



# Προγραμματισμός Ι

## ΜΕΤΑΒΛΗΤΕΣ-ΣΤΑΘΕΡΕΣ-Ι/Ο ΚΟΝΣΟΛΑΣ

Τμήμα Πληροφορικής & Επικοινωνιών  
Δρ. Θεόδωρος Γ. Λάντζος  
<http://www.teiser.gr/icd/staff/lantzog>  
lantzog@teiser.gr

# Κανόνες Ομαλής Λειτουργίας

- Ερχόμαστε στην ώρα μας
- Δεν καπνίζουμε και τρώμε εντός της αίθουσας



- Επιτρέπεται το νερό, τα αναψυκτικά και ο καφές με την προϋπόθεση να μην λερώνουμε το χώρο και πετάμε τα σκουπίδια εκτός των καλαθιών.
- Κινητά αθόρυβα και μόνον σε περίπτωση άμεσης ανάγκης.



- Σε περίπτωση συναγερμού, αποχωρούμε από την αίθουσα για το σημείο συγκέντρωσης σταδιακά, χωρίς πανικό και πιέσεις.
- Η αποχώρηση από την αίθουσα για οποιοδήποτε λόγο πρέπει να γίνεται με ηρεμία και χωρίς να διακόπτουμε και ενοχλούμε τους συναδέλφους



# Συναντήσεις και Forum

---

- Οι διαλέξεις θα διεξάγονται κάθε Τρίτη 14-16 στην αίθουσα 103 της ΣΤΕΦ
- Κάθε Πέμπτη 15-17 Θα υπάρχει ανοιχτό discussion forum προγραμματισμού στις επάνω αίθουσες της ΣΤΕΦ
- Ανακοινώσεις στην σελίδα μου  
<http://www.teiser.gr/icd/staff/lantzoz>
- Ερωτήσεις δια μέσο email οι οποίες όμως θα απατούνται και αναλύονται στο forum.  
[lantzoz@teiser.gr](mailto:lantzoz@teiser.gr)

# Οδηγός Επιτυχίας



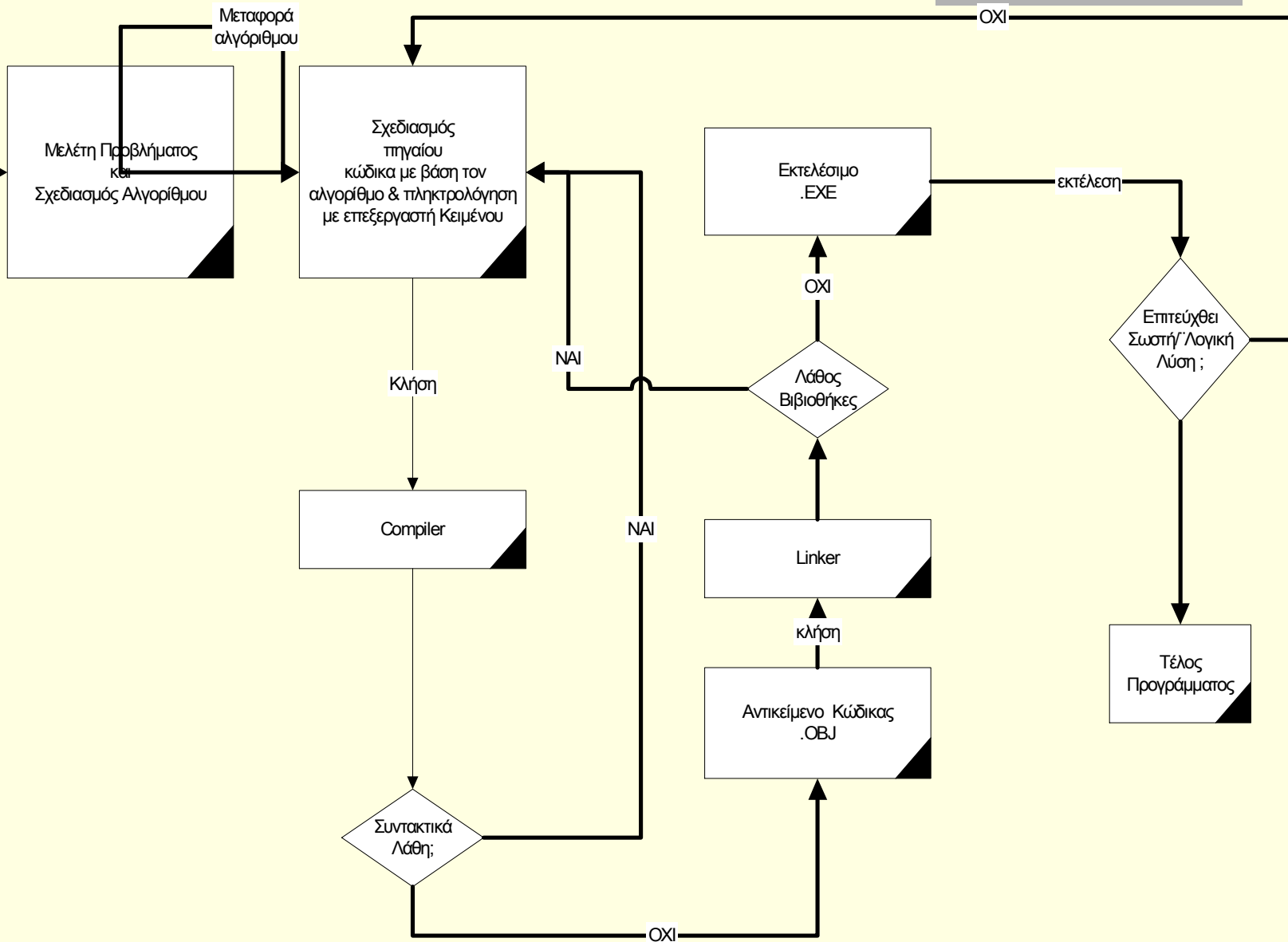
- Στην τάξη θα αναπτυχθούν και θα αναλυθούν πολλά θέματα προγραμματισμού εκτενέστερα τα οποία είναι πολύ δύσκολα στην κατανόηση τους με ατομική μελέτη.
- Συνήθως το υλικό αυτό αποτελεί αντικείμενο εξετάσεων
- Εάν χάσετε την παρουσίαση το υλικό δεν θα επαναλαμβάνετε μέσω email ή γραφείου.
- Η εκμάθηση προγραμματισμού Η/Υ απαιτεί συνεχή μάθηση και πειραματισμό. Μαθαίνουμε από την συνεχή επίλυση προβλημάτων και με την βοήθεια των καθηγητών.
- Η εργαστηριακή ομάδα καθηγητών θα είναι σε συνεχή παρακολούθηση της εξέλιξης του μαθήματος.
- 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.

