

Εισαγωγή στην Πληροφορική & στον Προγραμματισμό

Αρχές Προγραμματισμού Η/Υ (με τη γλώσσα C)

Διάλεξη #1

Πέμπτη, 22 Μαρτίου 2024

Παναγιώτης Παύλου

c-programming-24@allos.gr

Γενικά για το μάθημα

Εσείς και εμείς μαζί

Δομή Μαθήματος

Στο μάθημα υπάρχουν δύο παράλληλες ροές. Για τη ροή του προγραμματισμού με τη γλώσσα C, οι «εργαστηριακές διαλέξεις» πραγματοποιούνται ...

Σε κάθε διάλεξη θα δίνονται απλές **εργασίες** για το σπίτι και λίγες ημέρες αργότερα πρέπει να τις **υποβάλλετε ηλεκτρονικά**. Η βαθμολογία τους συμμετέχει στην τελική βαθμολογία του μαθήματος, βάσει όσων έχουν ειπωθεί με τον διδάσκοντα Καθηγητή κ. Λ. Αλεξόπουλο.

Η τελική εξέταση θα γίνεται πάνω σε Η/Υ με ηλεκτρονικό τρόπο, ανάλογα με τα τότε ισχύοντα.

Σκοπός

Ο σκοπός του προγραμματιστικού μέρους του μαθήματος, στο τέλος του εξαμήνου, είναι ο κάθε φοιτητής και η κάθε φοιτήτρια:

- Να έχει τη βασική αντίληψη για τα ζητήματα του προγραμματισμού
- Να μπορεί να κατανοήσει τη λειτουργία ενός προγράμματος ή ενός τμήματος κώδικα
- Να μπορεί ο κάθε συμμετέχων να δημιουργήσει ένα σχετικά απλό πρόγραμμα σε επίπεδο αλγορίθμου και τεχνικής λεπτομέρειας
- Να προετοιμαστεί στη χρήση του προγραμματισμού για επιστημονικές και άλλες εφαρμογές

Υλικό

Το υλικό το οποίο σας παρέχεται στα πλαίσια του μαθήματος, σας παρέχει όλες τις απαιτούμενες γνώσεις ώστε να το ολοκληρώσετε επιτυχώς, ενώ θα είναι στη διάθεσή σας με διάφορες μορφές:

- Παρουσιάσεις από τις διαλέξεις (σε μορφή PDF)
- Ηλεκτρονική (σε PDF) και έντυπη μορφή του «Εγχειριδίου της Γλώσσας Προγραμματισμού C»
- Αναφορές (links) σε διάφορα site με σχετικό υλικό (ως links μέσα στα PDF)
- Προτεινόμενα σχετικά βιβλία διεθνούς βιβλιογραφίας
 - [K & R – The C programming language](#)
 - [Deitel-Deitel – C How to Program](#)
- Κοινότητες προγραμματιστών (ενδεικτικά το [SO](#) και [αυτό](#))
- Τη σελίδα ερωταποκρίσεων του μαθήματος

Τι θα μάθουμε

Η παραπάνω ύλη θα καλύπτει τουλάχιστον τα εξής αντικείμενα, με τη χρήση της γλώσσας προγραμματισμού C:

- Η πληροφορία στον Η/Υ
 - Αριθμοί
 - Πίνακες / Κείμενα
 - Δομές
- Αλγόριθμοι
 - Διαγράμματα ροής
 - Ροή εκτέλεσης
 - Έλεγχο ροής
- Συναρτήσεις
 - Δημιουργία
 - Βιβλιοθήκες συναρτήσεων
- Βασικές (standard) βιβλιοθήκες
 - Διαχείριση μνήμης
 - Ανάγνωση και εγγραφή αρχείων στον Η/Υ
 - Άλλες βιβλιοθήκες συναρτήσεων

Σημεία προσοχής

- Επειδή η περισσότερη ουσιαστική πληροφορία πάνω στον προγραμματισμό είναι στην Αγγλική γλώσσα, παρότι οι διαλέξεις είναι στα Ελληνικά, θα παρουσιάζονται και θα χρησιμοποιούνται οι **Αγγλικοί όροι**
- Ο προγραμματισμός είναι τέχνη γι' αυτό δεν γίνεται να τον μάθει κάποιος διαβάζοντας 1 εβδομάδα πριν τις εξετάσεις. **Απαιτεί παρακολούθηση, εξάσκηση και συζήτηση**
- **Ερωτήσεις** κατά τη διάρκεια του μαθήματος μπορούν να γίνουν σε συγκεκριμένα σημεία.
- Είμαστε **στη διάθεσή σας** για την επίλυση αποριών ή προβλημάτων με την προϋπόθεση ότι αυτή γίνεται αφού έχετε καταβάλει πρώτα τη σχετική προσπάθεια μόνοι σας και αφού έχετε ελέγξει μήπως υπάρχει ήδη απαντημένη η ερώτηση στο σχετικό site.

ΠΡΟΣΟΧΗ! Οι ερωτήσεις αποστέλλονται με e-mail στο:

c-programming-24@allos.gr

ενώ οι απαντήσεις θα δημοσιεύονται στο

<https://qna.c-programming.allos.gr>

Ερωτήσεις?

- Διαβάστε τις σημειώσεις, διαβάστε τις διαφάνειες και δείτε τα videos **πριν** ρωτήσετε
- **Συμβουλευτείτε** τη σελίδα ερωταποκρίσεων του μαθήματος
<https://qna.c-programming.allos.gr>
- **Στείλτε** τις ερωτήσεις σας πριν και μετά το μάθημα στο
c-programming-24@allos.gr
- Εάν έχετε **πρόβλημα** με κάποιο κώδικα στείλτε μαζί τον κώδικα και τα μηνύματα λάθους από το CLI ως κείμενα με copy/paste. Εάν θεωρείτε ότι επιπλέον βοηθά και ένα στιγμιότυπο οθόνης, είναι καλοδεχούμενο.
- Επαναλαμβάνουμε : Μην στείλετε ποτέ κώδικα ως εικόνα μας είναι παντελώς άχρηστος!



