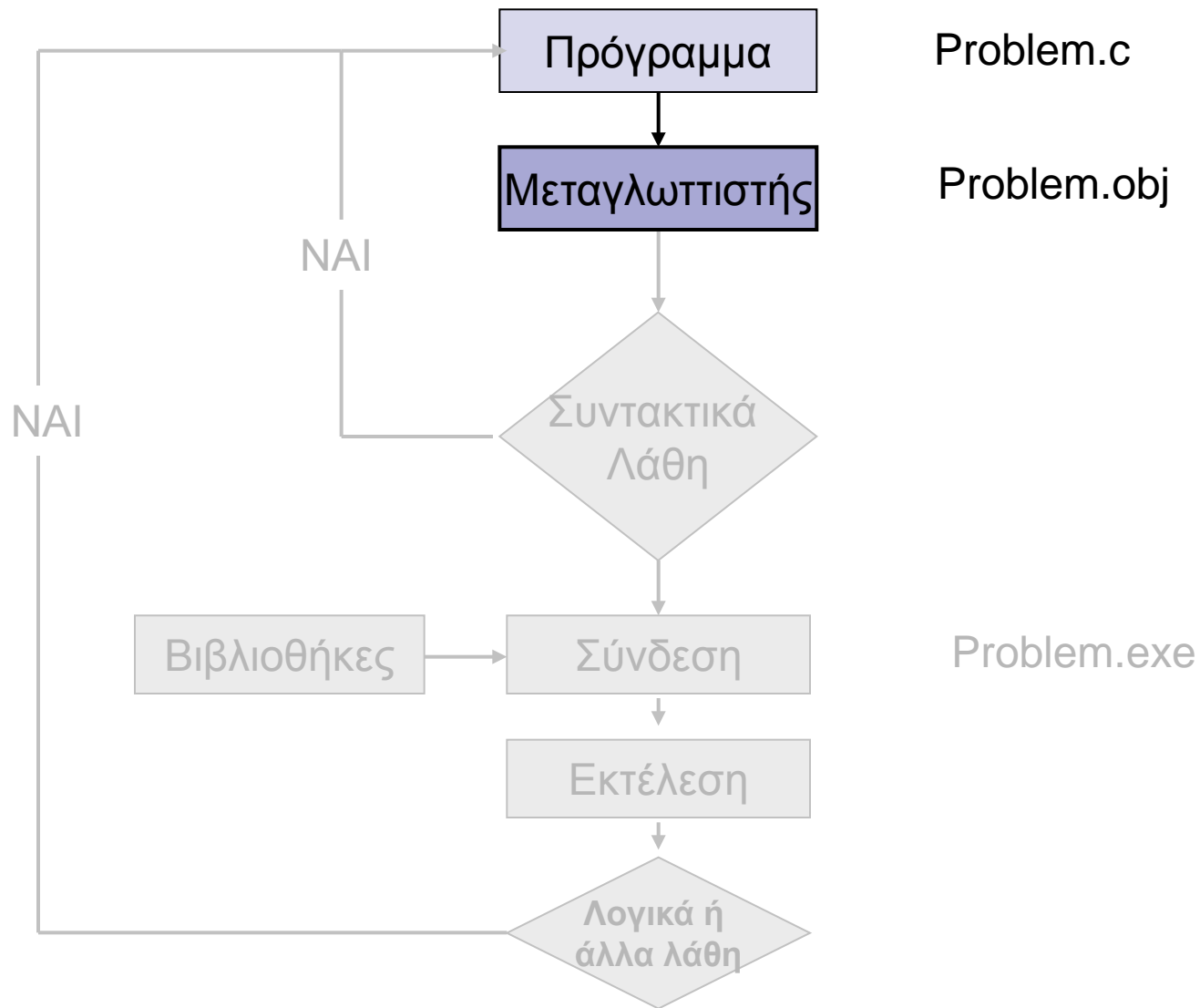
```
    }
```

```
    mo=s/N;                //Ypologismos Mesou Orou
```

```
    printf("MO = %f \n",mo);    //Emfanish apotelesmatos
```

```
}
```

# Κύκλος Υλοποίησης



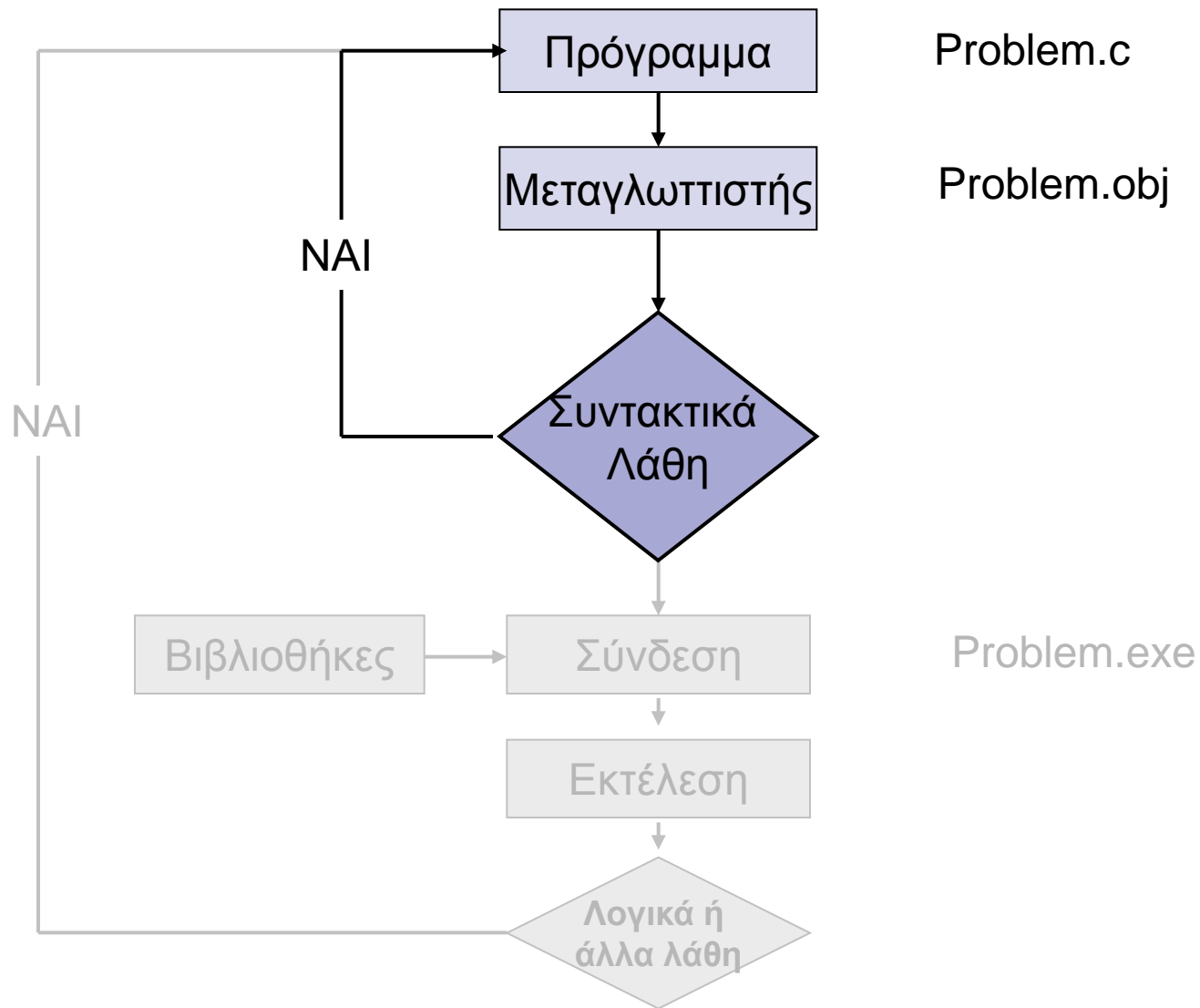
# Μεταγλωττιστής (Compiler)

