



**Τμήμα Μηχανικών Η/Υ & Πληροφορικής
Πανεπιστήμιο Πατρών**

Εισαγωγή στον Προγραμματισμό

Η γλώσσα προγραμματισμού C

- Καθηγητής Κλεάνθης Θραμπουλίδης
- Καθηγητής Σπύρος Σιούτας
- Αναπληρωτής καθηγητής Χρήστος Μακρής
- Ε.ΔΙ.Π. Άρης Ηλίας

- Το e-class του μαθήματος μάθημα–
<https://eclass.upatras.gr/courses/CEID1416/>

ΕΚΔΟΣΕΙΣ ΚΛΕΙΔΑΡΙΘΜΟΣ

ΔΕΥΤΕΡΗ ΒΕΛΤΙΩΜΕΝΗ ΕΚΔΟΣΗ

Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ



BRIAN W. KERNIGHAN
DENNIS M. RITCHIE



Κωδικός Βιβλίου στον Εύδοξο: 13956

Συγγραφείς: BRIAN W. KERNIGHAN, DENNIS M. RITCHIE

Αριθμός Έκδοσης 2η

Έτος Τρέχ. Έκδοσης 2008

Λέξεις κλειδιά C, ΑΡΧΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ, ΔΟΜΗΜΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ, ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ, ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ Η/Υ, ΠΛΗΡΟΦΟΡΙΚΗ, ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ, ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΕ C, ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΕΣ ΤΕΧΝΙΚΕΣ

Θεματικές Ενότητες

ISBN 978-960-461-132-4

Εκδόσεις ΕΚΔΟΣΕΙΣ ΚΛΕΙΔΑΡΙΘΜΟΣ ΕΠΕ

Δέσιμο Μαλακό Εξώφυλλο

Διαστάσεις [17 x 24]

ΔΙΑΔΙΚΑΣΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

2^η Έκδοση

C

ΑΠΟ ΤΗ C ΣΤΗ JAVA



ΕΚΔΟΣΕΙΣ ΤΖΙΟΛΑ



ΑΠΟ ΤΗ C ΣΤΗ JAVA

Από το διαδικαστικό στον Αντικειμενοστρεφή
προγραμματισμό

Η Java, η γλώσσα που θα σηματοδοτήσει την αρχή της νέας χιλιετίας, υιοθετεί το μεγαλύτερο τμήμα της C και προσθέτει νέες κατασκευές, κύρια για την υποστήριξη της Αντικειμενοστρεφούς μορφής προγραμματισμού. Το έργο, βασισμένο σε πολυετή εμπειρία από διδασκαλία και ανάπτυξη πραγματικών συστημάτων αποτελείται από δύο τόμους. Ο Α τόμος εισάγει τον αναγνώστη με απλό τρόπο στις βασικές έννοιες του διαδικαστικού προστακτικού προγραμματισμού (imperative procedural programming) και τον εξοικειώνει με τη διαδικασία ανάπτυξης συστημάτων λογισμικού. Το βιβλίο δεν προϋποθέτει προηγούμενη γνώση άλλης γλώσσας προγραμματισμού.

Διαδικαστικός Προγραμματισμός - C

- ▶ εισάγει τον αναγνώστη με απλό τρόπο στις βασικές έννοιες του διαδικαστικού προστακτικού προγραμματισμού (imperative procedural programming)
- ▶ εξοικειώνει τον αναγνώστη με την διαδικασία ανάπτυξης συστημάτων λογισμικού.

Το βιβλίο δεν προϋποθέτει προηγούμενη γνώση άλλης γλώσσας προγραμματισμού.

Αντικειμενοστρεφής Προγραμματισμός - Java

- ▶ εισάγει με ένα ξεχωριστό τρόπο, χρησιμοποιώντας έμφως της καθημερινής ζωής, τις βασικές έννοιες της τεχνολογίας Αντικειμένων (Object Technology) και τον τρόπο που είναι που γλώσσα προστακτικού προγραμματισμού της αποστηρίξει.
- ▶ εξοικειώνει τον αναγνώστη με την διαδικασία ανάπτυξης αντικειμενοστρεφών προγραμμάτων. Είκοσι οκτώ από εργαστηριακές ασκήσεις, με αναλυτικές οδηγίες και παραδείγματα πηγαίου κώδικα, οδηγούν τον αναγνώστη με ένα πρωτότυπο τρόπο, βήμα-βήμα στην ανάπτυξη της σύστασης εφαρμογής.

Ο Κλεάνθης Θραμπουλίδης έχει δουλέψει ως μηχανικός λογισμικού στην ανάπτυξη πραγματικών συστημάτων για 20 περίπου χρόνια. Διδάσκει Ανάπτυξη Σχεδιασμό και Υλοποίηση Συστημάτων Λογισμικού στο Τμήμα Ηλεκτρολόγων Μηχανικών και Ηλεκτρονικής Τεχνολογίας του Πανεπιστημίου Πειραιώς. Είναι ορατός συνιδρυτής με στέλεχος εταιρείων ανάπτυξης συστημάτων λογισμικού. Το τελευταίο βιβλίο του χρησιμοποιεί με μεγάλη επιτυχία την τεχνολογία Αντικειμένων σε Ελλάδα και Ευρώπη και εμπνευστής προγράμματος (SPRINT, ESO, STRIDE, IST...). Σχεδίασε την Αντικειμενοστρεφή Γλώσσα PROLOG για την διερεύνηση κινήσεων σε προηγμένα χρονοπρογραμματισμένα δίκτυα αεροπλανικών εγχειρίστων.

Κωδικός Βιβλίου στον Εύδοξο: 18549064

Συγγραφείς: Θραμπουλίδης Κλεάνθης Χ.

Η Java, η γλώσσα που θα σηματοδοτήσει την αρχή της νέας χιλιετίας, υιοθετεί το μεγαλύτερο τμήμα της C και προσθέτει νέες κατασκευές, κύρια για την υποστήριξη της Αντικειμενοστρεφούς μορφής προγραμματισμού. Το έργο βασισμένο σε πολυετή εμπειρία από διδασκαλία και ανάπτυξη πραγματικών συστημάτων αποτελείται από δύο τόμους. Ο Α τόμος εισάγει τον αναγνώστη με απλό τρόπο στις βασικές έννοιες του διαδικαστικού προστακτικού προγραμματισμού (imperative procedural programming) και τον εξοικειώνει με τη διαδικασία ανάπτυξης συστημάτων λογισμικού. Το βιβλίο δεν προϋποθέτει προηγούμενη γνώση άλλης γλώσσας προγραμματισμού.

Αριθμός Έκδοσης 2η έκδ.

Έτος Τρέχ. Έκδοσης 2003

Λέξεις κλειδιά C, Strings, Διαχείριση, Δικτυακοί Τύποι, Δομές, Δυαδικές Δομές Δέντρου, Εκδόσεις Τζιόλα, Θραμπουλίδης, Πίνακες, Προγραμματισμός C, Συναρτήσεις, χεδιαση Προγραμμάτων

Θεματικές Ενότητες

ISBN 9789608050792

Εκδόσεις ΤΖΙΟΛΑ

Δέσιμο Σκληρό Εξώφυλλο

Διαστάσεις [17 x 24]

Αριθμός 512

Σελίδων

Διαθέτης ΕΚΔΟΣΕΙΣ Α. ΤΖΙΟΛΑ & ΥΙΟΙ Α.Ε.

(Εκδότης)

Τύπος Σύγγραμμα

Ιστοσελίδα [Εδώ](#)

Βιβλίου

Κωδικός Βιβλίου στον Εύδοξο: 122079784

Συγγραφείς: Γ. Σ. Τσελίκης - Ν. Δ. Τσελίκας

Αριθμός Έκδοσης	4
Έτος Τρέχ. Έκδοσης	2023
Λέξεις κλειδιά	Γλώσσα C, Γλώσσες Προγραμματισμού
Θεματικές Ενότητες	
ISBN	9786188676206
Εκδόσεις	Γ.Σ.Τσελίκης - Ν.Δ.Τσελίκας
Δέσιμο	Μαλακό Εξώφυλλο
Διαστάσεις	17x24
Αριθμός Σελίδων	888
Διαθέτης (Εκδότης)	ΓΕΩΡΓΙΟΣ ΤΣΕΛΙΚΗΣ
Τύπος	Σύγγραμμα
Ιστοσελίδα Βιβλίου	Εδώ

Δρ. Γ. Σ. Τσελίκης | Δρ. Ν. Δ. Τσελίκας

C Από τη θεωρία στην Εφαρμογή
Δ' Έκδοση

500+ ώρες από το κ. κ. τη μάθη μελέτη, παρτίκι διακρίνει
Πραγματικό από το κλάσμα, κλάσμα, κλάσμα, κλάσμα
Από το κλάσμα, κλάσμα, κλάσμα, κλάσμα, κλάσμα
Πραγματικό από το κλάσμα, κλάσμα, κλάσμα, κλάσμα
| Επεξεργασία C++ - Java

C Από τη θεωρία στην Εφαρμογή

ΠΡΟΛΟΓΟΣ
1. ΕΙΣΑΓΩΓΗ
2. ΟΡΓΑΝΙΣΜΟΣ ΤΟΥ ΚΩΔΙΚΑ
3. ΟΡΓΑΝΙΣΜΟΣ ΤΟΥ ΚΩΔΙΚΑ
4. ΟΡΓΑΝΙΣΜΟΣ ΤΟΥ ΚΩΔΙΚΑ
5. ΟΡΓΑΝΙΣΜΟΣ ΤΟΥ ΚΩΔΙΚΑ
6. ΟΡΓΑΝΙΣΜΟΣ ΤΟΥ ΚΩΔΙΚΑ
7. ΟΡΓΑΝΙΣΜΟΣ ΤΟΥ ΚΩΔΙΚΑ
8. ΟΡΓΑΝΙΣΜΟΣ ΤΟΥ ΚΩΔΙΚΑ
9. ΟΡΓΑΝΙΣΜΟΣ ΤΟΥ ΚΩΔΙΚΑ
10. ΟΡΓΑΝΙΣΜΟΣ ΤΟΥ ΚΩΔΙΚΑ

C Από τη θεωρία στην Εφαρμογή

C Από τη θεωρία στην Εφαρμογή

Κωδικός Βιβλίου στον Εύδοξο: 122074103

Συγγραφείς: Paul Deitel, Harvey Deitel

Αριθμός Έκδοσης 9η Έκδοση

Έτος Τρέχ. Έκδοσης 2023

Λέξεις κλειδιά C, C Προγραμματισμός, C Προγραμματισμός, 9η Έκδοση, C++, Αλγόριθμοι, Δομημένος Προγραμματισμός, Πίνακες, Προγραμματισμός, Προεπεξεργαστής, Συναρτήσεις