# Πώς δημιουργούμε πρόγραμμα Η/Υ;

---

1. Ανάλυση του προβλήματος
2. Επινόηση & Σχεδιασμός λύσης (Αλγόριθμο)
3. Μεταφορά Αλγορίθμου σε Κώδικα (πηγαίο)
4. Μεταγλώττιση πηγαίου κώδικα
5. Παραγωγή του αντικείμενου κώδικα
6. Σύνθεση του αντικείμενου κώδικα
7. Εκτέλεση προγράμματος

# Στάδια Υλοποίησης Προγράμματος



# Βασικά στοιχεία προγράμματος

---

```
/*  
This program prints out the sentence "This is a test"  
***/  
  
#include <stdio.h>  
  
void main()  
{  
    printf("This is a test.\n");  
}
```

Το πρώτο πρόγραμμα

# Δομή προγράμματος

---

- Σχόλια πρόλογου [προαιρετικό]
- Οδηγίες αναφοράς σε βιβλιοθήκες [προαιρετικό]
- Συναρτήσεις [προαιρετικό]
- Κύρια Συνάρτηση
  - {
    - Δηλώσεις [προαιρετικό]
    - Σώμα προγράμματος
  - }



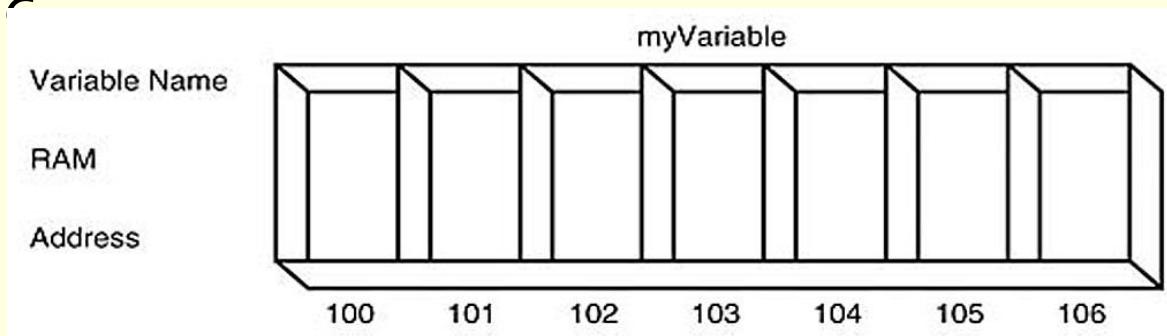
# Μεταβλητές

---

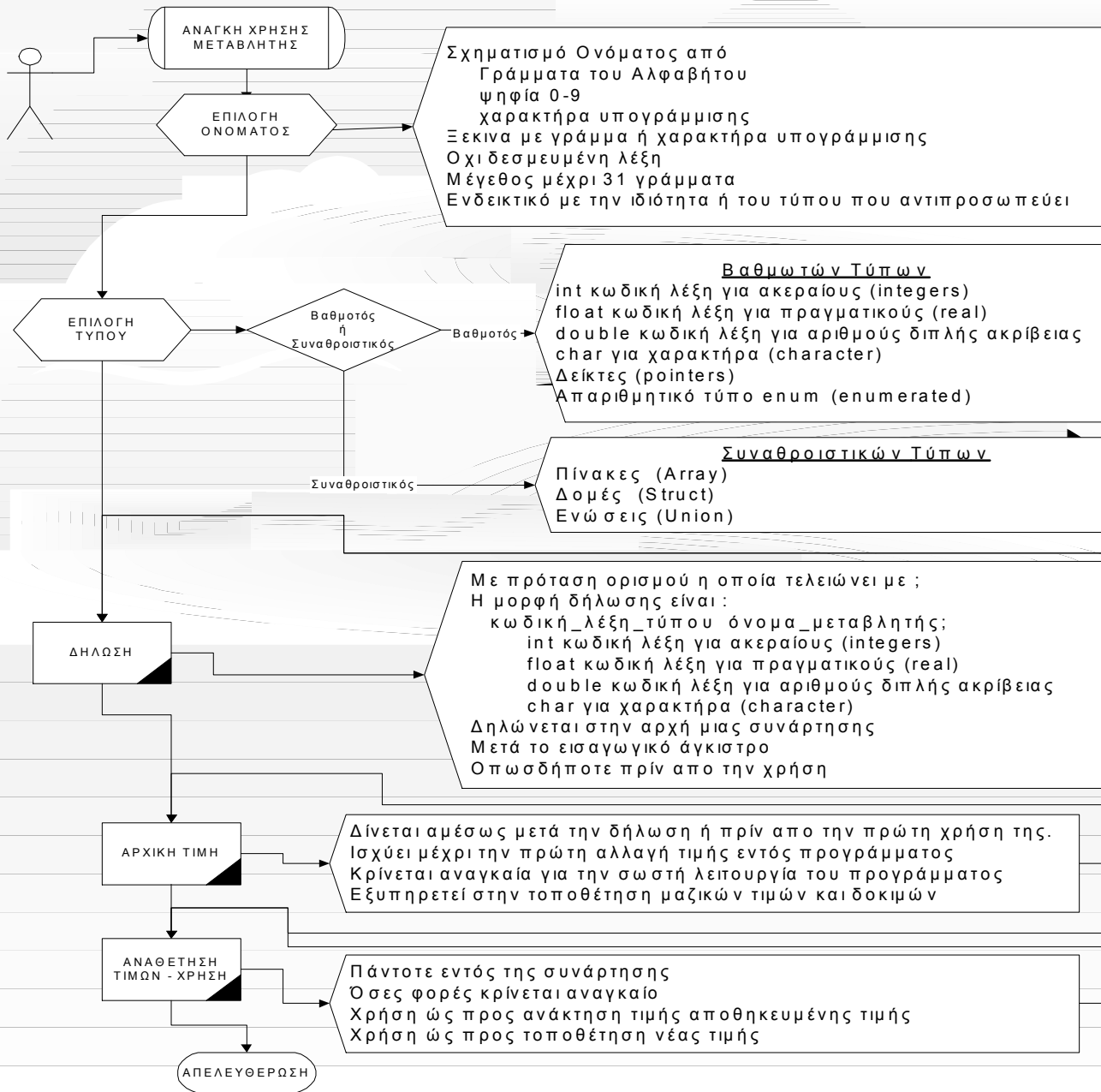
- Διαχείριση πληροφορίας σε μορφή αριθμητικών δεδομένων, γραμμάτων ή ακολουθίας γραμμάτων
- Τα δεδομένα αποθηκεύονται στην μνήμη και χρειάζονται ένα μέσο για να κληθούν από τα προγράμματα
- Η λύση στο πρόβλημα αναφορά και διαχείρισης των δεδομένων από την μνήμη δόθηκε με την έννοια της μεταβλητής