- Οι υπολογιστές δεν κατανοούν καμία φυσική γλώσσα ή γλώσσα προγραμματισμού υψηλού επιπέδου
- Αντιλαμβάνονται μόνο Εντολές, διευθύνσεις μνήμης, δεδομένα ..... ΟΛΑ σε δυαδική μορφή
- Χρειάζεται μετατροπή του κώδικα που γράφεται σε γλώσσα προγραμματισμού σε μια μορφή που να μπορεί να εκτελεστεί από τον υπολογιστή
- Απαιτείται η χρήση ενός ειδικού προγράμματος, του **μεταγλωττιστή**

# Μεταγλωττιστής (Compiler)

- Είναι ένα πρόγραμμα που δέχεται σαν είσοδο ένα **πρόγραμμα** σε μία γλώσσα προγραμματισμού και παράγει ένα ισοδύναμο πρόγραμμα σε **γλώσσα μηχανής** (εντολές σε δυαδική μορφή τις οποίες μπορεί να εκτελέσει ο υπολογιστής)
- Το αποτέλεσμα της μεταγλώττισης είναι ο αντικειμενικός κώδικας ή κώδικας αντικείμενο (αρχείο με κατάληξη **.obj**)
- Στο στάδιο της μεταγλώττισης γίνεται και η εκσφαλμάτωση
- Το πρόγραμμα που παράγει ο μεταγλωττιστής δεν είναι εκτελέσιμο

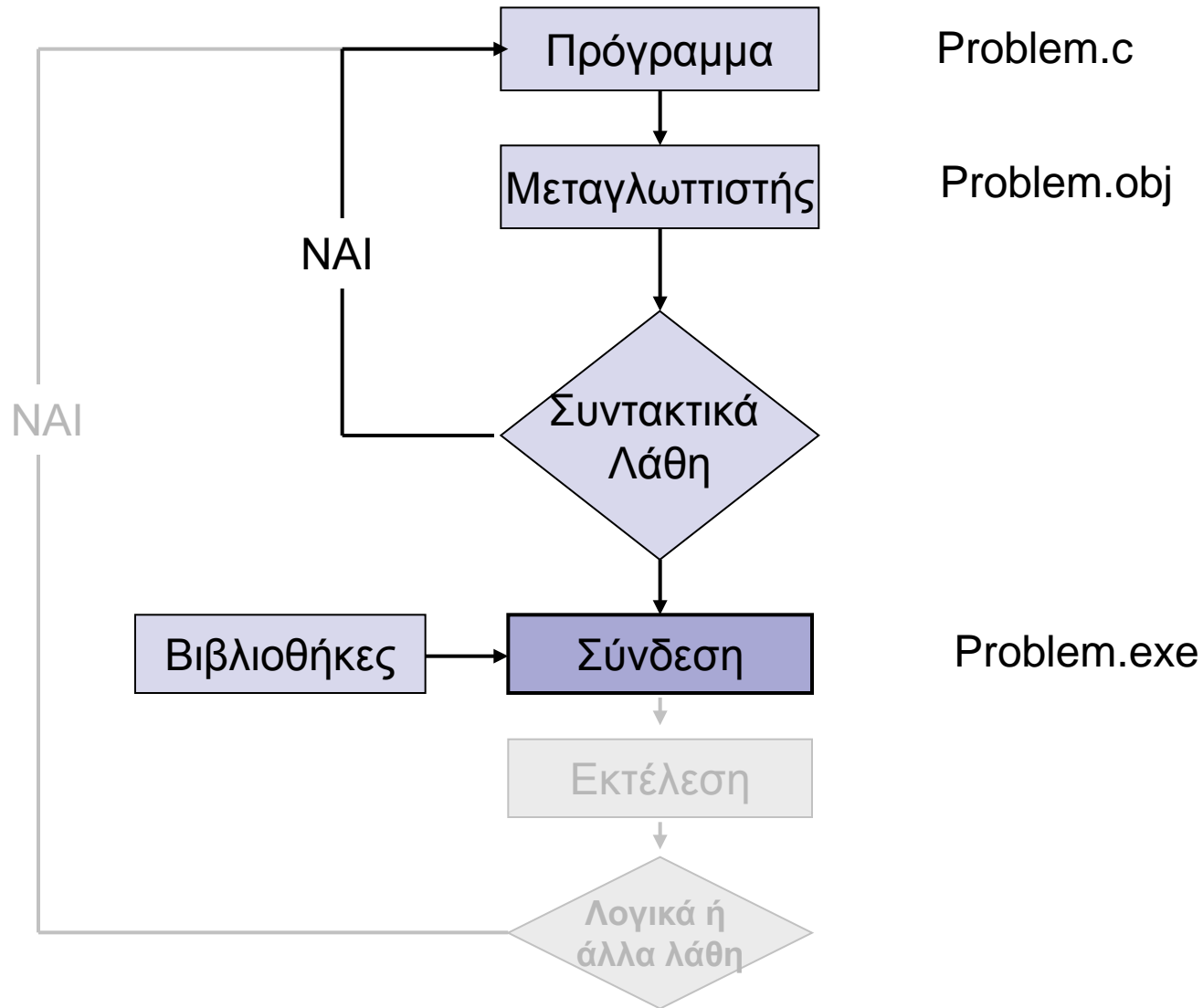
# Κύκλος Υλοποίησης



# Εκσφαλμάτωση

- Τα λάθη που μπορεί να υπάρξουν σε ένα πρόγραμμα είναι
  - **Συντακτικά:** τα λάθη αυτά ανιχνεύονται από το μεταγλωττιστή και εμφανίζεται κατάλληλο διαγνωστικό μήνυμα
  - **Λογικά:** τα λάθη αυτά εμφανίζονται από το προγραμματιστή κατά τη διάρκεια και μετά το τέλος της εκτέλεσης του προγράμματος