Θεματικές Ενότητες

ISBN 9789605127541

Εκδόσεις Χ. ΓΚΙΟΥΡΔΑ ΣΙΑ ΕΕ

Δέσιμο Μαλακό Εξώφυλλο

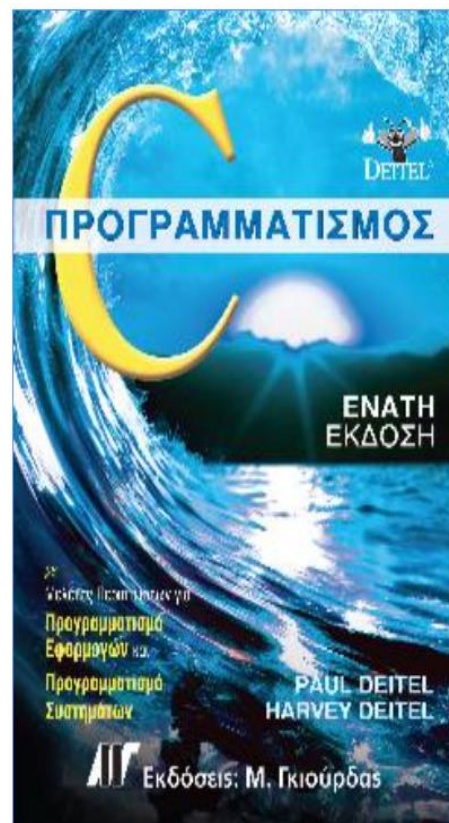
Διαστάσεις 17x24

Αριθμός Σελίδων 856

Διαθέτης (Εκδότης) Χ. ΓΚΙΟΥΡΔΑ & ΣΙΑ ΕΕ

Τύπος Σύγγραμμα

Ιστοσελίδα Βιβλίου [Εδώ](#)



ΕΙΣΗΓΗΣΗ ΚΑΙ ΕΠΕΞΗΓΗΣΗ ΤΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗΣ ΓΛΩΣΣΗΣ C ΚΑΙ ΤΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗΣ ΓΛΩΣΣΗΣ C++

Αντιγράψτε τον κωδικό ISBN του βιβλίου που θέλετε να αγοράσετε και να το εισαγάγετε στο πεδίο αναζήτησης της ιστοσελίδας. Ο κωδικός ISBN είναι ο μοναδικός αριθμός που αναγνωρίζει το βιβλίο. Μπορείτε να το βρείτε στην εσωτερική όψη του βιβλίου ή στην εσωτερική όψη της εξώφυλλου. Ο κωδικός ISBN είναι ο μοναδικός αριθμός που αναγνωρίζει το βιβλίο. Μπορείτε να το βρείτε στην εσωτερική όψη του βιβλίου ή στην εσωτερική όψη της εξώφυλλου.

Εάν έχετε οποιαδήποτε ερώτηση σχετικά με το βιβλίο, μπορείτε να επικοινωνήσετε με τον εκδότη Χ. Γκιούρδα & ΣΙΑ ΕΕ. Ο εκδότης είναι ο αποκλειστικός δικαιούχος των δικαιωμάτων πνευματικής ιδιοκτησίας του βιβλίου. Ο εκδότης είναι ο αποκλειστικός δικαιούχος των δικαιωμάτων πνευματικής ιδιοκτησίας του βιβλίου. Ο εκδότης είναι ο αποκλειστικός δικαιούχος των δικαιωμάτων πνευματικής ιδιοκτησίας του βιβλίου. Ο εκδότης είναι ο αποκλειστικός δικαιούχος των δικαιωμάτων πνευματικής ιδιοκτησίας του βιβλίου.

Εάν έχετε οποιαδήποτε ερώτηση σχετικά με το βιβλίο, μπορείτε να επικοινωνήσετε με τον εκδότη Χ. Γκιούρδα & ΣΙΑ ΕΕ. Ο εκδότης είναι ο αποκλειστικός δικαιούχος των δικαιωμάτων πνευματικής ιδιοκτησίας του βιβλίου. Ο εκδότης είναι ο αποκλειστικός δικαιούχος των δικαιωμάτων πνευματικής ιδιοκτησίας του βιβλίου. Ο εκδότης είναι ο αποκλειστικός δικαιούχος των δικαιωμάτων πνευματικής ιδιοκτησίας του βιβλίου. Ο εκδότης είναι ο αποκλειστικός δικαιούχος των δικαιωμάτων πνευματικής ιδιοκτησίας του βιβλίου.

Βιβλιογραφία

- ✓ Herbert Schildt, Ο οδηγός της C, εκδόσεις Γκιούρδας
- ✓ Ν. Χατζηγιαννάκης, Η γλώσσα C σε βάθος, Κλειδάριθμος
- ✓ Robert C. Martin, Clean Code: A Handbook of Agile Software Craftsmanship
1st Edition Pearson, August 2008
- ✓ Brian W. Kernigan, Rob Pike, The UNIX Programming Environment Bell
Laboratories; PRENTICE-HALL; Whitehall Books, 1984
- ✓ W. Stevens, Stephen Rago, Advanced Programming in the UNIX
Environment, 3rd Edition, Addison Wesley, 2008

Ιστορία της C (1)

Η C επινοήθηκε το 1972 από τον Dennis Ritchie στα εργαστήρια Bell. Δημιουργήθηκε για να εξυπηρετήσει το λειτουργικό σύστημα Unix, το οποίο έως τότε ήταν γραμμένο σε assembly. Ο δημιουργός του Unix, Ken Thompson, φίλος και συνεργάτης του Ritchie, είχε δημιουργήσει την πρόγονο της C, τη γλώσσα B.

Κοινή καταγωγή από τη γλώσσα BCPL, η οποία είχε αναπτυχθεί από τον Martin Richards κατά το πέρασμά του από το Τεχνολογικό Ινστιτούτο της Μασσαχουσέτης (MIT) το 1967, στηριζόμενη στη γλώσσα CPL (Cambridge Programming Language).

Και οι τρεις γλώσσες κατασκευάστηκαν στο πλαίσιο του προγράμματος MAC και του απογόνου του Multics, τα οποία στόχευαν στην κατανομή των πόρων των υπολογιστών σε πολλούς χρήστες.

Τα δύο αυτά προγράμματα, στα οποία συνέπραξαν το MIT, η General Electric και τα εργαστήρια Bell, αποτέλεσαν τη θερμοκοιτίδα πολλών προγραμμάτων λογισμικού, που κυριαρχούν από τη δεκαετία του 1960 έως σήμερα.

Γιατί C;

- ✓ Είναι σχετικά μικρή και εύκολη στην εκμάθηση υποστηρίζοντας top down/modular σχεδιασμό και δομημένο προγραμματισμό
- ✓ Υπάρχει μεγάλη εγκατεστημένη βάση εφαρμογών που αναπτύχθηκαν με τη γλώσσα αυτή.
- ✓ Μπορεί να διαχειριστεί: δυαδικά ψηφία (bits), ψηφιολέξεις (bytes), συμβολοσειρές (words), δείκτες (pointers).
- ✓ Μπορεί να χρησιμοποιηθεί για χαμηλού επιπέδου προγραμματισμό επιτρέποντας άμεση πρόσβαση στους πόρους του υπολογιστή. Συνεπώς είναι κατάλληλη γλώσσα για ανάπτυξη προγραμμάτων συστήματος (systems programs) όπως είναι: λειτουργικά συστήματα (operating systems), διερμηνευτές (interpreters), συντάκτες (editors), συμβολομεταφραστές (assemblers), μεταγλωττιστές (compilers), διαχειριστές βάσεων δεδομένων (database managers).

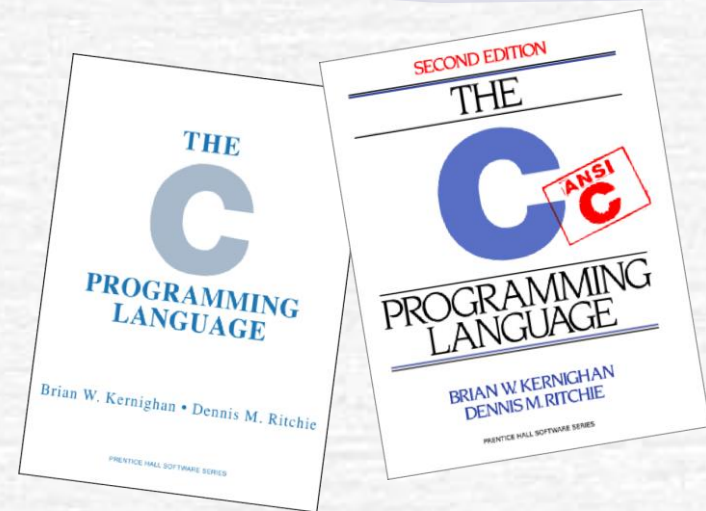
Ιστορία της C (2)

1969-1973: AT&T Bell Labs από Dennis Ritchie ως συνέχεια των BCPL, B (ανάπτυξη νέας έκδοσης UNIX).

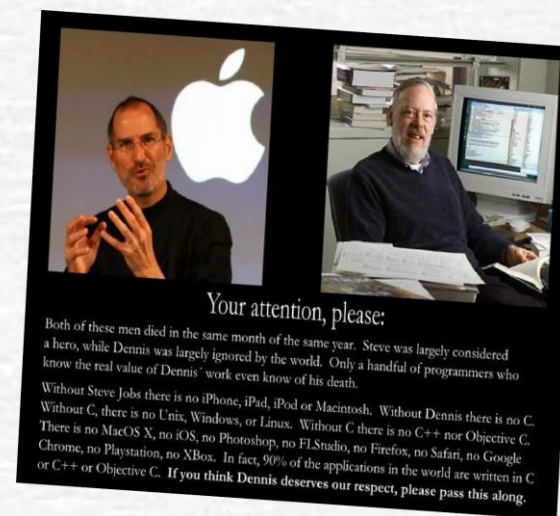
1978: “The C Programming Language” K&R: Kernighan & Ritchie.

1983: Σύσταση ANSI Standardization Committee X3J11.

1989-1990: Αποδοχή ANSI/ISO Standard (ANSI C) C89



- **ISO/IEC 9899:1990** Information Technology – Programming Language C.
- **ISO/IEC 9899 AM1** – αποτελεί τροποποίηση του C πρωτοτύπου. Έγινε αποδεκτό το 1995.
- **ISO/IEC 9899 TCOR1** – Technical Corrigendum 1. Έγινε αποδεκτό το 1995.
- **ISO/IEC 9899 TCOR2** – Technical Corrigendum 2. Έγινε αποδεκτό το 1996.
- **C99** ["ISO/IEC 9899:1999 - Programming languages - C"](https://www.iso.org/standard/54262.html). Iso.org. 2011-12-08. Retrieved 2014-04-08.
- **C11** ["ISO/IEC 9899:2011 - Information technology -- Programming languages -- C"](https://www.iso.org/standard/62618.html). www.iso.org.
- **C18 (or C17)** ["ISO/IEC 9899:2018 - Information technology -- Programming languages -- C"](https://www.iso.org/standard/70490.html). www.iso.org.
- **ΕΠΕΤΑΙ C23 (2024)**



Anjuta
AppCode
C++Builder
Code::Blocks
CodeLite
Dev-C++ GPL
Eclipse CDT
Geany GPL
GNAT Programming
JetBrains CLion
KDevelop
LabWindows/CVI
Microsoft Visual Studio
Microsoft Visual Studio Code
MonoDevelop
NetBeans C/C++ pack
OpenWatcom
Oracle Solaris Studio
Qt Creator
Rational Software Architect (Eclipse IBM)
SlickEdit
Ultimate++ TheIDE
Understand
Xcode (Apple)

Compilers supporting ANSI C

https://en.wikipedia.org/wiki/ANSI_C

- [Amsterdam Compiler Kit](#) (C K&R and C89/90)
- [ARM RealView](#)
- [Clang](#), using [LLVM](#) backend
- [GCC](#) (full C89/90, C99 and C11)
- HP C/ANSI C compiler (C89 and C99)
- [IBM XL C/C++](#) (C11, starting with version 12.1) [\[33\]](#)
- [Intel's ICC](#)
- [LabWindows/CVI](#)
- [LCC](#)
- [OpenWatcom](#) (C89/90 and some C99)
- [Microsoft Visual C++](#) (C89/90 and some C99)
- [Pelles C](#) (C99 and C11. Windows only.)
- [vbcc](#) (C89/90 and C99)
- [Tiny C Compiler](#) (C89/90 and some C99)
- [Oracle Developer Studio](#)

Γλώσσες Προγραμματισμού

Αλγόριθμος: Διαδικασία επίλυσης προβλήματος.