Γιατί γλώσσες προγραμματισμού

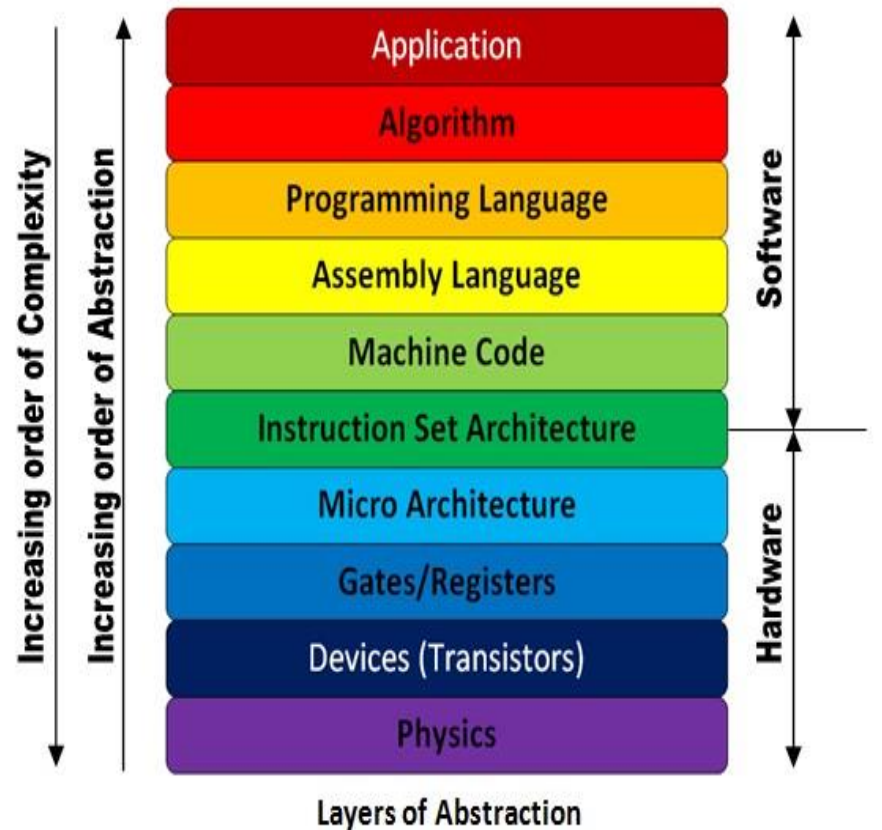
Τι ξεχωρίζει τη γλώσσα C

Αφαιρετικά επίπεδα λειτουργίας

Ο Η/Υ είναι μία συσκευή η οποία εκτελεί εντολές αδιάκοπα. Το λογισμικό ([software](#)) γράφεται για κάποιο **σκοπό**. Αποτελείται από τις εντολές που καθορίζουν το πώς θα λειτουργήσει η συσκευή ([hardware](#)), ώστε να επιτευχθεί ο παραπάνω σκοπός.

Τόσο το hardware όσο και το software παραδοσιακά στους Η/Υ οργανώνονται σε διάφορα επίπεδα αφαίρεσης ([abstraction layers](#)). Η λέξη επίπεδα (layers) δεν χρησιμοποιείται τυχαία αλλά για να τονίσει τη στρωματοποιημένη δομή τους.

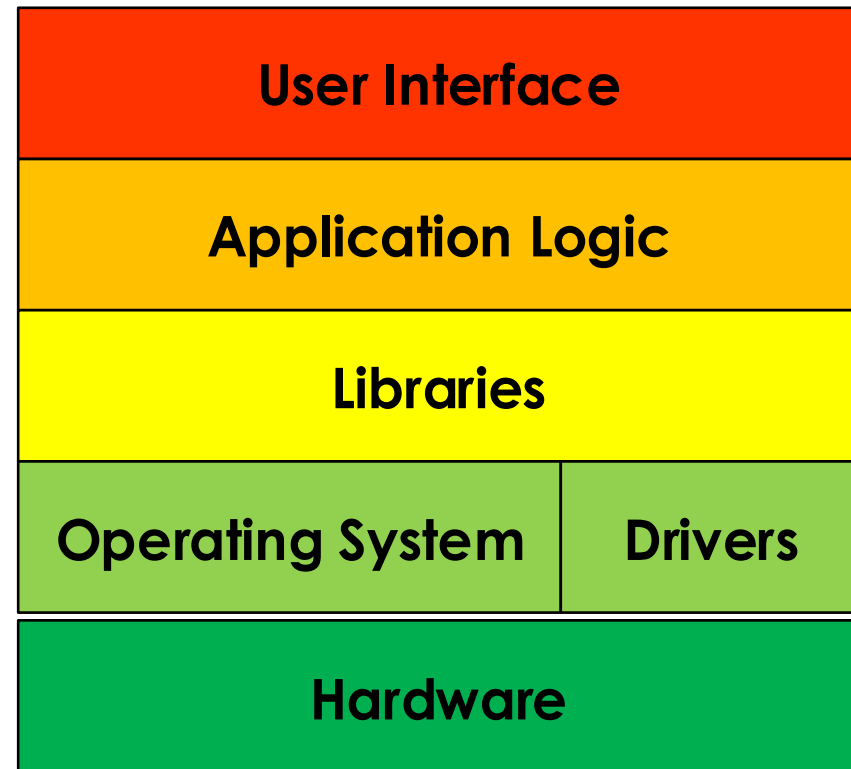
Πιο σημαντικό είναι όμως ο όρος «αφαίρεση» (abstraction) ο οποίος τονίζει ότι κάποιος μπορεί να ασχοληθεί με ένα μόνο layer γνωρίζοντας μόνο την εξωτερική συμπεριφορά των υπολοίπων layers στα οποία βασίζεται. Με τον ίδιο τρόπο που ο οδηγός ενός αυτοκινήτου δεν χρειάζεται να γνωρίζει μηχανολογία.