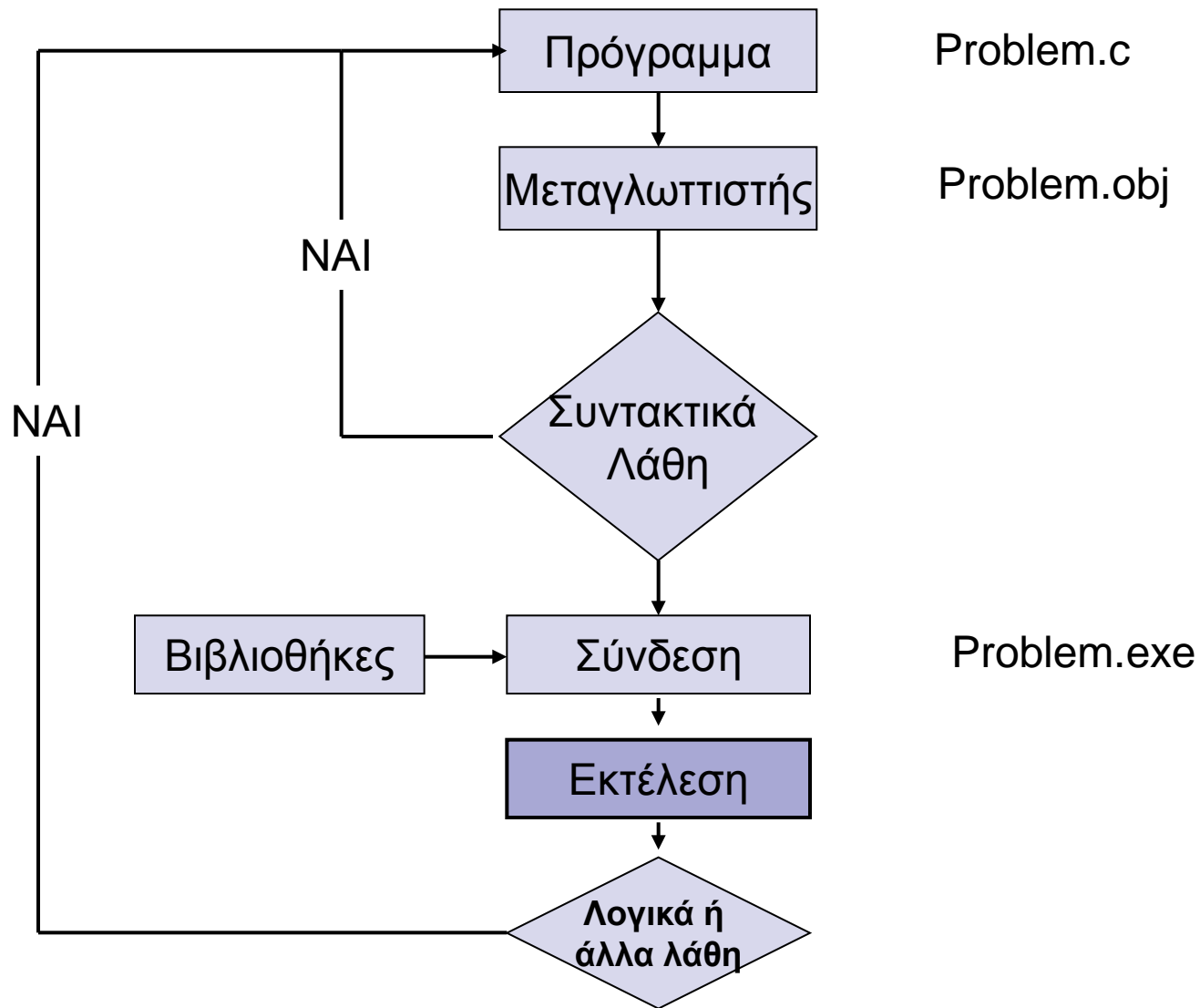
# Κύκλος Υλοποίησης



## Συνδέτης (Linker)

- Είναι ένα ειδικό πρόγραμμα που συνδέει το αντικείμενο πρόγραμμα με άλλα τμήματα προγράμματος που βρίσκονται στις βιβλιοθήκες της γλώσσας ή τα έχει γράψει ο προγραμματιστής
- Το αποτέλεσμα είναι το τελικό εκτελέσιμο πρόγραμμα (executable)