Γλώσσα Προγραμματισμού: Γλώσσα διατύπωσης αλγορίθμων.

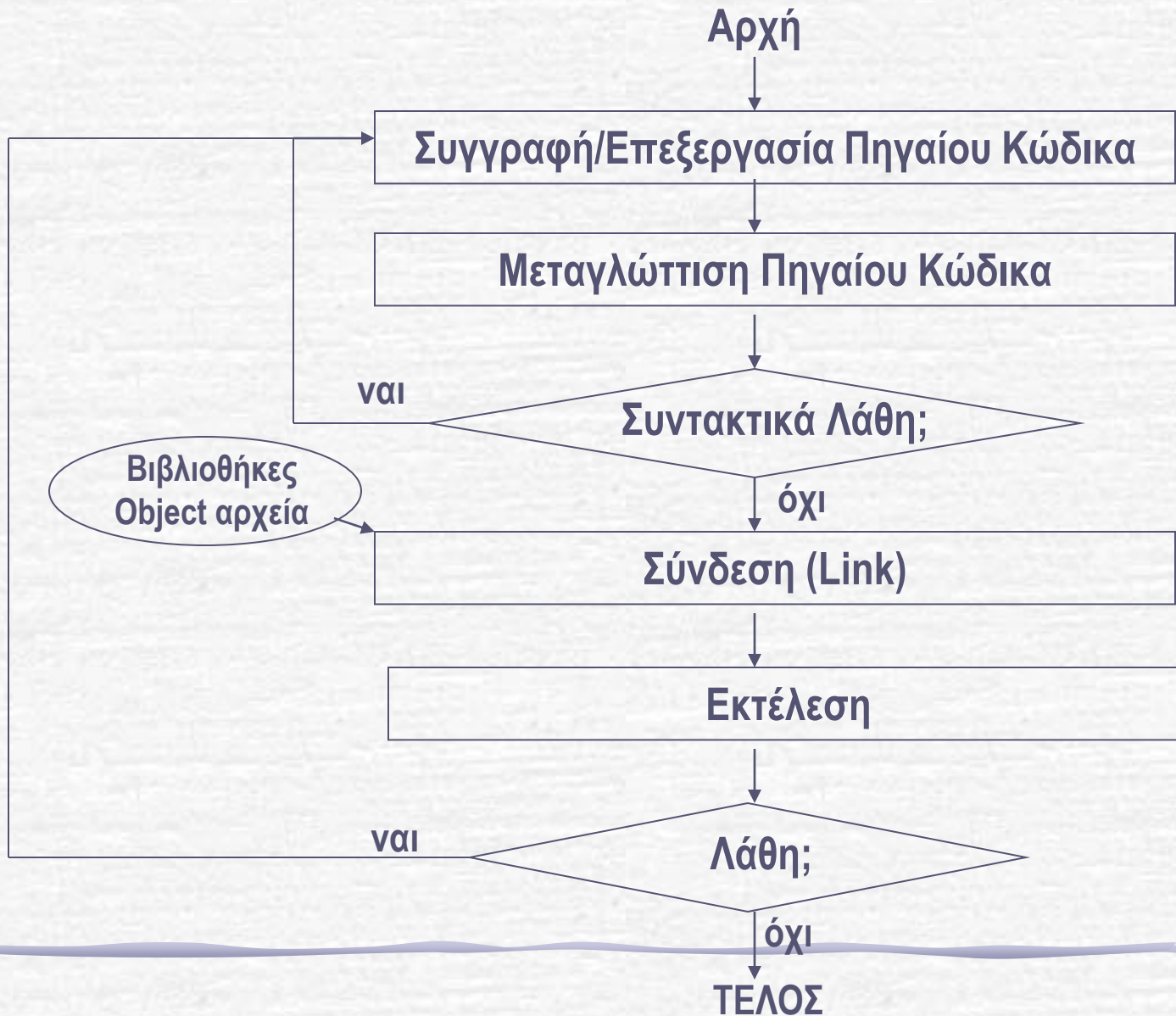
Πρόγραμμα: Αλγόριθμος διατυπωμένος σε κάποια γλώσσα προγραμματισμού.

**Γλώσσα
Προγραμματισμού**

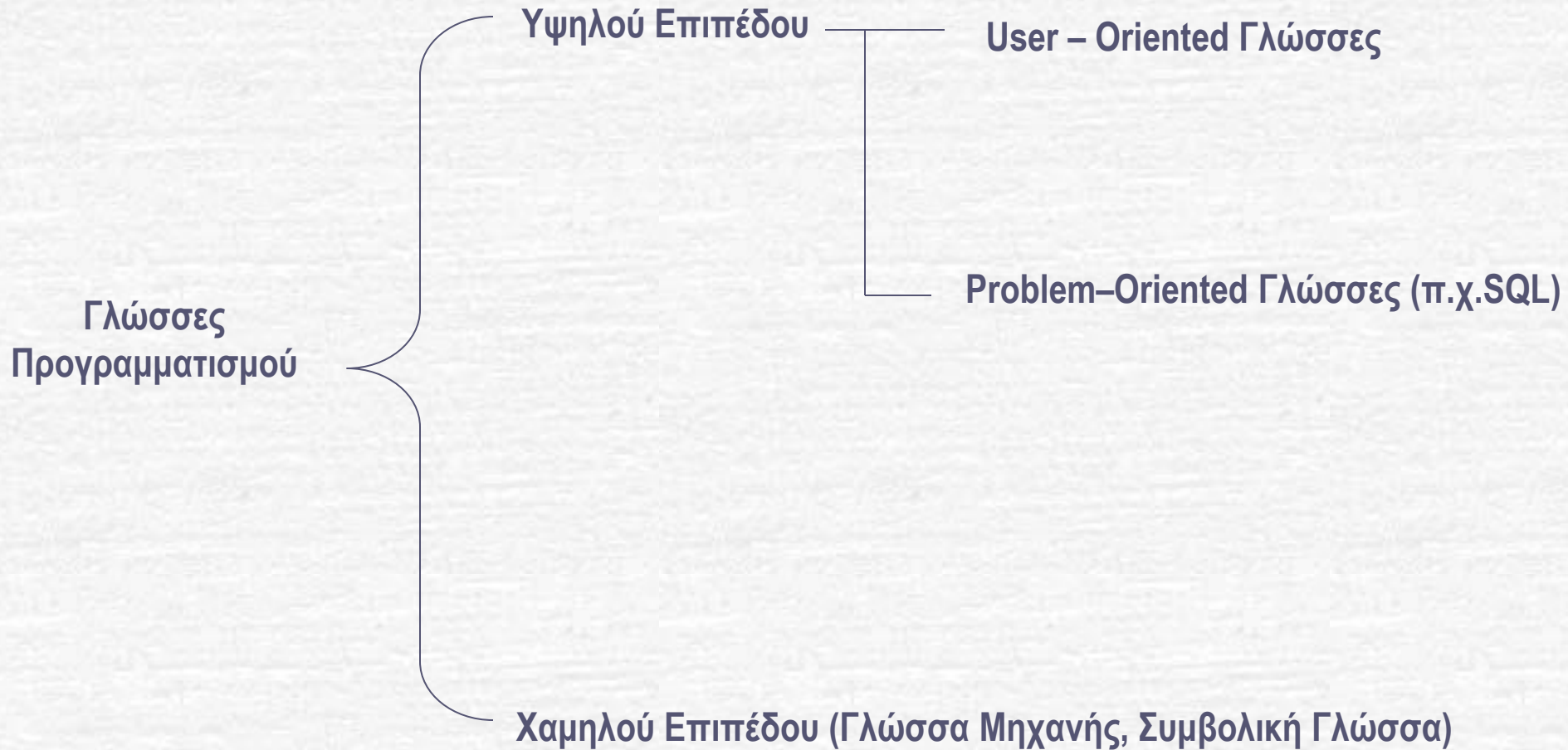
- Λεξιλόγιο (σύνολο εντολών)
- Συντακτικό (κανόνες σύνταξης προτάσεων)
- Σημασιολογία (μοντέλο υπολογισμού)

Γραμματικές δομής φράσεως, γραμματικές ελεύθερες από συμφραζόμενα
(context free grammars)

Φάση Υλοποίησης



Είδη Γλωσσών Προγραμματισμού (1)



Μοντέλο Von Neumann (1950)

Ο υπολογιστής αποτελείται από:

1. Μνήμη
2. Αριθμητική και Λογική Μονάδα
3. Μονάδα Ελέγχου
4. Είσοδος / Έξοδος

Ένα πρόγραμμα αποτελείται από ένα πεπερασμένο αριθμό εντολών, το πρόγραμμα πρέπει να αποθηκεύεται στη μνήμη και οι εντολές εκτελούνται η μία μετά την άλλη, σειριακά.

Είδη Γλωσσών Προγραμματισμού (2)

User Oriented Γλώσσες Προγραμματισμού

✓ Επιτακτικές ή Διαδικαστικές ή Προστακτικές Γλώσσες

π.χ. FORTRAN, ALGOL, COBOL, BASIC, PL/1, C, ADA

- Αντικειμενοστραφείς (C++, Java)
- Οπτικές (Visual Basic)

✓ Εφαρμοστικές ή Δηλωτικές ή Περιγραφικές Γλώσσες

- Συναρτησιακές (Lisp, Haskell)
- Λογικές (Prolog)

Πρακτικές Προγραμματισμού

- ❑ Κατά βήμα εκλέπτυνση (παραγωγή διαδοχικών «εκδόσεων» προγράμματος, κάθε έκδοση είναι ορθή και αυτάρκης, προκύπτει άμεσα από την προηγούμενη και περιγράφει το πρόγραμμα σε μεγαλύτερο βαθμό λεπτομέρειας)
- ❑ Αρθρωτός προγραμματισμός (διάσπαση προβλήματος σε μικρότερα και απλούστερα)

Κατά Βήμα Εκλέπτυνση

- ❑ Διάσπαση σε στοιχειώδη τμήματα
- ❑ Απομόνωση τμημάτων που στην πραγματικότητα δεν εξαρτώνται από άλλα
- ❑ Αναβολή των αποφάσεων αναπαράστασης
- ❑ Προσεκτική απόδειξη ότι κάθε διαδοχικό βήμα εκλέπτυνσης είναι πιστή επέκταση προηγούμενων βημάτων
- ❑ Βελτίωση προκύπτοντος αλγορίθμου

Αρθρωτός Προγραμματισμός

Τμήμα λογισμικού:

- Περιέχει εντολές και δομές δεδομένων
- Μπορεί να μεταγλωττιστεί ανεξάρτητα και να αποθηκευτεί σε μία βιβλιοθήκη
- Μπορεί να συμπεριλαμβάνεται σε ένα πρόγραμμα το οποίο θα χρησιμοποιεί τις λειτουργίες του τμήματος
- Μπορεί να χρησιμοποιεί άλλα τμήματα

Τεχνοτροπίες Προγραμματισμού

- ❑ Προγραμματισμός για επαναχρησιμοποίηση
- ❑ Προγραμματισμός με πλεονασμό (για ανοχή σε σφάλματα)
- ❑ Αμυντικός Προγραμματισμός

Διαδικασία Ανάπτυξης Προγράμματος

- **[E1]** Προσδιορίστε τις εισόδους και εξόδους της διεργασίας που απαιτείται για την εκτέλεση του έργου.
 - Τα βήματα της διεργασίας θα περιγράψει το πρόγραμμα σας.
- **[E2]** Προσδιορίστε ποιες από τις εισόδους είναι μεταβλητές και ποιες σταθερές.
- **[E3]** Δώστε τη λεκτική περιγραφή της διεργασίας **αξιοποιώντας αφαιρετικότητα στις διεργασίες.**
- **[E4]** Ελέγξτε αν εσείς, εκτελώντας τις ενέργειες που η λεκτική περιγραφή ορίζει, πετυχαίνετε τον στόχο της διεργασίας, για παράδειγμα τον προσδιορισμό του βαθμού του μαθήματος.
- **[E5]** Μετασχηματίστε τη λεκτική περιγραφή σε C κώδικα.
- **[E6]** Μετασχηματίστε τον πηγαίο κώδικα σε εκτελέσιμο.