Αφαιρετικά επίπεδα λειτουργίας

Ως προς τη δομή του software:

- το υψηλότερο είναι η διεπαφή προς τον χρήστη (**user interface**) ή γενικότερα η διεπαφή του με το περιβάλλον
- το χαμηλότερο από αυτά τα επίπεδα είναι το λειτουργικό σύστημα (**operating system**) και οι οδηγοί (**drivers**) που επικοινωνούν απευθείας με το hardware
- όλα τα υπόλοιπα επίπεδα βρίσκονται ανάμεσα σε άλλα επίπεδα αφαίρεσης



Παραδείγματα



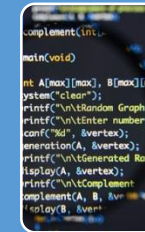
http

- Web σελίδα από τον server στον browser



Αρχεία και φάκελοι

- Το λογισμικό και οι χρήστες ασχολούνται με αρχεία και φακέλους



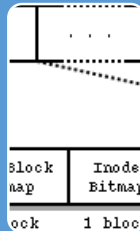
Γλώσσα υψηλού επιπέδου

- Κώδικας σε C



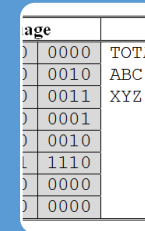
TCP

- Μεταφορά πληροφορίας από Η/Υ σε Η/Υ



Filesystem

- Η δομή της αποθήκευσης των αρχείων στον δίσκο



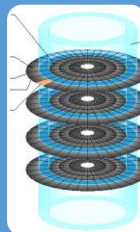
Γλώσσα Μηχανής

- Εντολές «επιπέδου επεξεργαστή»



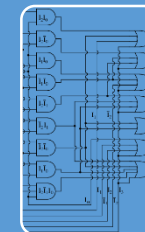
Ethernet

- Αφορά τη φυσική σύνδεση Η/Υ με Η/Υ



Blocks

- Η τμηματοποίηση του δίσκου, ανάλογα με την τεχνολογία του



Λογικά Κυκλώματα

- Λογικές πύλες, Transistors

Ερωτήσεις?

- Διαβάστε τις σημειώσεις, διαβάστε τις διαφάνειες και δείτε τα videos **πριν** ρωτήσετε
- **Συμβουλευτείτε** τη σελίδα ερωταποκρίσεων του μαθήματος

<https://qna.c-programming.allos.gr>

- **Στείλτε** τις ερωτήσεις σας πριν και μετά το μάθημα στο

c-programming-24@allos.gr

- Εάν έχετε **πρόβλημα** με κάποιο κώδικα στείλτε μαζί τον κώδικα και τα μηνύματα λάθους από το CLI ως κείμενα με copy/paste. Εάν θεωρείτε ότι επιπλέον βοηθά και ένα στιγμιότυπο οθόνης, είναι καλοδεχούμενο.
- Επαναλαμβάνουμε : Μην στείλετε ποτέ κώδικα ως εικόνα μας είναι παντελώς άχρηστος!



Γλώσσες προγραμματισμού

Υπάρχουν [πάρα πολλές](#) γλώσσες προγραμματισμού. Κάποιες από αυτές βασίζονται σε άλλες [προγενέστερες](#). Κάποιες από αυτές είναι κατάλληλες για συγκεκριμένες εργασίες ενώ άλλες είναι γενικότερης χρήσης. Πασιγνωστες γλώσσες προγραμματισμού είναι οι:

C/C++	Java	JavaScript	Python	FORTRAN
C#	Kotlin	Perl	Dart	Assembly
Rust	Julia	Pascal	PHP	VisualBasic

Αυτές που αναφέρονται με κόκκινα γράμματα είναι γλώσσες που έχουν παρόμοιο συντακτικό με τη C. Άρα μαθαίνοντας μία από αυτές γνωρίζουμε το βασικό συντακτικό και για τις υπόλοιπες.

Άλλες γλώσσες του Η/Υ

Υπάρχουν επίσης «γλώσσες» για τον υπολογιστή οι οποίες δεν αφορούν εντολές προς τον επεξεργαστή, άρα δεν αποτελούν γλώσσες προγραμματισμού. Τέτοιες είναι οι:

- HTML, XML και JSON, οι οποίες περιγράφουν δεδομένα
- CSS, που περιγράφει την εμφάνιση δεδομένων (κυρίως HTML)
- XSLT, που περιγράφει μετασχηματισμούς δεδομένων και πολλές άλλες.

Γλώσσες που ξεχωρίζουν

Ιδιαίτερη σημασία πρέπει να δοθεί στη γλώσσα Assembly η οποία είναι μία «πιο εμφανίσιμη» μορφή της γλώσσας μηχανής. Αυτή είναι η γλώσσα που «μιλάει» ο ίδιος ο επεξεργαστής. **Οποιοσδήποτε κώδικας εκτελείται σε έναν Η/Υ είναι οπωσδήποτε σε γλώσσα μηχανής (όχι assembly).**

Η γλώσσα μηχανής, άρα και η Assembly είναι **διαφορετική** για κάθε τύπο επεξεργαστή. Αυτό καθιστά αρκετά περίπλοκο το να γνωρίζει κάποιος όλες τις **παραλλαγές**. Επίσης δημιουργεί μεγάλο πρόβλημα στο να δημιουργηθεί ένας **ενιαίος** κώδικας που να λειτουργεί σε διαφορετικούς επεξεργαστές και λειτουργικά συστήματα. Να είναι όπως λέμε cross-platform.

Τη λύση στα παραπάνω προβλήματα, παραμένοντας ταυτόχρονα όσο το δυνατόν πιο κοντά στη λογική του επεξεργαστή, μας τη δίνει η C.

Γιατί η C?

Η C είναι μία γλώσσα που λίγοι προγραμματιστές δεν θα την μάθουν στη ζωή τους. Μερικά σημαντικά χαρακτηριστικά της είναι:

- Είναι πολύ κοντά στην γλώσσα μηχανής, παρέχει τις ίδιες λειτουργίες
- Αυτό την καθιστά διαχρονική γλώσσα, δεν αναμένεται να εγκαταληφθεί
- Πάνω σε αυτή βασίζονται πολλές σύγχρονες γλώσσες (με πιο γνωστές τις Java, JavaScript, PHP)
- Ένας κώδικας σε C που χρησιμοποιεί τις τυπικές βιβλιοθήκες αναμένεται να λειτουργεί απροβλημάτιστα σε οποιοδήποτε λειτουργικό σύστημα, εφόσον γίνει compile γι' αυτό. Είναι δηλαδή cross-platform.
- Λόγω της ευρείας χρήσης της έχει αρκετές έτοιμες βιβλιοθήκες πέρα από τις τυπικές

Υψηλού επιπέδου γλώσσες προγραμματισμού

Οι γλώσσες προγραμματισμού (εκτός της γλώσσας μηχανής και της Assembly) ονομάζονται και υψηλού επιπέδου γλώσσες με την έννοια ότι αποτελούν ένα abstraction layer **πάνω** από τη γλώσσα μηχανής.