# Κύκλος Υλοποίησης



# Εισαγωγή σε περιβάλλον προγραμματισμού (IDE)

- IDE : Integrated Development Environment  
Code::Blocks

<http://www.codeblocks.org/>



# Νέο Πρόγραμμα

- Δημιουργήστε ένα νέο project με τίτλο

**MesosOros**

**Αποθήκευση στον φάκελο**

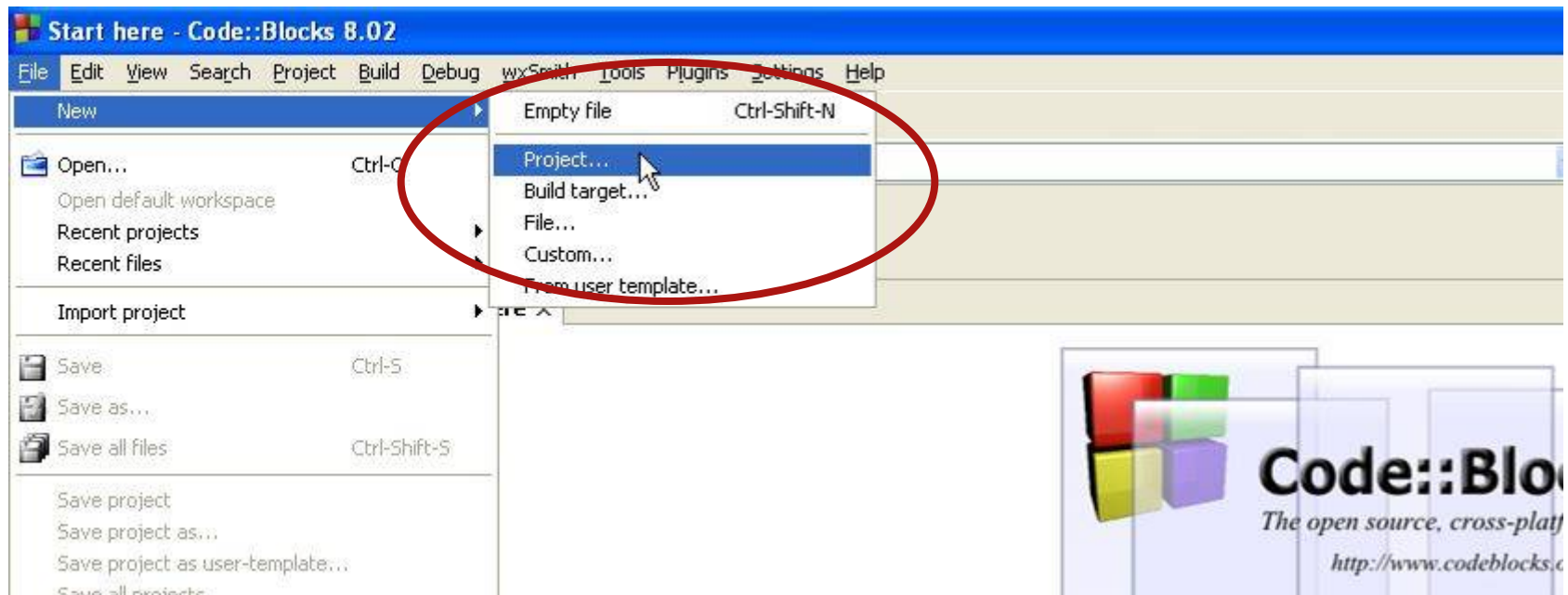
**Domimenos\_AMxxx\05\_10\_2010**

**όπου xxx το Α.Μ. Σας**

**π.χ. Domimenos\_AM50\05\_10\_2010**

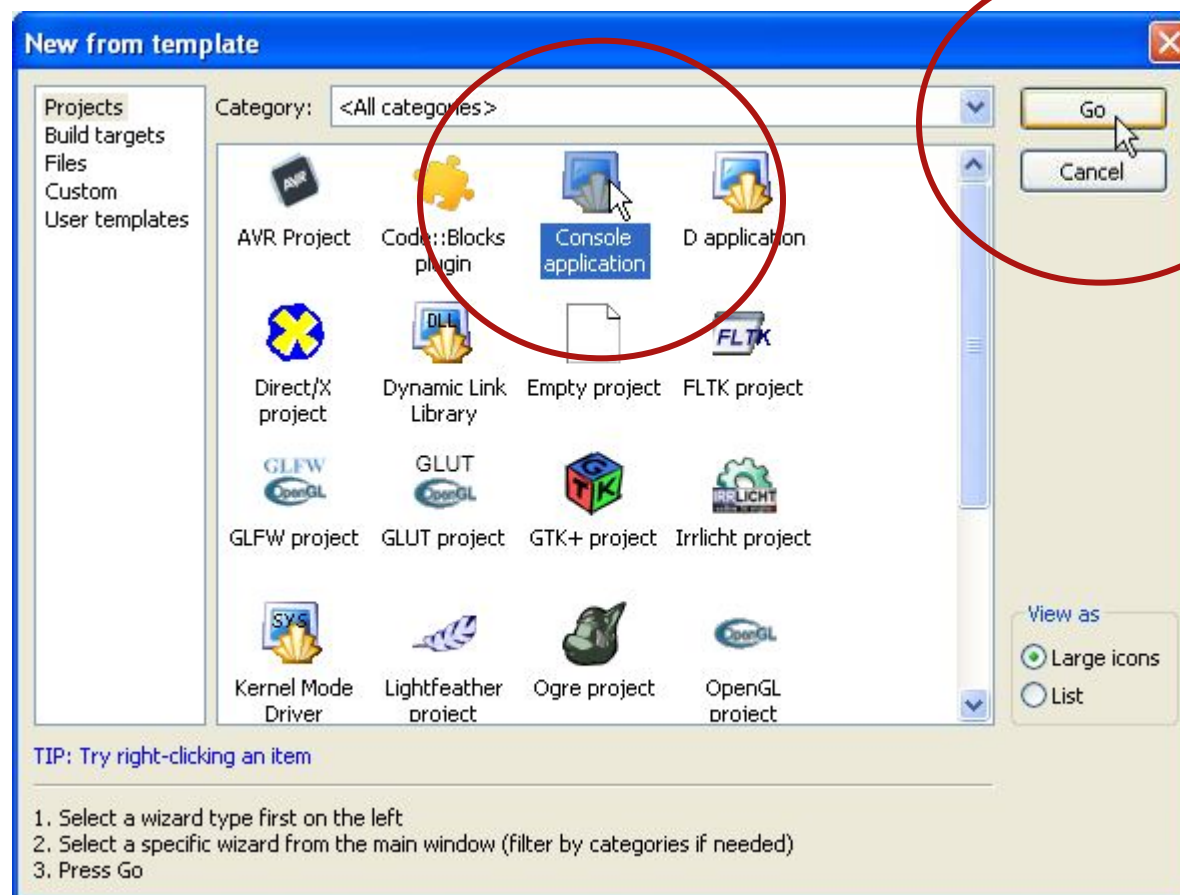
# Code::Blocks

- Δημιουργία ενός νέου προγράμματος (Project)
  - Δημιουργούνται αυτόματα όλα τα αρχεία που σχετίζονται με ένα πρόγραμμα