Λεκτική περιγραφή - Ορισμός

- είναι μια περιγραφή, σε προστακτική μορφή, των ενεργειών που πρέπει να κάνει ένας άνθρωπος για να φέρει σε πέρας το έργο που του αναθέτουμε.
- Περιγράφει τη διεργασία που πρέπει να εκτελεστεί για να πάρουμε την έξοδο με βάση τις εισόδους.
- Δημιουργείται **πριν τη συγγραφή του κώδικα.**

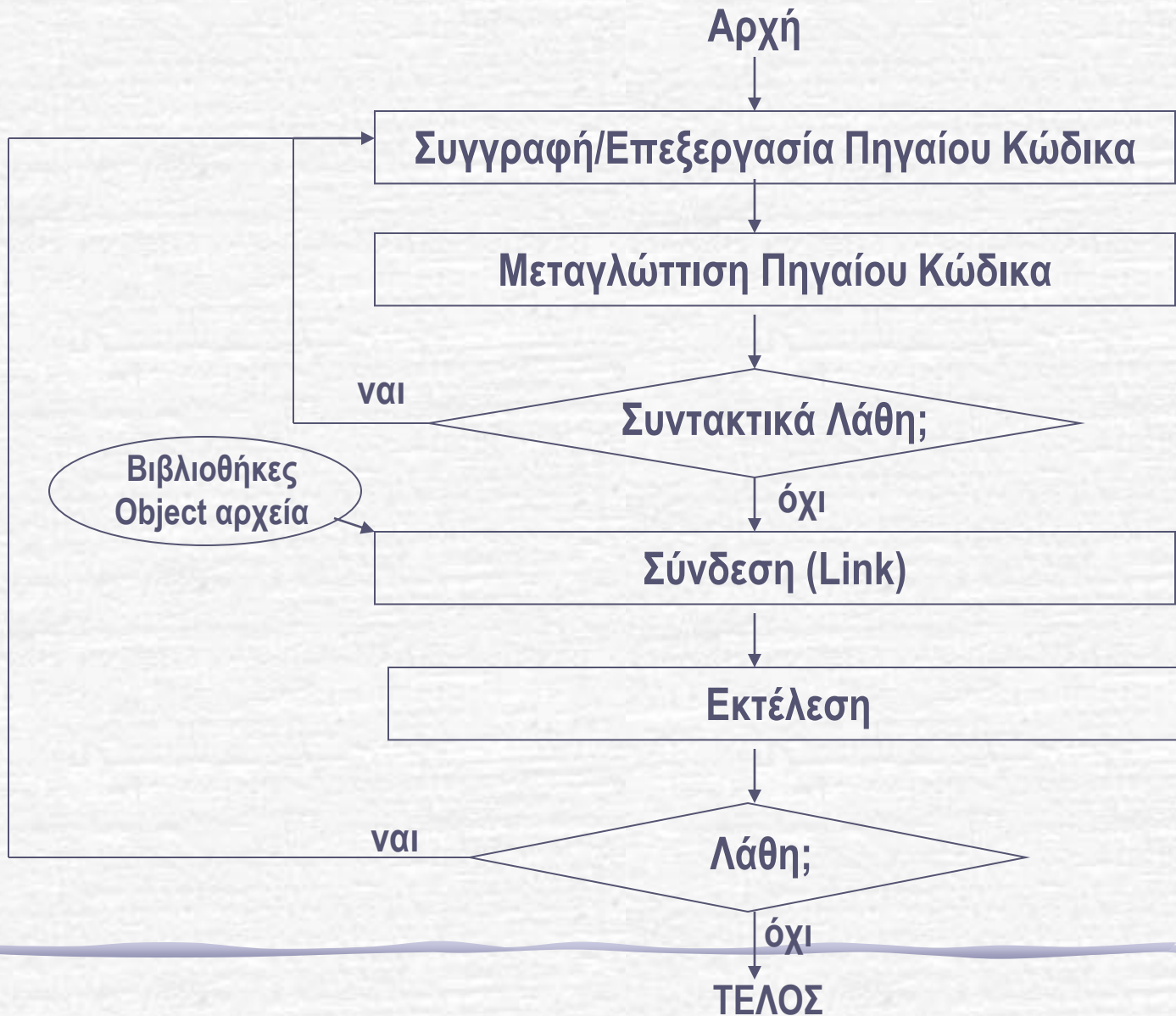
Δομημένη Υλοποίηση

1. Δομή Ακολουθίας Εντολών
2. Δομή Απόφασης (επιλογή υπό συνθήκη)
3. Δομή Επανάληψης (επανάληψη υπό συνθήκη)

Απαραίτητη προϋπόθεση το να ισχύουν τα εξής:

- σε κάθε δομή προγραμματισμού (και κατ' επέκταση στο πρόγραμμα) υπάρχει μόνο μία είσοδος (αρχή της ροής ελέγχου) και μία μόνο έξοδος (τέλος της ροής ελέγχου)
- η ροή μεταξύ εισόδου και εξόδου σε κάθε δομή είναι ομαλή και απρόσκοπτη

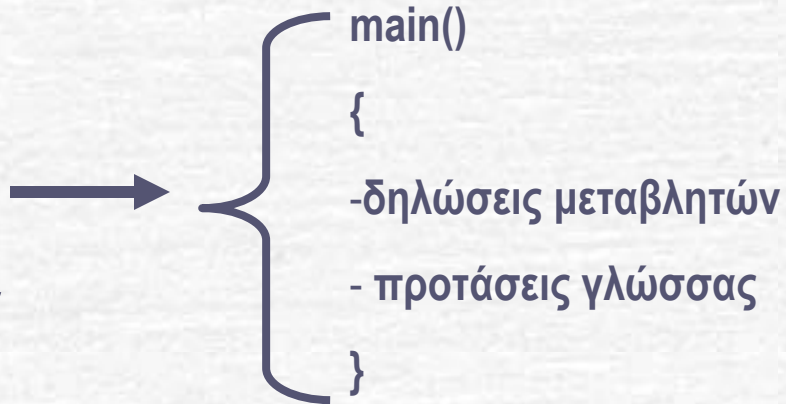
Φάση Υλοποίησης



Δομή Προγράμματος (1)

■ Γλώσσα C

- Εντολές προεπεξεργαστή
- Δηλώσεις συναρτήσεων
- Δηλώσεις μεταβλητών
- Κυρίως πρόγραμμα
- Ορισμοί συναρτήσεων



Δομή Προγράμματος (2)

■ Κυρίως Πρόγραμμα

- Είσοδος Δεδομένων
- Επεξεργασία Δεδομένων
- Έξοδος Αποτελεσμάτων



Προτάσεις
Γλώσσας

Το πρώτο μου πρόγραμμα σε C

Χρήση Standard I/O library
(για την
printf)

Ειδικός χαρακτήρας *new line*
(\n)

Βασική
συνάρτηση

Αρχή προγράμματος

Τέλος
προγράμματος

```
#include <stdio.h>
int main()
{
printf("Hello World!\n");
return 0;
}
```

Τέλος εντολής (;)

Παράμετρος συνάρτησης
(string)

Συνάρτηση εκτύπωσης

Ένα Απλό Πρόγραμμα (test.c)

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

→ εντολή προεπεξεργαστή

```
int k=0; float f; int n=5;
```

→ δηλώσεις μεταβλητών

```
int a;
```

```
int main ( )
```

→ κυρίως πρόγραμμα

```
{
```

```
int i;
```

```
/* this is a comment */
```

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

```
{ scanf("%d", &a);
```

→ εισαγωγή δεδομένων

```
    k=k+a;
```

→ επεξεργασία δεδομένων

```
    }
```

```
    f=k/n;
```

```
    printf("%f\n",f);
```

→ έξοδος δεδομένων

```
}
```

Βασικά Σημεία του Προγράμματος (1)

1. `#include <stdio.h>`

Περιέχεται στην αρχή κάθε προγράμματος και περιλαμβάνει οδηγίες προς τον προεπεξεργαστή.

2. `int main()`

Το σημείο έναρξης κάθε προγράμματος.

3. `{ }`

Δηλώνονται οι εντολές που είναι στη `main` και πρέπει να εκτελεστούν

Βασικά Σημεία του Προγράμματος (2)

4. printf(), scanf()

Βασικές συναρτήσεις εισόδου/εξόδου της βασικής βιβλιοθήκης της C

5. \n

Ακολουθία διαφυγής για την αναπαράσταση δύσκολων χαρακτήρων (π.χ. \t, \b, \\)

6. %f

Δηλώνεται η φόρμα (format) εμφάνισης πραγματικών.