Ο κώδικας σε αυτές τις γλώσσες γράφεται σε απλά αρχεία κειμένου που μπορούν να ανοίξουν ακόμα και με το σημειωματάριο (notepad) των Windows.

Για να εκτελεστούν όμως πρέπει με κάποιον τρόπο να μετατραπούν σε γλώσσα μηχανής. Σε αυτό υπάρχουν τρεις λύσεις που χρησιμοποιούνται:

1. [Interpreter](#) (διερμηνέας)
Φορητότητα (Portability) μεταξύ συστημάτων, Προϋποθέτει την εγκατάσταση του στο σύστημα που εκτελείται ο κώδικας – Scripting γλώσσες : JavaScript, Python, PHP, Perl, Lua, Dart
2. Compiler (μεταφραστής)
Ταχύτητα εκτέλεσης, ο τελικός κώδικας είναι αυτόνομος – Compiled γλώσσες : C/C++, FORTRAN, Pascal, Go
3. [JIT/AOT Compiler](#) (Just-in-time/Ahead-of-time compiler)
Μίξη των παραπάνω – Compiled σε bytecode : VisualBasic, C#, Java, Kotlin

Παράδειγμα Scripting

Μία πολύ διαδεδομένη **scripting** γλώσσα είναι η PHP. Ακολουθεί ένα πρόγραμμα που εμφανίζει έναν χαιρετισμό στην οθόνη:

```
<?php
```

Edit

```
echo 'Hello world!';
```

```
sleep(10);
```

Το οποίο εκτελείται (εφόσον είναι εγκατεστημένη η PHP στον υπολογιστή) γράφοντας στη γραμμή εντολών:

```
> php file-name.php
```

Run

Ερωτήσεις?

- Διαβάστε τις σημειώσεις, διαβάστε τις διαφάνειες και δείτε τα videos **πριν** ρωτήσετε
- **Συμβουλευτείτε** τη σελίδα ερωταποκρίσεων του μαθήματος
<https://qna.c-programming.allos.gr>
- **Στείλτε** τις ερωτήσεις σας πριν και μετά το μάθημα στο
c-programming-24@allos.gr
- Εάν έχετε **πρόβλημα** με κάποιο κώδικα στείλτε μαζί τον κώδικα και τα μηνύματα λάθους από το CLI ως κείμενα με copy/paste. Εάν θεωρείτε ότι επιπλέον βοηθά και ένα στιγμιότυπο οθόνης, είναι καλοδεχούμενο.
- Επαναλαμβάνουμε : Μην στείλετε ποτέ κώδικα ως εικόνα μας είναι παντελώς άχρηστος!



Το πρώτο μου πρόγραμμα

Παρουσίαση και επεξήγηση του πρώτου προγράμματος σε απλοποιημένη C ή C* - Γνωριμία με τη γλώσσα

Η απλοποιημένη C ή C*

Στα πρώτα μαθήματα, αντί της πλήρους C, θα μάθουμε μια απλοποιημένη C (που θα συμβολίζουμε με C*) ώστε:

- Να αποκτήσετε αλγοριθμική σκέψη χωρίς περισπασμούς από τις τεχνικές λεπτομέρειες που απαιτεί η C
- Να εισαχθείτε τμηματικά στην πλήρη C, αφού πρώτα έχετε εξοικειωθεί με τα πιο βασικά σημεία της γλώσσας

Σημειώστε πως...

Ό,τι αναφέρουμε για την C, ισχύει και για την απλοποιημένη C, αλλά όχι το αντίστροφο. Για τα στοιχεία της C* θα χρησιμοποιείται το **μωβ** χρώμα, ώστε να ξεχωρίζουν εύκολα και να γνωρίζετε από τώρα ποια είναι αυτά που θα αντικατασταθούν με στοιχεία της C.

Τι είναι ένα πρόγραμμα

- Ένα [πρόγραμμα υπολογιστή](#) είναι μια ακολουθία εντολών που εκτελούνται μία κάθε φορά, η μία μετά την άλλη, τοποθετημένες ώστε να επιτελούν ένα συγκεκριμένο επιθυμητό σκοπό.
- Οι εντολές μπορούν να ομαδοποιούνται μεταξύ τους ώστε να επιτελούν επιμέρους λειτουργίες
- Οι εντολές και οι λειτουργίες συνήθως είναι παραμετρικές
- Το αντικείμενο των εντολών συχνά είναι ο μετασχηματισμός της πληροφορίας
- Πολύ συχνά χρησιμοποιούμε έτοιμες λειτουργίες με σκοπό την ταχύτερη δημιουργία του προγράμματος

() vs [] vs {}



ΠΡΟΣΟΧΗ!

()

Παρενθέσεις
Brackets

προτεραιότητα + συναρτήσεις

[]

Αγκύλες
Square Brackets






πίνακες

{ }

Άγκιστρα
Curly Brackets

ομάδες εντολών

Βασικά στοιχεία του 1^{ου} προγράμματος σε C*

1. Η συνάρτηση `main` 
 2. Εντολές 
 3. Block εντολών 
 4. Συνάρτηση 
 5. Κείμενα 
- ```
function main()
{
 smPrint("Hello world!\n");
 return 0;
}
```

# Περισσότερα στοιχεία του 1<sup>ου</sup> προγράμματος σε C\*

---

- Εκτός από τον πυρήνα της γλώσσας όλες οι συναρτήσεις είναι από εξωτερικές βιβλιοθήκες οι οποίες στην C\* θεωρούμε ότι είναι διαθέσιμες. Σε μεταγενέστερο μάθημα θα δούμε ποιες συναρτήσεις είναι διαθέσιμες σε κάθε βιβλιοθήκη της C και πως τις εισάγουμε στον κώδικά μας.
- Η C έχει ελεύθερη σύνταξη, δηλαδή ανάμεσα στα σύμβολα και τις λέξεις μπορούν να υπάρχουν κενοί χαρακτήρες (space, tab, enter) οπουδήποτε και οσοιδήποτε. Αυτό επιτρέπει την καλή οπτική τοποθέτηση των εντολών και των παραμέτρων τους. Ο μόνος περιορισμός είναι ότι εντός των εισαγωγικών (δηλαδή στα κείμενα) δεν ισχύει η ελεύθερη σύνταξη και δεν επιτρέπεται η αλλαγή γραμμής!