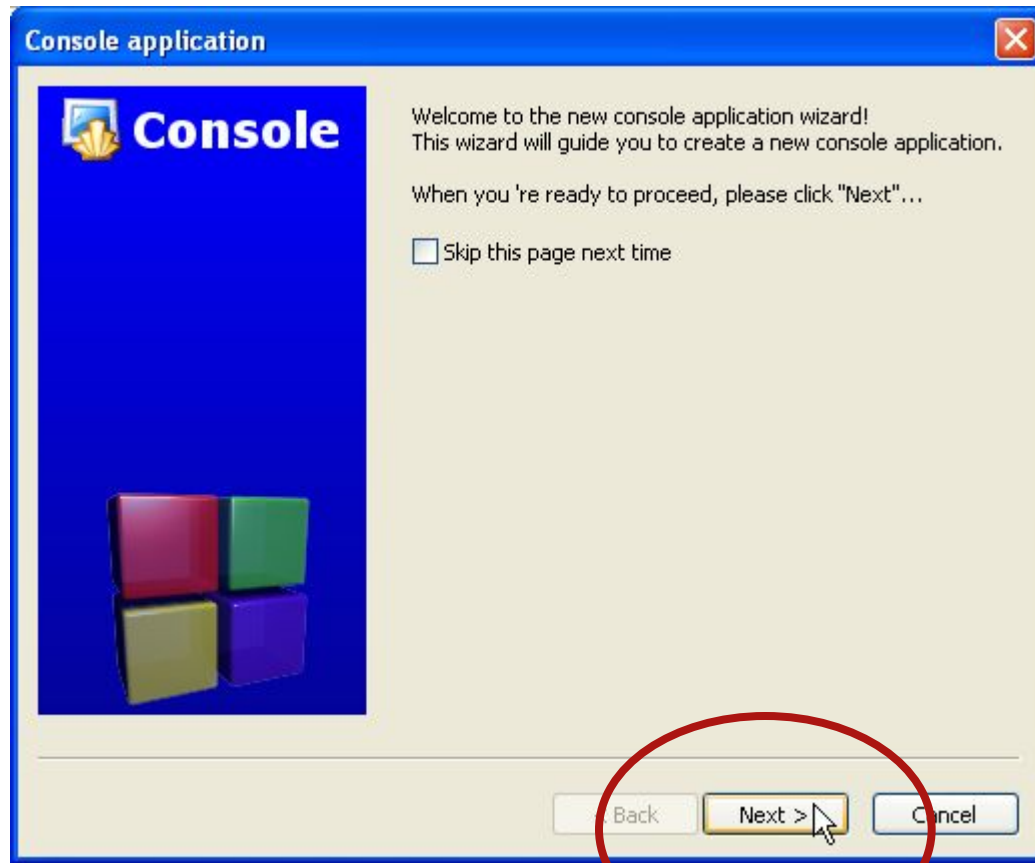


# Code::Blocks

- Επιλογή τύπου εφαρμογής

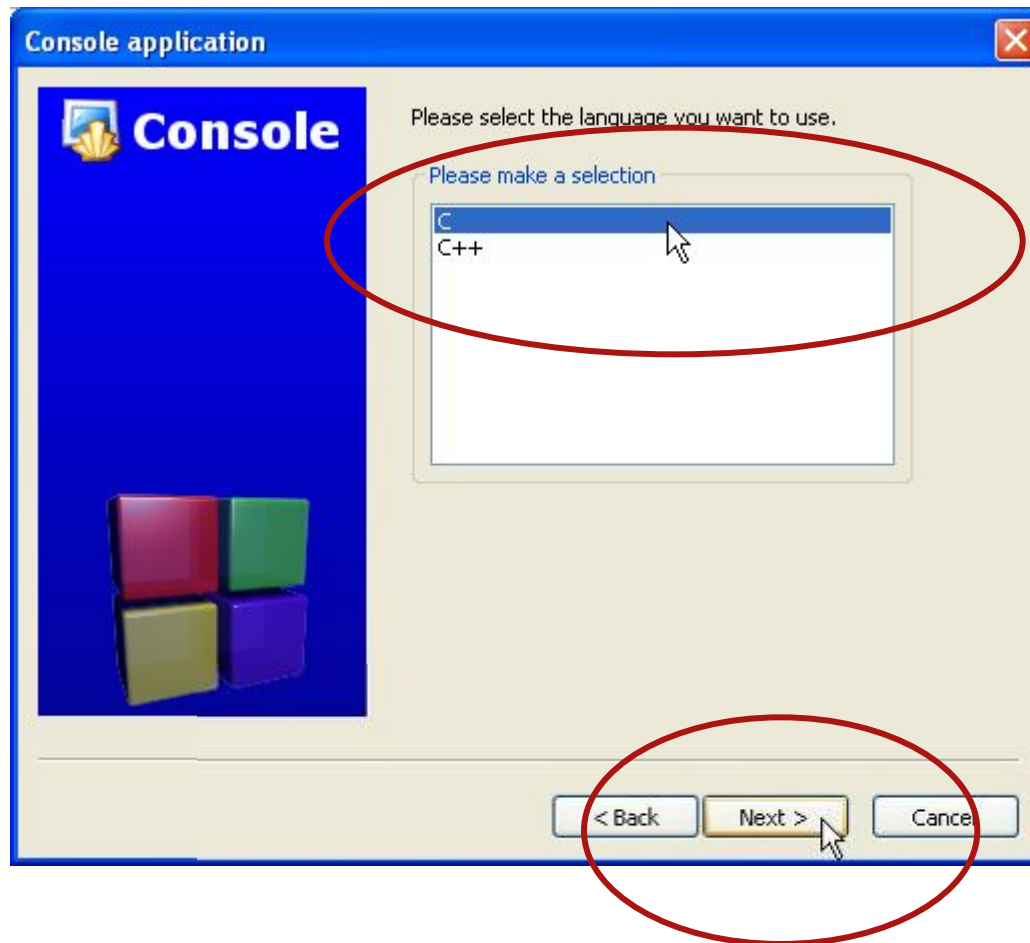


# Code::Blocks



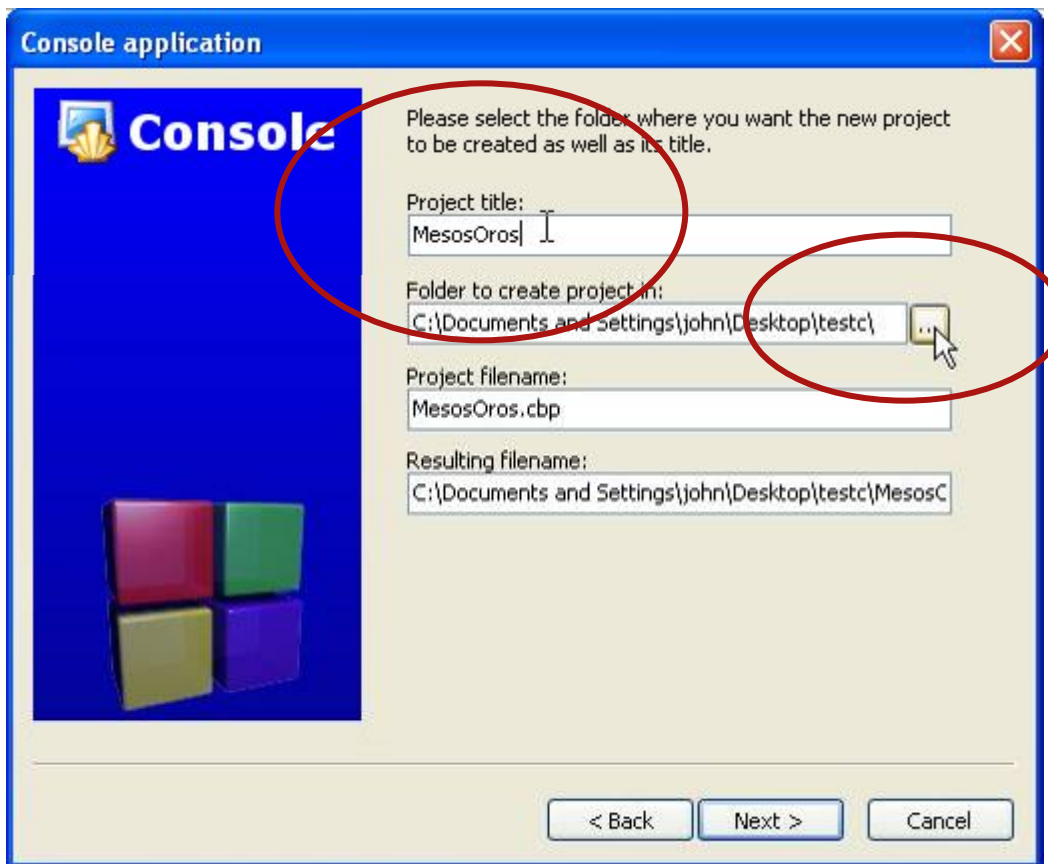
# Code::Blocks

- Επιλογή γλώσσας προγραμματισμού

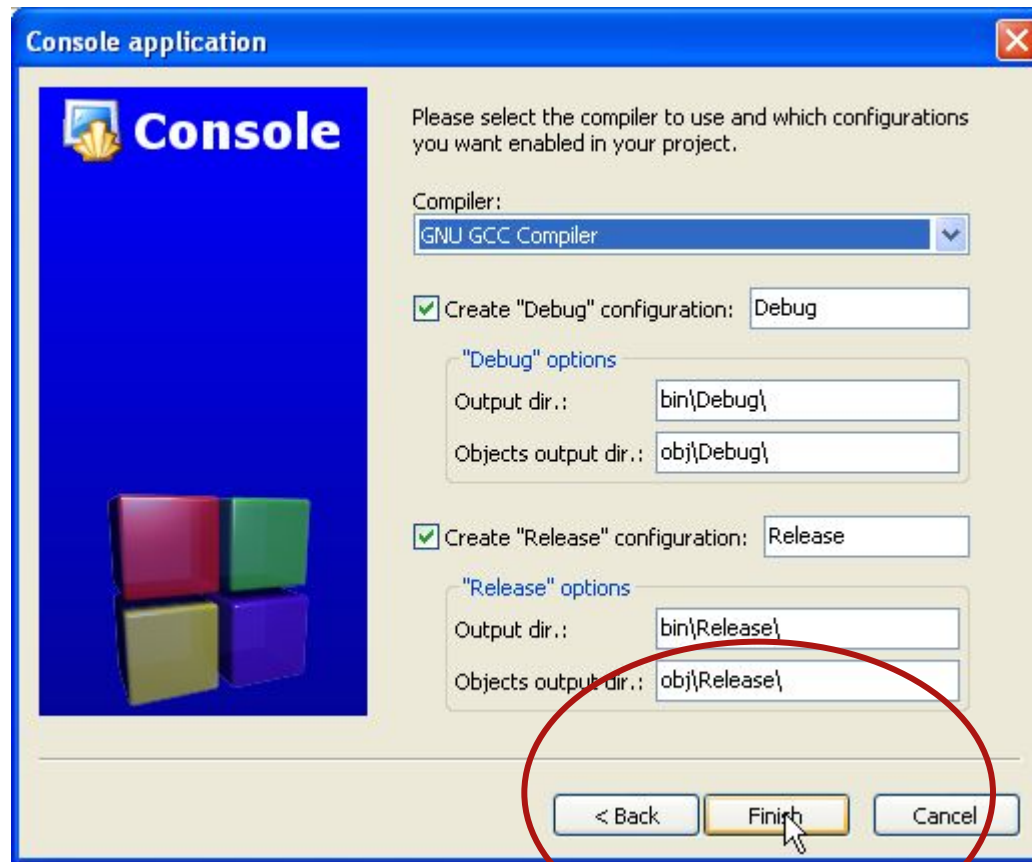


# Code::Blocks

- Δώστε όνομα στο πρόγραμμά σας
- Επιλέξτε φάκελο

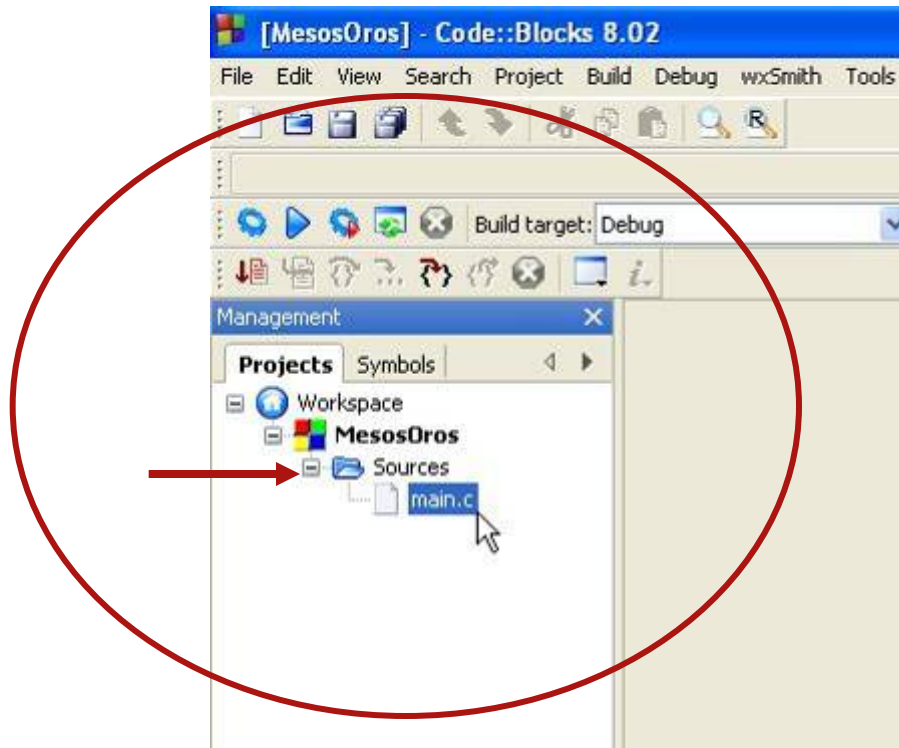


# Code::Blocks



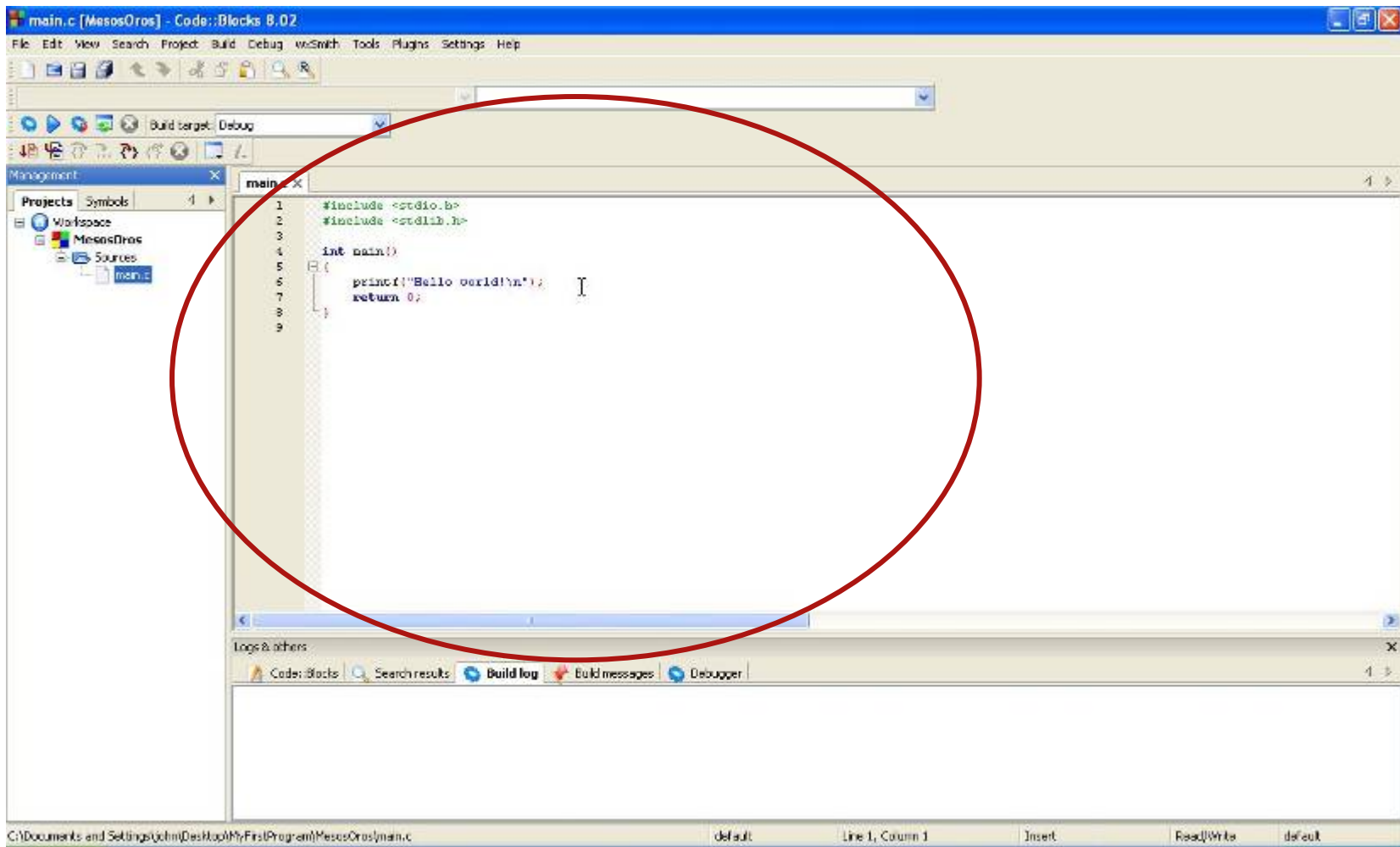
# Code::Blocks

- Επιλέξτε το αρχείο main.c με διπλό κλικ