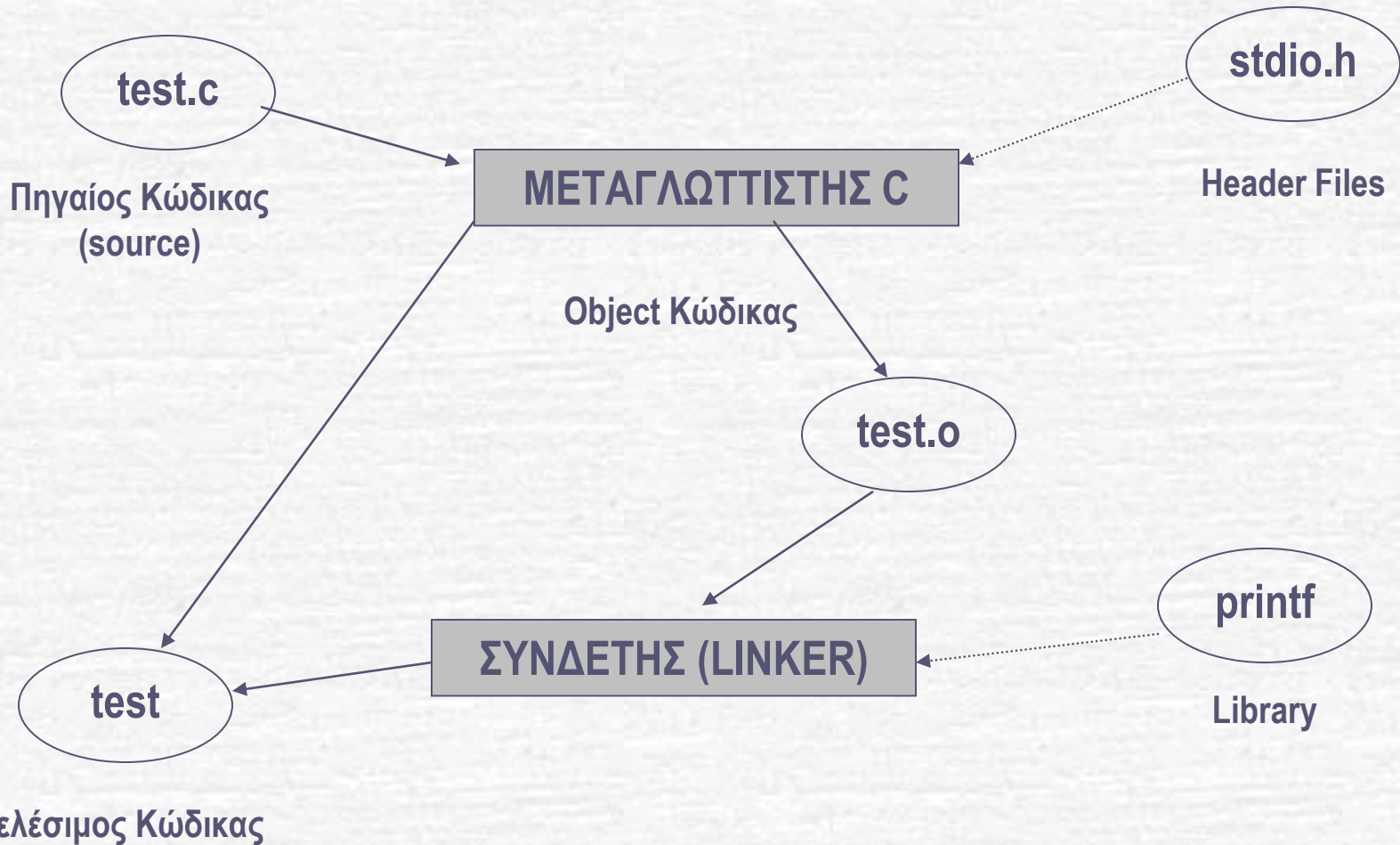
7. ;

Όλες οι δηλώσεις των C-προγραμμάτων τελειώνουν σε ; (semicolon), που δηλώνει τέλος εντολής.

Χρήση Σχολίων σε C

```
#include <stdio.h>
main()
{
    // This line is comments
    // This line also.
    // All those lines are NOT executed.
    /*
    Everything here is comments.
    we can place not needed lines of code here a=a+4;
    */
    printf("This program shows the use of comments \n");
}
```


Διαδικασία Μεταγλώττισης



C standard library (https://el.wikipedia.org/wiki/C_πρότυπη_βιβλιοθήκη)

Στοιχεία Γλώσσας Προγραμματισμού

- **Αλφάβητο**

- Οι χαρακτήρες από τους οποίους σχηματίζονται οι λέξεις της γλώσσας

- **Λεξιλόγιο**

- Οι λέξεις που χρησιμοποιεί η γλώσσα

- **Συντακτικό**

- Οι κανόνες σύνταξης των προτάσεων της γλώσσας

- **Σημασιολογία**

- Οι κανόνες ερμηνείας των προτάσεων της γλώσσας

Στοιχεία Γλώσσας Προγραμματισμού

- **Αλφάβητο**

- Οι χαρακτήρες από τους οποίους σχηματίζονται οι λέξεις της γλώσσας

- **Λεξιλόγιο**

- Οι λέξεις που χρησιμοποιεί η γλώσσα

- **Συντακτικό**

- Οι κανόνες σύνταξης των προτάσεων της γλώσσας

- **Σημασιολογία**

- Οι κανόνες ερμηνείας των προτάσεων της γλώσσας

Αλφάβητο

- Αποτελείται από 96 χαρακτήρες

- Χαρακτήρας κενού
- Χαρακτήρας ελέγχου οριζοντίου στηλοθέτη (tab)
- Χαρακτήρας ελέγχου καθέτου στηλοθέτη
- Χαρακτήρας αλλαγής σελίδας (form feed)
- Χαρακτήρας νέας γραμμής (new-line)

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

0 1 2 3 4 5 6 7 8 9

_ { } [] # () < > % : ; . ? * + - / ^ & | ~ = ! , \ " ' `

Λεξιλόγιο – Κατηγορίες Λέξεων

■ Δεσμευμένες λέξεις (reserved words)

- Λέξεις κλειδιά (π.χ. *int*, *if*, *for*)
- Ονόματα συναρτήσεων βιβλιοθήκης (π.χ. *printf()*, *isdigit()*)
- Ονόματα εντολών προεπεξεργαστή (π.χ. *#include*, *#define*)

■ Αναγνωριστές (identifiers)

- Ονόματα μεταβλητών, σταθερών, συναρτήσεων, τύπων δεδομένων, τα οποία δίνονται από τον προγραμματιστή

■ Τελεστές (operators)

- Ειδικά σύμβολα που παριστάνουν μία συγκεκριμένη στοιχειώδη διεργασία που εκτελείται σε δεδομένα

<code>auto</code>	<code>do</code>	<code>goto</code>	<code>signed</code>	<code>unsigned</code>
<code>break</code>	<code>double</code>	<code>if</code>	<code>sizeof</code>	<code>void</code>
<code>case</code>	<code>else</code>	<code>int</code>	<code>static</code>	<code>volatile</code>
<code>char</code>	<code>enum</code>	<code>long</code>	<code>struct</code>	<code>while</code>
<code>const</code>	<code>extern</code>	<code>register</code>	<code>switch</code>	
<code>continue</code>	<code>for</code>	<code>return</code>	<code>typedef</code>	
<code>default</code>	<code>float</code>	<code>short</code>	<code>union</code>	

Πίνακας από: Γ. Τσελίκη, Ν. Τσελίκας, C: Από τη Θεωρία στην Εφαρμογή

Στα νέα πρότυπα προστέθηκαν νέες λέξεις κλειδιά όπως:

`_Bool`, `_Complex`, **(C99)** `_Thread_local` **(C11)**

Κανόνες Δημιουργίας Αναγνωριστών (1)

- Αποτελούνται από γράμματα, αριθμούς και τον ειδικό χαρακτήρα “_”.
- Αρχίζουν πάντα από γράμμα ή “_” (στη δεύτερη περίπτωση ο επόμενος χαρακτήρας πρέπει να είναι μικρό γράμμα).
Δεν μπορεί να είναι ίδια με κάποια δεσμευμένη λέξη (C)
- Διάκριση μικρών και κεφαλαίων (case sensitive)
- Όριο μήκους (π.χ. 31, για ANSI C)

Παραδείγματα

~~2class~~ class1 ~~Class-1~~ Class_1 ~~Class#1~~ CLaSs1

Κανόνες Δημιουργίας Αναγνωριστών (2)

- Αποφύγετε ονόματα ενός χαρακτήρα όπως i , j , x , y .
- Χρησιμοποιήστε εκφραστικά ονόματα.
- Χρησιμοποιήστε μικρά γράμματα, για ονόματα μεταβλητών. Κατά κανόνα χρησιμοποιούμε κεφαλαία γράμματα για μακροεντολές
- Διατηρήστε ενιαίο κανόνα όσον αφορά το διαχωριστικό για μεταβλητές που αποτελούνται από 2 ή περισσότερες λέξεις.

Αναγνωριστές - Αναγνωσιμότητα

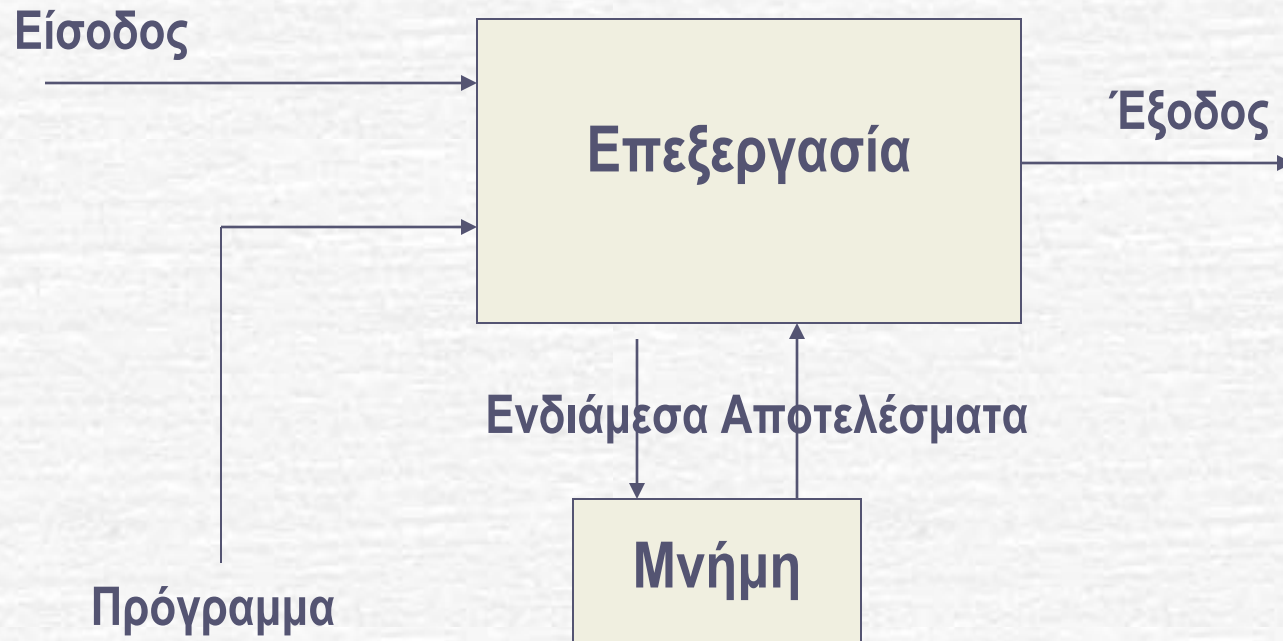
```
x=20;  
If ( x > y )  
    function1( );  
else  
    function2( );
```

```
baros=20;  
If ( baros > max_baros )  
    ADD_baros( );  
else  
    DECREASE_baros( );
```

Δεδομένα

- Είναι η είσοδος, η ακατέργαστη γνωστή πληροφορία, με την επεξεργασία της οποίας θα παραχθούν τα αποτελέσματα (η λύση του προβλήματος)
- Δεδομένα=τιμές ή πλειάδες τιμών που σχετίζονται με τις οντότητες του προβλήματος.