# Τα στοιχεία του κώδικα

Αναγνωριστικά / Τελεστές / Λέξεις κλειδιά / Σταθερές

Είναι λεκτικά δικής μας επιλογής που χρησιμοποιούμε για να ονομάσουμε συνήθως μεταβλητές και συναρτήσεις.

Είναι σύμβολα, τα οποία περιγράφουν μία πράξη μεταξύ «ποσοτήτων» ή οριοθετούν μια περιοχή του κώδικα.

Είναι λέξεις που έχουν συγκεκριμένο ρόλο στη C και δεν μπορούν να χρησιμοποιηθούν ως αναγνωριστικά. Τα αναγνωριστικά της C\* αργότερα θα δούμε πως αλλάζουν για την C.

```
function main()
{
 smPrint("Hello world!\n");
 return 0;
}
```

Παριστάνουν κάποια συγκεκριμένη πληροφορία με σταθερή τιμή.

# Μορφοποιημένη έξοδος

---

Πως εμφανίζω ένα μήνυμα στην C\*

# Μορφοποιημένη έξοδος (`smPrint`)

---

Για την εμφάνιση τιμών στην οθόνη χρησιμοποιείται η `smPrint`. Το κείμενο της πρώτης παραμέτρου της `smPrint` εμφανίζεται στην οθόνη, όχι όμως αυτούσιο. Υπάρχουν δύο περιπτώσεις όπου το κείμενο διαφοροποιείται. Η πρώτη περίπτωση προκύπτει όταν συναντάται ο χαρακτήρας `%` και η δεύτερη όταν συναντάται ο χαρακτήρας `\`.

Έτσι για παράδειγμα όταν η `smPrint` κληθεί ως εξής:

```
smPrint("Please wait for % more seconds\n", 123);
```

**Στο σημείο που υπάρχει ο χαρακτήρας `%`**, σημαίνει ότι εκεί πρέπει να εμφανιστεί μία τιμή (εδώ η 123). Για κάθε χαρακτήρα `%` που υπάρχει στο κείμενο μία ακόμα τιμή πρέπει να δίνεται ως παράμετρος στην `smPrint`.

# Μορφοποιημένη έξοδος (`smPrint`)

---

**Στο σημείο που υπάρχει ο χαρακτήρας `\` (backslash),** σημαίνει ότι ο επόμενος χαρακτήρας αποκτά διαφορετική σημασία. Αυτές οι ακολουθίες χαρακτήρων ονομάζονται **`escape sequence`** και ο χαρακτήρας `\` ονομάζεται **`escape character`**. Εδώ η ειδική ακολουθία χαρακτήρων η `\n` υποδεικνύει μια αλλαγή γραμμής.

Χωρίς την αλλαγή γραμμής η επόμενη `smPrint` θα εμφάνιζε το αποτέλεσμα της δίπλα από αυτό της όποιας `smPrint` είχε τυχόν προηγηθεί. Οι περισσότεροι από αυτούς τους χαρακτήρες αναφέρονται στον πίνακα του παραρτήματος I των σημειώσεων.

Σημειώστε ότι αφού οι χαρακτήρες `%` και `\` έχουν ειδική χρήση και συνδυάζονται με τους χαρακτήρες που τους ακολουθούν, τότε πως θα μπορούσαμε να εμφανίσουμε ένα μήνυμα που να τους περιέχει; Η λύση είναι ότι απλά τους γράφουμε δύο φορές δίπλα-δίπλα δηλαδή ως `%%` ή ως `\\`.

Επίσης παρόμοιο πρόβλημα εμφανίζεται και με τον χαρακτήρα `"` ο οποίος ανοίγει και τερματίζει ένα κείμενο. Για να τον περιλάβουμε σε ένα κείμενο γράφουμε `\"`

Προσοχή! Στη περίπτωση του `%%` θα πρέπει να μην δοθεί κάποια αντίστοιχη παράμετρος στην `smPrint`!

# Ερωτήσεις?

---

- Διαβάστε τις σημειώσεις, διαβάστε τις διαφάνειες και δείτε τα videos **πριν** ρωτήσετε
- **Συμβουλευτείτε** τη σελίδα ερωταποκρίσεων του μαθήματος  
<https://qna.c-programming.allos.gr>
- **Στείλτε** τις ερωτήσεις σας πριν και μετά το μάθημα στο  
[c-programming-24@allos.gr](mailto:c-programming-24@allos.gr)
- Εάν έχετε **πρόβλημα** με κάποιο κώδικα στείλτε τον κώδικα ως κείμενο με copy/paste. Εάν θεωρείτε ότι επιπλέον βοηθά και ένα στιγμιότυπο οθόνης, είναι καλοδεχούμενο.
- Επαναλαμβάνουμε : Μην στείλετε ποτέ κώδικα ως εικόνα μας είναι παντελώς άχρηστος!



# Δεδομένα, Παραστάσεις & Μεταβλητές

---

Πως παριστάνονται στην C



# Μεταβλητές στη C\*

---

Όταν είναι επιθυμητό να αποθηκευθεί στη μνήμη του Η/Υ το αποτέλεσμα μιας πράξης, ένας αριθμός ή κάποια άλλη πληροφορία χρησιμοποιούνται οι μεταβλητές. Αυτές μοιάζουν με τις γνωστές μεταβλητές των μαθηματικών, όμως στους Η/Υ η μεταβλητή συνδέεται με 2 πράγματα (στην C θα δούμε και ένα τρίτο που ονομάζεται τύπος δεδομένων):

- Την ονομασία της
- Την τιμή της μεταβλητής

Αυτό φαίνεται και στη **δήλωση** μιας μεταβλητής. Π.χ.:

`let x = 1234;`

# Ονομασία & Αναγνωριστικά (Identifiers)

---

Με τον όρο Αναγνωριστικά (Identifiers) αποκαλούμε τις ονομασίες στη C, τις οποίες επιλέγει ο προγραμματιστής για να ονοματίσει τις μεταβλητές, τις συναρτήσεις, κ.α. στον κώδικα που αναπτύσσει.