# Code::Blocks

- Παράδειγμα προγράμματος



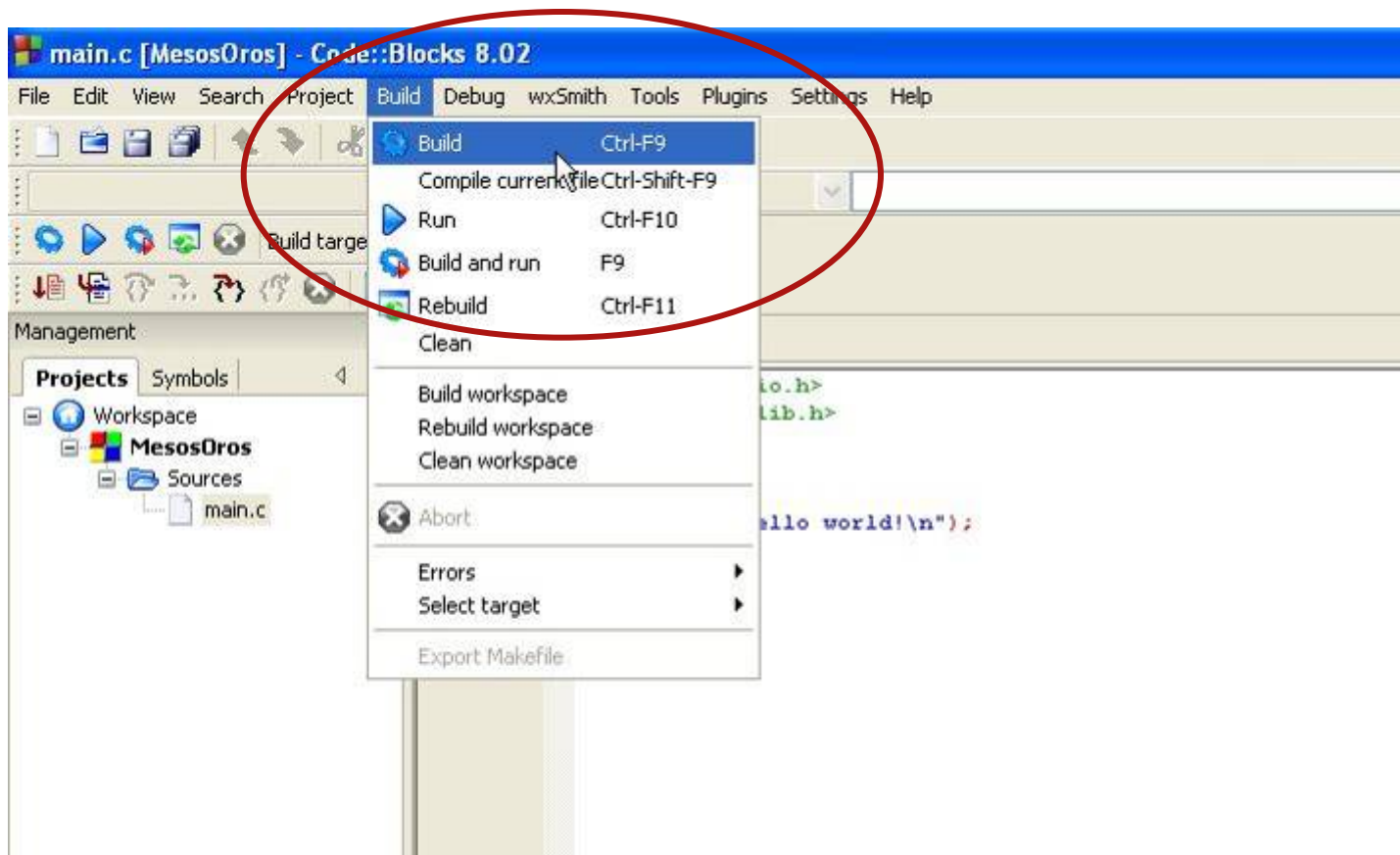
The screenshot shows the Code::Blocks IDE interface. The main editor window displays the following C code:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     printf("Hello world!\n");
7     return 0;
8 }
9
```

The code is highlighted with a red oval. The IDE interface includes a menu bar (File, Edit, View, Search, Project, Build, Debug, Window, Tools, Plugins, Settings, Help), a toolbar, a