Δεδομένα



Μεταβλητές (1)

- Είναι υπολογιστικές οντότητες μέσω των οποίων χρησιμοποιούνται τα δεδομένα.
- Το περιεχόμενο (τιμή) μίας μεταβλητής μπορεί να μεταβάλλεται κατά τη διάρκεια εκτέλεσης ενός προγράμματος
- Η χρήση των μεταβλητών πηγάζει από την ανάγκη για
 - αποθήκευση δεδομένων
 - ανάκληση αποθηκευμένων δεδομένων

Μεταβλητές (2)

- Οι μεταβλητές αποτελούν αναφορά
 - σε μία θέση μνήμης
 - στο περιεχόμενο μίας θέσης μνήμης
- Χαρακτηριστικά μεταβλητής
 - το όνομά της (θέση μνήμης)
 - η τιμή της (περιεχόμενο)
- Οι μεταβλητές παριστάνουν οντότητες του προβλήματος που επιλύει το πρόγραμμα
 - Π.χ. Πρόβλημα υπολογισμού της επιφάνειας ενός ορθογωνίου.
Οντότητες: μήκος, πλάτος, εμβαδόν
Μεταβλητές: *mikos*, *platos*, *embadon*

Σταθερές - Τιμές

- Σταθερές (constants) είναι υπολογιστικές οντότητες που έχουν σταθερή τιμή κατά την διάρκεια εκτέλεσης ενός προγράμματος (π.χ. PI για το 3.14)
- Τιμές (literals) είναι δεδομένα που δεν χρειάζονται αποθήκευση, χρησιμοποιούνται όπως είναι (π.χ. οι αριθμοί 2, 3.5)

Δήλωση Σταθερών (1)

- `#define <όνομα> <τιμή>`
- `#define TRUE 1`
- `#define FALSE 0`
- `#define JANUARY 1`
- `#define PI 3.1453`

Δήλωση Σταθερών (2)

- Μεταβλητές που δεν αλλάζουν τιμή (Σταθερές)

π.χ. `const float pi=3.14;`

Σταθερά Απαρίθμησης

- `enum boolean { NO, YES};`
- `enum boolean { TRUE, FALSE};`
- `enum months { JAN=1, FEB, MAR, APR, MAY, JUN,
JUL, AUG, SEP, OCT, NOV, DEC};` 1

Τύποι Δεδομένων

- Τα δεδομένα δεν έχουν όλα τις ίδιες ιδιότητες, δεν είναι όλα του ίδιου τύπου.
- Τύπος δεδομένων = πεδίο τιμών + τρόποι διαχείρισης = πεδίο τιμών + πράξεις
- Τύποι δεδομένων
 - βαθμωτοί ή ατομικοί ή πρωτογενείς (*int, float, double, char*)
 - συναθροιστικοί ή σύνθετοι (πίνακες, δείκτες, ενώσεις, δομές)
(- ενσωματωμένοι
 - παραγόμενοι)

Πρωτογενείς Τύποι Δεδομένων

- Ακέραιος (int)
- Πραγματικός (float, double)
- Χαρακτήρας (char)
- Λογικός (η C δεν διαθέτει) αν και υπάρχει ο τύπος bool στη C99.

Τύποι Μεταβλητών

Τύπος	Συνηθισμένο μέγεθος (bytes)	Εύρος τιμών (min-max)	Ψηφία ακρίβειας
<code>char</code>	1	-128 ... 127	
<code>short int</code>	2	-32.768 ... 32.767	
<code>int</code>	4	-2.147.483.648...2.147.483.647	
<code>long int</code>	4	-2.147.483.648...2.147.483.647	
<code>float</code>	4	Μικρότερη θετική τιμή: $1.17 \cdot 10^{-38}$ Μεγαλύτερη θετική τιμή: $3.4 \cdot 10^{38}$	6
<code>double</code>	8	Μικρότερη θετική τιμή: $2.2 \cdot 10^{-308}$ Μεγαλύτερη θετική τιμή: $1.8 \cdot 10^{308}$	15
<code>long double</code>	8, 10, 12, 16		
<code>unsigned char</code>	1	0 ... 255	
<code>unsigned short int</code>	2	0 ... 65535	
<code>unsigned int</code>	4	0 ... 4.294.967.295	
<code>unsigned long int</code>	4	0 ... 4.294.967.295	

• Π.χ.

```
int a; /* Δήλωση ακέραιας μεταβλητής με όνομα a. */  
float b; /* Δήλωση πραγματικής μεταβλητής με όνομα b. */
```

Γιατί τύποι δεδομένων;

- Οικονομία χώρου (λόγω διαφορετικής αποθήκευσης)
- Καλύτερος έλεγχος κατά τη μετάφραση και εκτέλεση του προγράμματος

Δήλωση Μεταβλητής

- <τύπος δεδομένων> <λίστα μεταβλητών>;
 ή
 <τύπος δεδομένων> <μεταβλητή>=<τιμή>;
 π.χ. float mikos, platos, embadon;
 int i=1;
- Μεταβλητές που δεν αλλάζουν τιμή (Σταθερές)
 π.χ. const float pi=3.14;

Βασικές Πράξεις

Τελεστής ανάθεσης: $x=3;$

Αριθμητικοί τελεστές (+, -, /, *, %)

$x=5+3;$

$y=5*6;$

Διαφορά στην ερμηνεία του τελεστή διαίρεσης μεταξύ ακεραίων και πραγματικών τελεστών.

Τύπος «ακέραιος»

- Αναπαράσταση ακεραίων, θετικών ή αρνητικών αριθμών
- Λέξη κλειδί: *int*
- Εύρος: εξαρτάται από το μήκος λέξης του Η/Υ, π.χ. για λέξη 16-bit: -32768 έως 32767 ή 0 έως 65335
- Προσδιοριστές: short, long, unsigned
- Ο short έχει μήκος ≥ 16 bit, ο long ≤ 32 bit και ο int 16 ή 32 bits (C89)
- Ακέραιες τιμές: τις χειρίζεται σαν ακεραίους αριθμούς ανάλογα με το μέγεθος, π.χ. την 125 την χειρίζεται σαν *int*, ενώ την 135840 σαν *long int* (μπορούμε όμως να επιβάλουμε τον τύπο χειρισμού, π.χ. 6524L)

Μεγέθη C11

The size of type int is 4 bytes

The size of type short is 2 bytes

The size of type long is 4 bytes

The size of type long long is 8 bytes

min int: -2147483648, max int: 2147483648

min short: -32768, max short: 32767

min long: -2147483648, max long: 2147483648

```

#include <stdio.h>
#include <limits.h>
int main()
{
    /* INT_MAX είναι ο μέγιστος int: ορίζεται στο αρχείο κεφαλίδας
    limits.h. Ο τελεστής sizeof δίνει το μέγεθος ενός τύπου δεδομένου
    ή μίας μεταβλητής */
    printf( "The size of type int is %d bytes\n",sizeof(int) );
    printf( "The size of type short is %d bytes\n",sizeof(short) );
    printf( "The size of type long is %d bytes\n",sizeof(long) );
    printf( "The size of type long long is %d bytes\n",sizeof(long
long) );
    printf( "\nmin int: %d, max int: %d\n",INT_MIN,INT_MAX );
    printf( "\nmin short: %d, max short: %d\n",SHRT_MIN,SHRT_MAX );
    printf( "\nmin long: %d, max long: %d\n",LONG_MIN, LONG_MAX );
    return 0;
}

```

Τύπος «πραγματικός»

- Αναπαράσταση πραγματικών αριθμών
- Εκφράσεις
 - δεκαδικοί κινητής υποδιαστολής: 120.52, 0.035
 - Εκθετικής ή επιστημονικής μορφής: 1.2052e+02, 3.5e-2
- Λέξεις κλειδιά
 - *float*: απλής ακρίβειας (6-7 δεκαδικά ψηφία)
 - *double*: διπλής ακρίβειας (14-15 δεκαδ. ψηφία)
- Προσδιοριστές: *long* για το *double*
- Πραγματικές τιμές: θεωρούνται σαν *double*
π.χ. 0.13, 56.48, 8e-3, 15e05, 0.004e-0.4

Μεγέθη C11

<i>Αριθμός με δεκαδικά</i>	<i>Επιστημονική σημειογραφία</i>	<i>Εκθετική σημειογραφία</i>
123.456	1.23456x10 ²	1.23456e+02
0.00002	2.0x10 ⁻⁵	2.0e-5
50000.0	2.0x10 ⁴	5.0e+04

The size of type float is 4 bytes

The size of type double is 8 bytes

min float: 1.175494e-038, max float: 3.402823e+038

min double: 2.225047e-308, max double: 1.797693e+308

Διαδικαστικός προγραμματισμός» – Η γλώσσα C (από Πάρι Μαστοροκόστα), Έκδοση:
Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών

```
#include <stdio.h>
#include <float.h> /* για τα όρια του float */
int main()
{
    printf( "The size of type float is %d bytes\n",sizeof(float) );
    printf( " The size of type double is %d bytes\n",sizeof(double) );
    printf( "\nmin float: %e, max float: %e\n",FLT_MIN,FLT_MAX);
    printf( "\nmin double: %e, max double: %e\n",DBL_MIN,DBL_MAX);
    printf( "\n\nPress any key to continue" );
    return 0;
}
```

Πρότυπο IEEE 754

S Πρόσημο

1 bit

E Εκθέτης

8 bits

Σ Συντελεστής

23 bits

αριθμός $X = (-1)^S \cdot (1 + \Sigma) \cdot 2^{E-127}$,

πόλωση = 127

(s=πρόσημο, «0»=θετικός, «1»=αρνητικός)

A) Απλής ακρίβειας (E=8, Σ=23, σύνολο 32 bit) και

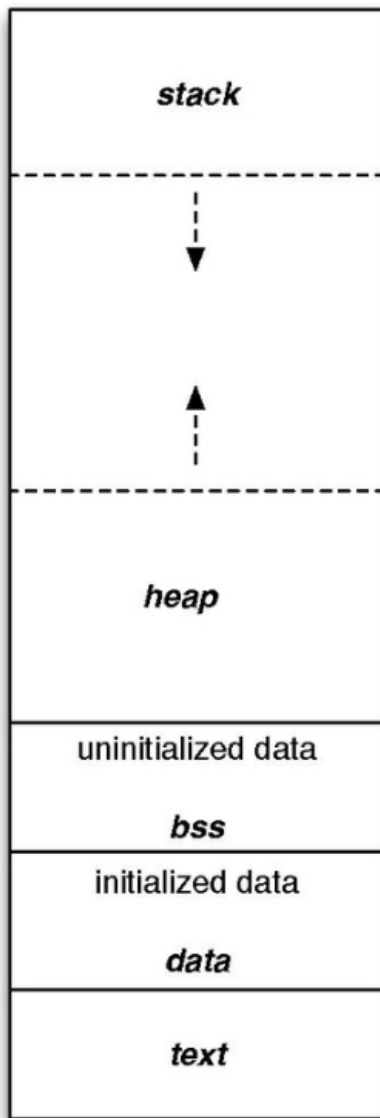
B) Διπλής ακρίβειας (E=11, Σ=52, σύνολο 64 bit)

Τύπος «χαρακτήρας»