Οι περιορισμοί για τα ονόματα αυτά είναι οι παρακάτω:

- Αποτελούνται από **λατινικούς** χαρακτήρες, **αριθμητικούς** χαρακτήρες και την **κάτω παύλα** `_` (underscore)
- Ο **πρώτος** χαρακτήρας **δεν** μπορεί να είναι αριθμητικός
- Οι **πεζοί** χαρακτήρες με τους αντίστοιχους **κεφαλαίους** θεωρούνται **διαφορετικοί** (είναι case sensitive)
- Δεν επιτρέπεται να συμπίπτουν με **δεσμευμένες λέξεις** της C
- Ενδεχομένως (ανάλογα τον compiler) να υπάρχει ένα όριο στο πόσο μεγάλο μπορεί να είναι ένα αναγνωριστικό

Έτσι το `someName1` , `some_name_1` και το `Somename1` είναι αποδεκτά αναγνωριστικά και διαφορετικά μεταξύ τους, ενώ το `1_name` δεν είναι αποδεκτό επειδή ξεκινά με αριθμητικό χαρακτήρα.

# Δεσμευμένες λέξεις στη C

Στη C και την C\* κάποιες λέξεις χρησιμοποιούνται από την ίδια τη γλώσσα. Ονομάζονται δεσμευμένες λέξεις. Αυτές είναι:

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

Υπόμνημα : Της C\* – Κοινές – Της C – Της C εκτός ύλης

Αυτές, λοιπόν, είναι οι λέξεις που δεν μπορούν να χρησιμοποιηθούν αυτούσιες ως identifiers (αναγνωριστικά).

# Αριθμητικές τιμές στη C\*

- Οι αριθμητικές τιμές στη C\* είναι πραγματικές τιμές. Η αναπαράσταση τους ονομάζεται **κινήτης υποδιαστολής**, δηλαδή πραγματικός αριθμός.
- Υπάρχουν τρεις τρόποι αναπαράστασης

- Ακέραια γραφή:

*[πρόσημο] ψηφία (0-9)*

όπου το πρόσημο είναι προαιρετικό ενώ υπάρχουν τα ψηφία του ακέραιου μέρους

Για την C\* απαγορεύεται να ξεκινά από 0 ο αριθμός.

- Απλή γραφή πραγματικών (κατά τα γνωστά από τα μαθηματικά):

*[πρόσημο] ψηφία (0-9) . ψηφία (0-9)*

υποχρεωτικά υπάρχουν η υποδιαστολή (που είναι τελεία, όχι κόμμα) και ένα ψηφίο τουλάχιστον (είτε στο ακέραιο, είτε στο δεκαδικό μέρος)

- Επιστημονική ή εκθετική γραφή (αντιστοιχεί στη μορφή **αριθμός** x 10<sup>v</sup>):

*[πρόσημο] ψηφία (0-9) . ψηφία (0-9) E [πρόσημο] ψηφία (0-9)*

υποχρεωτικά υπάρχουν το **e** (ή **E**), στα αριστερά του ο αριθμός γράφεται σε απλή γραφή και στα δεξιά ο εκθέτης ως προσημασμένος ακέραιος. Σε όλη την έκταση του αριθμού δεν υπάρχουν καθόλου κενά.

| Σωστό      | Λάθος     |
|------------|-----------|
| 1234.5678  | 1234,5678 |
| -1000.0001 | 12 .34    |
| +123.345   | 11.22.33  |
| -.34       | .         |
| 123.       |           |

| Σωστό       | Λάθος                  |
|-------------|------------------------|
| -12.34e10   | 12e 3                  |
| 13.1002e-30 | 1e8.2                  |
| 12.32E+4    | 12.34·10 <sup>+4</sup> |
| .132E+2     |                        |
| 1e3         |                        |

## ΣΗΜΕΙΩΣΗ

Οι αγκύλες εδώ συμβολίζουν ό,τι είναι προαιρετικό (π.χ. το πρόσημο για θετικούς αριθμούς). Δεν πληκτρολογούνται και δεν εμφανίζονται πουθενά. Δείτε και τα παραδείγματα δεξιά!

# Αριθμητικές πράξεις μεταξύ των τιμών

Με δεδομένες τις αριθμητικές ποσότητες, το επόμενο βήμα είναι η πραγματοποίηση αριθμητικών πράξεων μεταξύ τους.

Στη C είναι διαθέσιμα τα σύμβολα των 4<sup>ων</sup> πράξεων (+, -, \*, /) αλλά και το σύμβολο % το οποίο παριστάνει τον υπολογισμό του υπολοίπου της ακέραιας διαίρεσης.

Η προτεραιότητα των πράξεων είναι η γνωστή από τα μαθηματικά. Το % έχει την ίδια προτεραιότητα με τη διαίρεση.

Όπου χρειάζεται να τροποποιηθεί η προτεραιότητα χρησιμοποιούνται αποκλειστικά και μόνο παρενθέσεις. Οι πράξεις πραγματοποιούνται όπως συναντώνται από τα αριστερά προς τα δεξιά.

## Δοκιμάστε

$$5 + 3$$

$$12.34 * 1e-2$$

$$44 \% 5$$

$$12 / 3 / 4$$

$$(1 + 8) / (5 + 4)$$

$$5 / 8$$

$$4.0 / 2.0 * 2.0$$

$$4.4 \% 3.3$$

## Προτεραιότητα

( )

+ - (ως πρόσημα)

\* / %

+ - (ως πράξεις)

# Δήλωση και χρήση μιας μεταβλητής τοπικής εμβέλειας

---

Οι μεταβλητές δηλώνονται μέσα σε άγκιστρα, δηλαδή σε block εντολών. Από το σημείο της δήλωσής τους και μέχρι το τέλος του block μπορούν να χρησιμοποιηθούν, είτε δίνοντας τιμή σε αυτές, είτε χρησιμοποιώντας τις μεταβλητές μέσα σε παραστάσεις όπως και τις απλές τιμές. Μετά το τέλος του block η μεταβλητή διαγράφεται και δεν μπορεί να χρησιμοποιηθεί πλέον.