Management panel on the left with a Project Explorer showing 'Workspace', 'MesosOros', 'Sources', and 'main.c', and a status bar at the bottom indicating the current file path and cursor position.

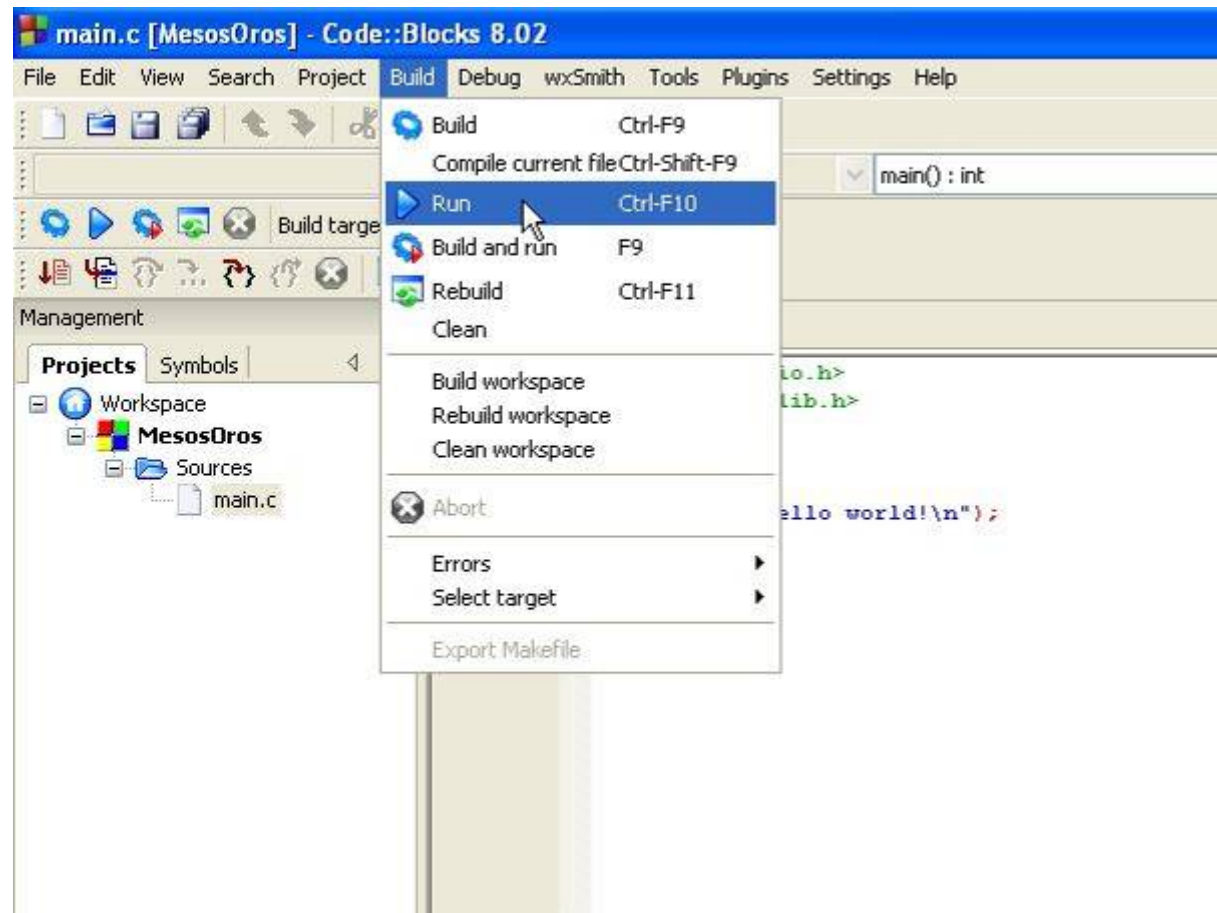
# Code::Blocks

- Επιλέξτε Build να παραχθεί το εκτελέσιμο

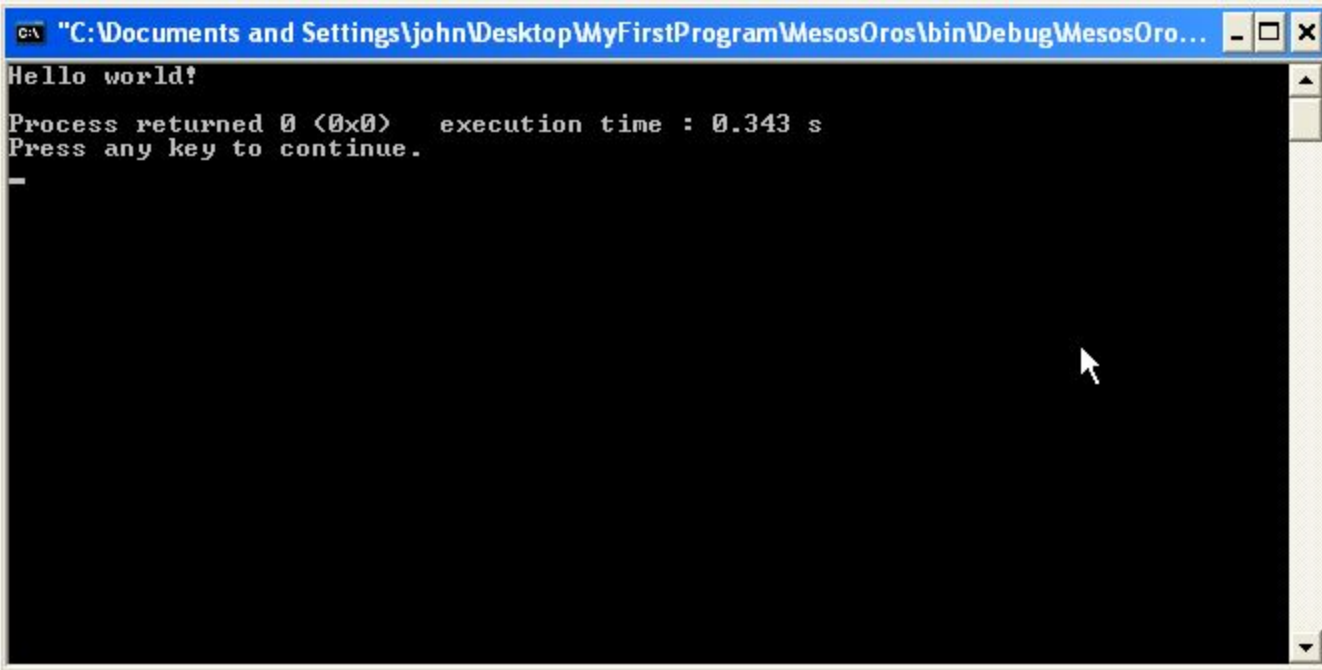


# Code::Blocks

- Επιλέξτε Run ώστε να εκτελεστεί το πρόγραμμά σας



# Code::Blocks



```
g++ "C:\Documents and Settings\john\Desktop\MyFirstProgram\MesosOros\bin\Debug\MesosOro...  
Hello world!  
Process returned 0 (0x0)   execution time : 0.343 s  
Press any key to continue.  
-
```

# Code::Blocks

- Γράψτε το παρακάτω πρόγραμμα μεταγλωττίστε το και τρέξτε το εκτελέσιμο

```
#include <stdlib.h>
#include <stdio.h>

main{
{
    float s = 0, a, mo;
    int i, N = 6;
    for ( i=1; i<=N; i++ )
    {
        printf( " Type a number: " );
        scanf( " %f " , &a );
        s += a;
    }
    mo = s / N;
    printf( " MO = %f " , mo );
}
```