- Αναπαράσταση απλών χαρακτήρων του αλφαβήτου της γλώσσας
- Λέξη κλειδί: `char`
- Η `C` χειρίζεται τους χαρακτήρες σαν ακεραίους. Κάθε χαρακτήρας θεωρείται σαν ακέραιος με τιμή τον αντίστοιχο κωδικό ASCII (στο δεκαδικό σύστημα, π.χ. 0 και 48)
- Οι τιμές σε μία τέτοια μεταβλητή αποδίδονται όταν ο χαρακτήρας εμπεριέχεται σε `' '`. Για παράδειγμα:
 - ‘`a`’ είναι ο χαρακτήρας `a`
 - ‘`b`’ είναι ο χαρακτήρας `b`
 - ‘`c`’ είναι ο χαρακτήρας `c`

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Memory Layout of a C program



https://commons.wikimedia.org/wiki/File:Program_memory_layout.pdf

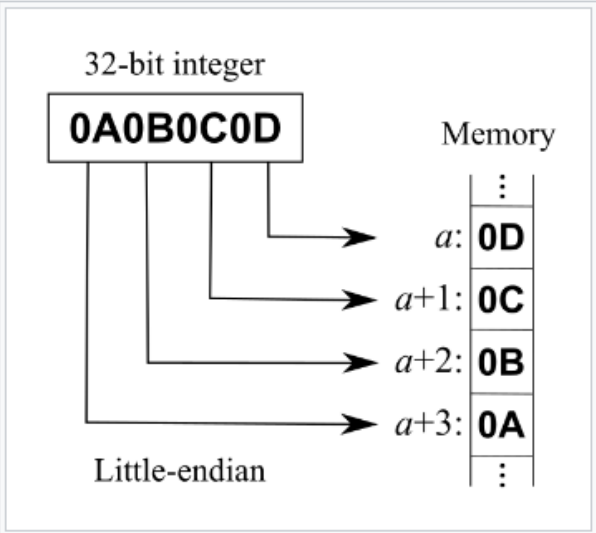
This file is licensed under the [Creative Commons Attribution-Share Alike 3.0 Unported](https://creativecommons.org/licenses/by-sa/3.0/) license.


Διεύθυνση μεταβλητής

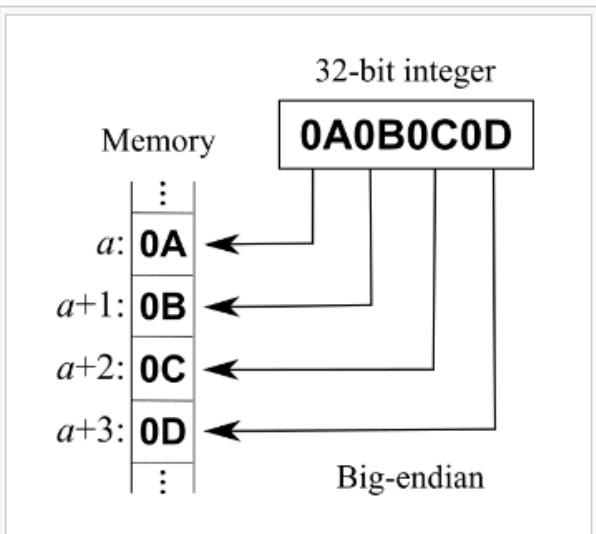
- Κάθε μεταβλητή καταλαμβάνει ένα ή περισσότερα byte στη μνήμη
 - char (1B), short (2B), int (4B), long (8B), float (4B), double (8B), κ.τ.λ.
 - Η διεύθυνση του πρώτου byte είναι η **διεύθυνση της μεταβλητής**


Computer		Programmers		
Address	Content	Name	Type	Value
90000000	00	sum	int (4 bytes)	000000FF (255 ₁₀)
90000001	00			
90000002	00			
90000003	FF			
90000004	FF	age	short (2 bytes)	FFFF (-1 ₁₀)
90000005	FF			
90000006	1F	average	double (8 bytes)	1FFFFFFFFFFFFFFF (4.45015E-308 ₁₀)
90000007	FF			
90000008	FF			
90000009	FF			
9000000A	FF			
9000000B	FF			
9000000C	FF			
9000000D	FF			
9000000E	90	ptrSum	int* (4 bytes)	90000000
9000000F	00			
90000010	00			
90000011	00			

Note: All numbers in hexadecimal



Little-endian: Το λιγότερο σημαντικό byte αποθηκεύεται στην "μικρότερη" θέση μνήμης. 



Big-endian: Το σημαντικότερο byte αποθηκεύεται στην "μικρότερη" θέση μνήμης. 

<https://en.wikipedia.org/wiki/Endianness>

ΔΕΙΚΤΕΣ

- Η C δίνει τη δυνατότητα να αποθηκεύουμε **διευθύνσεις μνήμης** σε ειδικές μεταβλητές που λέγονται **pointers**

```
int *i;      // pointer to int
char *c;     // pointer to char
double *d;   // pointer to double
```

- Ο κάθε pointer είναι associated με κάποιον τύπο δεδομένων.
 - Π.χ., το `i` στο παραπάνω παράδειγμα δείχνει σε `int`
- Ορίζονται με ένα αστεράκι πριν το όνομα της μεταβλητής
 - Π.χ., στο παρακάτω, μόνο το `p` είναι pointer

```
int i, a[100], *p;
```

Τελεστές Δεικτών

□ Τελεστής διεύθυνσης (address operator): **&**

- Επιστρέφει τη διεύθυνση μιας μεταβλητής

```
int i, *p;  
p = &i;    // p = address of i
```

□ Τελεστής αποαναφοράς (dereference operator): *****

- Μας επιτρέπει να προσπελάσουμε το περιεχόμενο της μνήμης που δείχνει ένας pointer

```
// συνέχεια από το προηγούμενο  
*p = 42;  
printf("%d\n", i);        // 42   i /= 2;  
printf("%d\n", *p);      // 21
```

Τελεστές Δεικτών

□ Τελεστής ανάθεσης: =

```
int i=5, *p, *q;  
p = &i; // p = address of i  
q = p;  
printf("%d\n", *q); //  
5
```

□ Προσοχή! Άλλο η ανάθεση δείκτη, κι άλλο η ανάθεση της τιμής που δείχνει ο δείκτης

```
q = p; // Ανάθεση δείκτη  
*q = *p; // Ανάθεση τιμής
```

Παράδειγμα

```
int x=1;
int y=2;
int *p;
p = &x; /* η p δείχνει τη διεύθυνση της x */
y = *p; /* η y γίνεται 1, δηλαδή η τιμή της x */
*p = 0; /* η x γίνεται 0 */
*p = *p + 10; /* x=x+10 */

*p += 1; /* x=x+1 */
++*p; /* x=x+1 */
(*p)++; /* x=x+1 */
```

Παράδειγμα

```
#include <stdio.h>

main()
{
    int x=5; int y=10; int *ptr; int **ptr2;
    ptr=&x; ptr2=&ptr;
    *ptr2=&y;
    printf("%d", *ptr);
}
```


Τελεστές Δεικτών

- Οι τελεστές **&** και ***** είναι «αντίστροφοι», συνεπώς εν γένει αλληλοαναιρούμενοι

```
int i, j, *p=NULL, *q=&i;
i = *&j;    // i=j
i = &*j;    // compile error! (j not ptr)
p = &*q;    // p=q
```

- Γενικά είναι καλό να αρχικοποιείτε πάντα τους pointers (π.χ., με NULL), αλλιώς μπορεί να «δείχνουν» σε αυθαίρετη διεύθυνση με ολέθρια αποτελέσματα!
 - Τι είναι το NULL???
 - Είναι ειδική τιμή (συγκεκριμένα η τιμή 0), που σημαίνει τον **κενό δείκτη**, δηλαδή έναν pointer που δεν δείχνει πουθενά
 - Ο NULL pointer δεν αποδεικτοδοτείται. Π.χ., το `p=NULL; *p = 5;` θα δημιουργήσει λάθος, δεν θα βάλει το 5 στη διεύθυνση 0.

NULL

□ Τι είναι το **NULL**???

- Είναι ειδική τιμή (συγκεκριμένα η τιμή 0), που σημαίνει τον **κενό δείκτη**, δηλαδή έναν pointer που δεν δείχνει πουθενά
- Ο NULL pointer δεν αποδεικτοδοτείται
- Π.χ., ο παρακάτω κώδικας θα δημιουργήσει λάθος, δεν θα βάλει το 5 στη διεύθυνση 0

```
int *p = NULL;  
*p = 5;
```

□ Και τι είναι το **void ***???

- Είναι ο τύπος δεδομένων γενικού δείκτη (δείκτη σε κενό)
- Δηλαδή δείκτης που δεν είναι associated με συγκεκριμένο τύπο (int, char, κ.τ.λ.)
- Κανονικός δείκτης, απλά δεν μπορεί να κάνει **pointer arithmetic**
- Μα, τι είναι pointer arithmetic??? 😊

Περιγραφή Μαθήματος

- Τιμές (Σταθερές)
- Ανάθεση/Καταχώρηση
- Είσοδος δεδομένων (scanf)
- Έξοδος αποτελεσμάτων (printf)

Τιμές (Σταθερές)

- Ο τύπος μίας τιμής (σταθεράς) είναι φανερός, αναγνωρίζεται άμεσα, από την εμφάνισή της, δεν χρειάζεται δήλωση:

π.χ.

123	->	int
25865L	->	long int
123.5	->	double
123.5f	->	float
'A'	->	char

- Ένα 0 σημαίνει οκταδικός, ένα 0x σημαίνει δεκαεξαδικός, για παράδειγμα το 31 γίνεται 037 ή 0x1f
- Μία σταθερά χαρακτήρα μπορεί να γραφεί είτε ως '\ooo' όπου ooo είναι ένα ως τρία οκταδικά ψηφία, ή '\xhh' όπου hh ένα ή περισσότερα δεκαεξαδικά ψηφία.
- Επισήμανση για το '\0'.

Δήλωση Σταθερών

- `#define <όνομα> <τιμή>`
- `#define TRUE 1`
`#define FALSE 0`
`#define JANUARY 1`
- `const int TRUE=1;`
`const double e=2.17;`

Σταθερά Απαρίθμησης

```
enum boolean {NO, YES};
```

```
enum months {JAN=1, FEB, MAR, APR, MAY,  
JUN, JUL, AUG, SEP, OCT, NOV, DEC};
```

Οι απαριθμήσεις παρέχουν ένα βολικό τρόπο για τη σύνδεση σταθερών τιμών με ονόματα. Οι μεταβλητές απαρίθμησης παρέχουν τη δυνατότητα ελέγχου και συχνά είναι καλύτερες από την `#define`.

Προτάσεις Προεπεξεργαστή

- Ειδικές προτάσεις (δεν τελειώνουν σε;)
- Εκτελούν κάποια επεξεργασία του πηγαίου κώδικα πριν τη μεταγλώττιση
- Πρόταση συμπερίληψης
 - `#include <όνομα αρχείου>` ή `#include "όνομα αρχείου"`
 - π.χ. `#include <stdio.h>`, `#include <string.h>`
- Πρόταση μακρο-αντικατάστασης (όχι σε εισαγωγικά ή κείμενο)
 - `#define <όνομα> <κείμενο>`
 - π.χ. `#define PI 3.1415`

Αριθμητικές Μετατροπές (101)

Έμμεσες

- Αν ένας από τους τελεστές είναι long double μετατρέπεται και ο άλλος σε long double
- Αλλιώς, αν ένας από τους τελεστές είναι double μετατρέπεται και ο άλλος σε double
- Αλλιώς αν ένας από τους τελεστές είναι float μετατρέπεται και ο άλλος σε float
- Αλλιώς ο char ή ο short μετατρέπεται σε int.