Πριν την πρώτη χρήση κάθε μεταβλητής θα πρέπει ο κώδικας να της αναθέτει κάποια τιμή. Αυτό συνήθως γίνεται κατά τη δήλωση. Εάν δεν ανατεθεί αρχική τιμή, η μεταβλητή θα έχει τυχαία τιμή και όχι μηδενική. Εκτός της δήλωσης η ανάθεση γίνεται με τη μορφή:

```
myVariable = 0.0;
```

Φυσικά εκτός από συγκεκριμένη τιμή, μπορεί να δοθεί και ολόκληρη παράσταση στα δεξιά του =. Επίσης προσέξτε ότι το = έχει την έννοια της ανάθεσης και όχι της εξίσωσης, οπότε το παρακάτω είναι σωστή σύνταξη και αυξάνει την τιμή της μεταβλητής κατά A.

```
myVariable = myVariable + A;
```

# Ερωτήσεις?

---

- Διαβάστε τις σημειώσεις, διαβάστε τις διαφάνειες και δείτε τα videos **πριν** ρωτήσετε
- **Συμβουλευτείτε** τη σελίδα ερωταποκρίσεων του μαθήματος  
<https://qna.c-programming.allos.gr>
- **Στείλτε** τις ερωτήσεις σας πριν και μετά το μάθημα στο  
[c-programming-24@allos.gr](mailto:c-programming-24@allos.gr)
- Εάν έχετε **πρόβλημα** με κάποιο κώδικα στείλτε τον κώδικα ως κείμενο με copy/paste. Εάν θεωρείτε ότι επιπλέον βοηθά και ένα στιγμιότυπο οθόνης, είναι καλοδεχούμενο.
- Επαναλαμβάνουμε : Μην στείλετε ποτέ κώδικα ως εικόνα μας είναι παντελώς άχρηστος!



# Παράδειγμα

Ένας κώδικας που υπολογίζει εμβαδόν του διπλανού σχήματος.

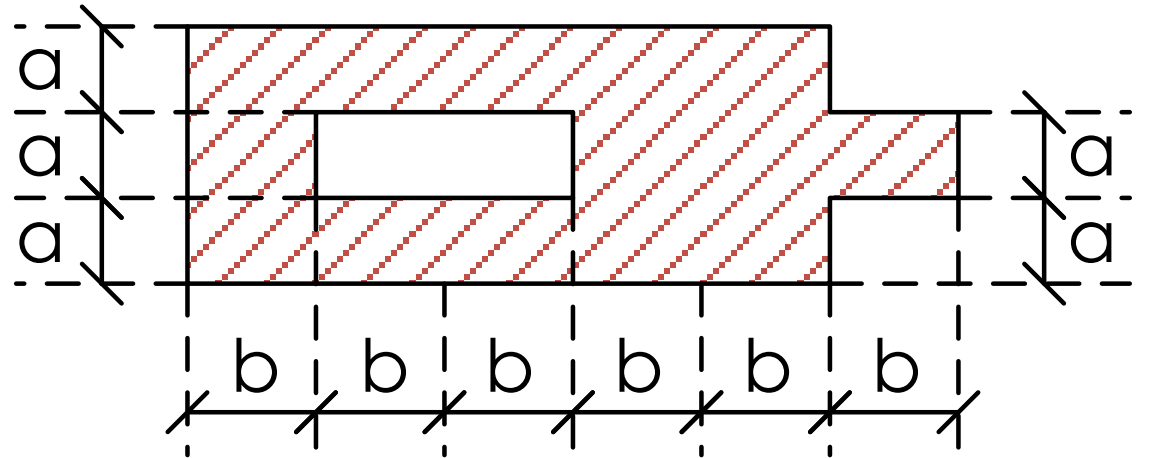
```
function main() {
 let a = 1.2, b = 2.1;

 let vSide = 3*a;
 let hSide = 5*b;

 let E = vSide * hSide + a*b - a * (2*b);

 smPrint("To envado einai %\n", E);

 return 0;
}
```





# Ερωτήσεις?

---

- Διαβάστε τις σημειώσεις, διαβάστε τις διαφάνειες και δείτε τα videos **πριν** ρωτήσετε
- **Συμβουλευτείτε** τη σελίδα ερωταποκρίσεων του μαθήματος  
<https://qna.c-programming.allos.gr>
- **Στείλτε** τις ερωτήσεις σας πριν και μετά το μάθημα στο  
[c-programming-24@allos.gr](mailto:c-programming-24@allos.gr)
- Εάν έχετε **πρόβλημα** με κάποιο κώδικα στείλτε τον κώδικα ως κείμενο με copy/paste. Εάν θεωρείτε ότι επιπλέον βοηθά και ένα στιγμιότυπο οθόνης, είναι καλοδεχούμενο.
- Επαναλαμβάνουμε : Μην στείλετε ποτέ κώδικα ως εικόνα μας είναι παντελώς άχρηστος!



# Ροή εκτέλεσης προγράμματος

---

Πως εκτελούνται οι εντολές ενός προγράμματος σε C

# Ροή εκτέλεσης εντολών

---

Σε όλες τις τυπικές γλώσσες προγραμματισμού, η εκτέλεση των εντολών γίνεται **σειριακά** όπως διαβάζουμε, δηλαδή από πάνω προς τα κάτω και στην ίδια γραμμή, από αριστερά προς τα δεξιά.

Ο όρος σειριακά σημαίνει ότι οι εντολές εκτελούνται:

- μία κάθε φορά
- η μία αμέσως μετά την άλλη
- κάθε στιγμή εκτελείται οπωσδήποτε κάποια εντολή

Τα παραπάνω, ορίζουν μία «ροή» που ονομάζουμε ροή εκτέλεσης του προγράμματος.

**Προσοχή!** Στη C, δεν επιτρέπεται εντολή γραμμένη έξω από κάποια συνάρτηση. Η συνάρτηση αποτελεί βασικό δομικό στοιχείο της γλώσσας.

# Ροή εκτέλεσης εντολών

Η εκτέλεση των εντολών μιας συνάρτησης ξεκινά με την εντολή που ακολουθεί το αρχικό άγκιστρό { της, και τελειώνει είτε με την εντολή που υπάρχει ακριβώς πριν το τελικό άγκιστρό της }, είτε με την εντολή **return** που παρουσιάζεται σε επόμενη διαφάνεια.

```
function printOranges (oranges) {
 smPrint ("Exw % portokalia", oranges);
 return;
}

function main () {
 printOranges (5);
 return 0;
}
```

Η εκτέλεση ενός προγράμματος ξεκινά με την εκτέλεση της συνάρτησης **main** και όπου καλείται μία συνάρτηση, η ροή της εκτέλεσης του προγράμματος προσωρινά μεταφέρεται στις εντολές της, μέχρι και την ολοκλήρωσή της. Κατόπιν η εκτέλεση συνεχίζεται με την εντολή που ακολουθούσε την κλήση της.

Η εκτέλεση ενός προγράμματος ολοκληρώνεται όταν ολοκληρωθεί η εκτέλεση της **main**.

# Κλήση συναρτήσεων

---

Πως καλείται μια συνάρτηση

# Μια ήδη γνωστή συνάρτηση

Έχουμε ήδη συναντήσει τη συνάρτηση **smPrint**. Αυτή, όπως και όλες οι συναρτήσεις **καλούνται** (=χρησιμοποιούνται) με τον ακόλουθο τρόπο.

```
smPrint ("%" , x);
```

Δηλαδή:

- αρχικά γράφεται το όνομα της συνάρτησης
- ακολουθούν οι παρενθέσεις και
- μέσα τους γράφονται τα **ορίσματα** χωρισμένα με κόμμα.

Σημειώστε εδώ τον όρο ορίσματα. Έτσι ονομάζουμε τις τιμές που δίνουμε (ή περνάμε) σε μία συνάρτηση, καθώς θα τον συναντήσουμε λίγο αργότερα.

# Μια άλλη συνάρτηση

---

Μια άλλη συνάρτηση είναι η **fabs** η οποία υπολογίζει την απόλυτη τιμή του ορίσματος που της δίνεται. Αυτή, επειδή έχει αριθμητικό αποτέλεσμα καλείται ως εξής:

```
newValue = fabs(someValue);
```

ή

```
newValue = someOtherValue + fabs(someValue);
```

Δηλαδή **το αποτέλεσμα** της συνάρτησης λειτουργεί όπως μία σταθερή τιμή ή μία μεταβλητή στην θέση της. Ακριβώς όπως και η αντικατάσταση στα μαθηματικά.

Σημειώστε ότι: Η συγκεκριμένη συνάρτηση φαίνεται «απλά να υπάρχει» πράγμα το οποίο δεν ισχύει ούτε στη C, ούτε στην **C\***. Σε αυτό θα αναφερθούμε όταν δούμε πως από την **C\*** πάμε στη C.

# Μερικές “έτοιμες” συναρτήσεις

---

Δείτε μερικές συναρτήσεις που μας είναι διαθέσιμες στη C και στην C\*



# Μερικές έτοιμες συναρτήσεις (math)

Η βιβλιοθήκη math μεταξύ άλλων περιλαμβάνει και τις σταθερές και συναρτήσεις του πίνακα.

Περισσότερες πληροφορίες θα δείτε στη σχετική ενότητα των σημειώσεων αλλά και online [εδώ](#).

| Σταθερές            |                                                                                          |
|---------------------|------------------------------------------------------------------------------------------|
| M_E                 | Η τιμή του e ( 2.718... )                                                                |
| M_PI                | Η τιμή του π ( 3.14159... )                                                              |
| M_SQRT2             | Η τιμή του $\sqrt{2}$ ( 1.414... )                                                       |
| M_LN2               | Η τιμή του $\ln 2$ ( 0.6931... )                                                         |
| Γενικά Μαθηματικά   |                                                                                          |
| function fabs(x)    | Επιστρέφει την απόλυτη τιμή του x                                                        |
| function sqrt(x)    | Επιστρέφει την τετραγωνική ρίζα του x                                                    |
| function pow(x, y)  | Επιστρέφει το x υψωμένο στο y                                                            |
| function log(x)     | Ο φυσικός λογάριθμος του x                                                               |
| function log10(x)   | Ο λογάριθμος του x ως προς το 10.                                                        |
| function exp(x)     | Ο αριθμός e υψωμένος στην x                                                              |
| function fmod(x, y) | Το υπόλοιπο της διαίρεσης x δια y                                                        |
| Στρογγυλοποίηση     |                                                                                          |
| function ceil(x)    | Στρογγυλοποιεί την τιμή του x στο μικρότερο ακέραιο που είναι μεγαλύτερος ή ίσος με το x |
| function floor(x)   | Στρογγυλοποιεί την τιμή του x στο μεγαλύτερο ακέραιο που είναι μικρότερος ή ίσος με το x |

| Τριγωνομετρικές συναρτήσεις             |                                                                                                                                                                                           |
|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| function sin(x)                         | Το ημίτονο της x, όταν η x είναι εκφρασμένη σε ακτίνια (rad)                                                                                                                              |
| function cos(x)                         | Το συνημίτονο της x, όταν η x είναι εκφρασμένη σε ακτίνια (rad)                                                                                                                           |
| function tan(x)                         | Η εφαπτομένη της x, όταν η x είναι εκφρασμένη σε ακτίνια (rad)                                                                                                                            |
| Αντίστροφες τριγωνομετρικές συναρτήσεις |                                                                                                                                                                                           |
| function asin(x)                        | Το τόξο ημιτόνου του x, εκφρασμένο σε ακτίνια (rad) στο διάστημα $[-\pi/2, \pi/2]$                                                                                                        |
| function acos(x)                        | Το τόξο συνημιτόνου του x, εκφρασμένο σε ακτίνια (rad) στο διάστημα $[0, \pi]$                                                                                                            |
| function atan(x)                        | Η τόξο εφαπτομένης του x, εκφρασμένη σε ακτίνια (rad) στο διάστημα $[-\pi/2, \pi/2]$                                                                                                      |
| function atan2(y, x)                    | Η τόξο εφαπτομένης που προκύπτει από τη διαίρεση y/x, λαμβάνοντας υπόψη και τα πρόσημα ώστε να υπολογιστεί η γωνία στο σωστό τεταρτημόριο. Επιστρέφεται μία τιμή στο διάστημα $[0, 2\pi]$ |

# Ερωτήσεις?

---

- Διαβάστε τις σημειώσεις, διαβάστε τις διαφάνειες και δείτε τα videos **πριν** ρωτήσετε
- **Συμβουλευτείτε** τη σελίδα ερωταποκρίσεων του μαθήματος  
<https://qna.c-programming.allos.gr>
- **Στείλτε** τις ερωτήσεις σας πριν και μετά το μάθημα στο  
[c-programming-24@allos.gr](mailto:c-programming-24@allos.gr)
- Εάν έχετε **πρόβλημα** με κάποιο κώδικα στείλτε τον κώδικα ως κείμενο με copy/paste. Εάν θεωρείτε ότι επιπλέον βοηθά και ένα στιγμιότυπο οθόνης, είναι καλοδεχούμενο.
- Επαναλαμβάνουμε : Μην στείλετε ποτέ κώδικα ως εικόνα μας είναι παντελώς άχρηστος!