Τέλος αν ένας από τους 2 τελεστές είναι long μετατρέπεται και ο άλλος σε long

Typecasting

- Δοκιμάστε το παρακάτω πρόγραμμα

```
#include <stdio.h>
int main()
{
float x;    x = 6/5;
printf("%f\n", x); return 0;
}
```

- Τι θα τυπώσει;;;
 - 1.0000
- Γιατί; Πως θα το κάνουμε να μας δείξει 1.20000

Typecasting

□ Διάφοροι τρόποι:

```
x = ((float) 6) / ((float) 5);
```

```
x = ((float) 6) / 5;
```

```
x = (float) (6/5);
```

```
x = 6.0/5;
```

```
x = 6.0f/5;
```

```
x = 6/5.0f;
```

Αριθμητικές Μετατροπές (2)

Σε οποιαδήποτε παράσταση μπορούν να επιβληθούν ρητές μετατροπές τύπου με έναν μοναδικό τελεστή που λέγεται *προσαρμογή* (*cast*).

Στην κατασκευή
(*όνομα τύπου*) παράσταση

η παράσταση μετατρέπεται στον κατανομαζόμενο τύπο με βάση τους κανόνες μετατροπής που προαναφέραμε.

Π.χ `sqrt((double) n);`

Είσοδος Δεδομένων (1)

- Οι συναρτήσεις εισόδου, εξόδου, μακροεντολές αποτελούν το ένα τρίτο της βιβλιοθήκης.
- *Ρεύμα* είναι μία πηγή ή προορισμός που σχετίζονται με δίσκο ή περιφερειακό. Υπάρχουν ρεύματα κειμένου και δυαδικά αν και σε μερικά (UNIX) θεωρούνται πανομοιότυπα.
- Ρεύμα κειμένου είναι μία ακολουθία γραμμών. Κάθε γραμμή έχει μηδέν ή περισσότερους χαρακτήρες και τερματίζεται με '\n' σαν επαναφορά κεφαλής και αλλαγή γραμμής. Δυαδικό ρεύμα είναι μία ακολουθία ανεπεξέργαστων bytes που καταγράφουν δεδομένα.

Είσοδος Δεδομένων (2)

Ένα ρεύμα συνδέεται με ένα αρχείο ή συσκευή ανοίγοντάς την. Η σύνδεση διακόπτεται με το κλείσιμο του ρεύματος. Το άνοιγμα ενός αρχείου επιστρέφει ένα δείκτη σε αντικείμενο τύπου FILE.

Όταν ένα πρόγραμμα εκτελείται, τα ρεύματα stdin, stdout, stderr είναι ήδη ανοιχτά.

Streams

□ Streams (ροές)

- Στα συστήματα Unix υπάρχει η έννοια των streams, για τη ροή δεδομένων προς και από ένα πρόγραμμα
- Διάβασμα αρχείων, πληκτρολογίου, γράψιμο σε περιφερειακά, ...
- Κάποια είναι standard
 - **stdin**: η ροή εισόδου δεδομένων (π.χ., το πληκτρολόγιο)
 - **stdout**: ροή εξόδου δεδομένων (π.χ., η οθόνη)
 - **stderr**: ροή εξόδου για αναφορά σφαλμάτων (π.χ., η οθόνη)

Είσοδος Δεδομένων

- Εντολή/Συνάρτηση *scanf()*:
scanf("<προσδιοριστής>", &<μεταβλητή>);
(h, l πρόθεμα προσδιορισμού μεγέθους)
- Προσδιοριστές:
d, i, o, u, x: ακέραιοι
c, s: χαρακτήρες, συμβολοσειρές
e, f, g: αριθμός κινητής υποδιαστολής
- Χαρακτήρες λευκών διαστημάτων: κενό, στηλογνώμονας (tab), αλλαγή γραμμής, αλλαγή σελίδας

Έξοδος Αποτελεσμάτων (1)

- Εντολή/Συνάρτηση *printf()*:
printf("<προσδιοριστής>", <ακολουθία μεταβλητών>);
- Έξοδος ακεραίων: *printf*("%d", num);
Προσδιοριστές: %d, %i, %x (ή %X), %o
(h,l πρόθεμα προσδιορισμού μεγέθους)
- Έξοδος πραγματικών: *printf*("%f", num);
Προσδιοριστές: %f, %e (ή %E), %g(ή %G)
- Έξοδος χαρακτήρων: *printf*("%c", ch); (χαρακτήρας)
printf("%d",ch); (κωδικός ASCII)
Προσδιοριστές: %c, %s.

- `%d` ακέραιος
- `%6d` ακέραιος με πλάτος 6 χαρακτήρες
- `%f` κινητής υποδιαστολής
- `%6f` κινητής υποδιαστολής με 6 ψηφία
- `%.2f` δύο δεκαδικά ψηφία
- `%6.2f` πλάτος έξι ψηφίων με δύο δεκαδικά ψηφία.

Έξοδος Αποτελεσμάτων (2)

■ Εκφράσεις

- Οθόνης: '\t', '\n'

- Αριθμών:

- %<ακέρ><προσδιορ> (καθορισμός πλάτους πεδίου), π.χ. %3d
- % [<ακέρ>] [, <ακέρ>] <προσδιορ> (καθορισμός πλάτους πεδίου και δεκαδικών ψηφίων), π.χ. %6.1f, %.2f, %6f

Σημαία	Λειτουργία
-	Αριστερή στοίχιση της εξόδου στο πεδίο πλάτους (η προκαθορισμένη στοίχιση είναι στα δεξιά)
+	Προσθήκη του προσήμου στις θετικές τιμές
#0	Προσθήκη του 0 μπροστά από οκταδικούς
#x	Προσθήκη του 0x μπροστά από δεκαεξαδικούς αριθμούς
#X	Προσθήκη του 0X μπροστά από δεκαεξαδικούς αριθμούς
0	Προσθήκη των απαιτούμενων μηδενικών μπροστά από την τιμή, ώστε να καλυφθεί το πεδίο πλάτους.
κενός χαρακτήρας	Προσθήκη του κενού χαρακτήρα μπροστά από τις μηδενικές τιμές

[Από «Διαδικαστικός προγραμματισμός» – Η γλώσσα C \(από Πάρι Μαστοροκώστα\), Έκδοση: Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών](#)

```
#include <stdio.h>
int main()
{
int x=323;
printf( "1:\t%-5d\n",x );
printf( "2:\t%+5d\n",x );
printf( "3:\t% d\n",x );
printf( "4:\t%#o\n",x );
printf( "5:\t%#x\n",x );
printf( "6:\t%#X\n",x );
printf( "7:\t%05d\n",x );
return 0;
}
```

Από «Διαδικαστικός προγραμματισμός» – Η γλώσσα C (από Πάρι Μαστοροκώστα), Έκδοση: Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών

Σύνολο Ακολουθιών Διαφυγής

\a	χαρακτήρας προειδοποίησης
\b	οπισθοχώρηση
\f	αλλαγή σελίδας
\n	νέα γραμμή
\r	επαναφορά κεφαλής
\t	οριζόντιος στηλογνώμονας
\v	κατακόρυφος στηλογνώμονας
\\	ανάποδη κάθετος
\?	λατινικό ερωτηματικό
\'	μονό εισαγωγικό
\''	διπλό εισαγωγικό
\ooo	οκταδικός αριθμός
\xhh	16-δικός αριθμός

Άλλες Συναρτήσεις Εισόδου Εξόδου

- Εντολή/Συνάρτηση `getchar()`:
`char c; c=getchar();`
- Εντολή/Συνάρτηση `putchar(char out)`
`char out; putchar(out);`

Παράδειγμα-Πρόγραμμα 1

```
#include <stdio.h>

main()
{
    int num;

    printf("Δώσε ένα αριθμό μεταξύ 65 και 90: \t");
    scanf("%d", &num);
    printf("\n Χαρακτήρας: %c\t ASCII κωδικός: %d", num, num);
}
```


ΑΠΟΤΕΛΕΣΜΑ

Δώσε ένα αριθμό μεταξύ 65 και 90: **70**

Χαρακτήρας: **F**

Κωδικός: **70**

Παράδειγμα-Πρόγραμμα 2

```
#include <stdio.h>
int main()
{
    int a,b;
    a = 2;
    b = 1;
    printf ("%d\n",a) ;
    printf ("%d\n",b) ;
    scanf ("%d", &a) ;
    scanf ("%d", &b) ;
    printf ("%d\n",a) ;
    printf ("%d\n",b) ;
}
```

Παράδειγμα-Πρόγραμμα 3

```
#include <stdio.h>
int main()
{
    float a,b;
    a = 2;
    b = 1;
    printf("%.5f\n",a);
    printf("%.5f\n",b);
    scanf("%f",&a);
    scanf("%f",&b);
    printf("%.2f %.3f",a,b);
}
```

Άσκηση 1

Να γραφεί πρόγραμμα σε C το οποίο διαβάζει δύο ακεραίους αριθμούς από το πληκτρολόγιο και στην συνέχεια τυπώνει το άθροισμα, την διαφορά, το γινόμενο, το ακέραιο ηλίκο και το υπόλοιπο της διαίρεσης των δύο αριθμών αυτών. Παράδειγμα εκτέλεσης:

Για $x = 15$ και $y = 4$ η έξοδος είναι:

$$15 + 4 = 19$$

$$15 - 4 = 11$$

$$15 * 4 = 60$$

$$15 / 4 = 3$$

$$15 \bmod 4 = 3$$

Άσκηση 2

Να γραφεί πρόγραμμα C το οποίο:

- (i) διαβάζει από το πληκτρολόγιο έναν double **d**
- (ii) αποθηκεύει τον d σε έναν float **f** με χρήση typecasting (float)
- (iii) τυπώνει τους d και f με ακρίβεια 12 δεκαδικών ψηφίων.

Άσκηση 3

- i. Γράψτε πρόγραμμα σε C το οποίο ζητάει από τον χρήστη να δώσει τις συντεταγμένες δύο σημείων στο επίπεδο. Στην συνέχεια, υπολογίζει την ευκλείδεια απόσταση των σημείων αυτών και την εκτυπώνει. (Υπόδ: πρέπει να χρησιμοποιηθεί η συνάρτηση υπολογισμού της τετραγωνικής ρίζας `sqrt()` της `math.h`)
- ii. Να γραφεί πρόγραμμα σε C το οποίο διαβάζει από το πληκτρολόγιο έναν πραγματικό αριθμό. Στην συνέχεια, θεωρώντας ότι ο αριθμός αυτός αναπαριστά θερμοκρασία σε βαθμούς Φαρενάιτ, να γίνει μετατροπή σε κλίμακα Κελσίου και το αποτέλεσμα να τυπωθεί. Υπόδ: ο τύπος μετατροπής από Φαρενάιτ σε Κελσίου είναι $C = (5/9) * (F - 32)$
- iii. Να γραφεί πρόγραμμα σε C το οποίο διαβάζει από το πληκτρολόγιο δύο αριθμούς και τους εναλλάσσει.
- iv. Να γραφεί πρόγραμμα το οποίο διαβάζει ένα αριθμό από το πληκτρολόγιο, υπολογίζει την τρίτη δύναμή του και τον εκτυπώνει.