# Τι είναι μεταβλητή;

- Μεταβλητή είναι μια περιοχή στην μνήμη του υπολογιστή στην οποία μπορούμε να αποθηκεύσουμε μια τιμή και να την ανακτήσουμε. Το όνομα μιας μεταβλητής είναι άμεσα συνδεδεμένο με την διεύθυνση με την οποία είναι αποθηκευμένο το δεδομένο.
- Την μνήμη του υπολογιστή μπορούμε να την φανταστούμε σαν μια σειρά άδεια κελιά τα οποία είναι στοιχισμένα σε μια διαδοχική σειρά. Το κάθε κελί ή θέση μνήμης αριθμείτε σειριακά. Αυτοί οι αριθμοί είναι γνωστοί ως διευθύνσεις μνήμη.



# ΚΥΚΛΟΣ ΖΩΗΣ ΜΕΤΑΒΛΗΤΗΣ



# Ο τύπος ακεραίου

- Χρησιμοποιείται για να παραστήσει ακεραίους αριθμούς
- Η δήλωση μιας μεταβλητής ακεραίου τύπου έχει τον φορμαλισμό
  - `int όνομα_μεταβλητής;` Π.χ. `int num;`
- Δηλώνεται πάντα στην αρχή μιας συνάρτησης και αμέσως μετά το {
- Το μέγεθος μια μεταβλητής ακεραίου τύπου είναι 16 bits και το σύνολο των δυνατών τιμών είναι  $2^{16} = 65536$  με εύρος τιμών -32768 έως +32768
- Προσδιοριστές
  - `long int` εξασφαλίζει αποθηκευτικό χώρο 32 bits
  - `unsigned int` δεσμεύει τιμές χωρίς πρόσημο 0-65535
  - `short int` τουλάχιστον 16 bits

Περιβάλλον Windows XP οι ακέραιοι έχουν 4 bytes με σύνολο δυνατών τιμών  $2^{32} = 4\ 294\ 967\ 296$

# Τύποι πραγματικών αριθμών

- Αναπαράσταση πραγματικών αριθμών
- Για αριθμούς που διαθέτουν κλασματικό μέρος
- Η δήλωση μιας μεταβλητής πραγματικού τύπου απλής ακρίβειας έχει τον φορμαλισμό
  - float όνομα\_μεταβλητής; Π.χ. float num;
  - Μέγεθος 32 bits
- διπλής ακρίβειας έχει τον φορμαλισμό
  - double όνομα\_μεταβλητής; πχ. double plank;
  - Μέγεθος 64 bits
- Προσδιοριστής
  - long πριν από τον τύπο double για δήλωση μεταβλητής κινητής υποδιαστολής εκτεταμένης ακρίβειας

πχ. long double plank



# Ο τύπος χαρακτήρα

- Ο τύπος χαρακτήρα παριστάνει απλούς χαρακτήρες του αλφάβητου της γλώσσας
- Η δήλωση μιας μεταβλητής χαρακτήρα έχει τον φορμαλισμό
  - `char όνομα_μεταβλητής; Π.χ. char choice;`
- 1byte για την αποθήκευση της τιμής μίας μεταβλητής χαρακτήρα
- Χαρακτήρες δια μέσο του κώδικα ASCII (American Standard Code for Information Interchange). Χαρακτήρας σε αντιστοίχιση με οκταψήφιο δυαδικό αριθμό

# Μηνύματα στην οθόνη

- Πως; -> Με την χρήση της εντολής `printf()`;
- Πού ; -> Εντός του σώματος του προγράμματος
- Απλή μορφή μηνυμάτων :
  - Δέχεται ένα όρισμα το οποίο και αποστέλλει στην έξοδο. Το όρισμα περικλείεται από “ μήνυμα ”  
πχ. `printf(“HELLO WORLD “);`
- Μορφοποιημένα μηνύματα
  - Δέχεται ένα όρισμα το οποίο και αποστέλλει στην έξοδο μορφοποιώντας το ανάλογα με τον οδηγό εντολών (μη εκτυπωμένους χαρακτήρες) που εσωκλείονται στο μήνυμα. Το μήνυμα περικλείεται από “ “  
Πχ. `printf(“My first name is : \t Theo \n Family Name is :\t Lantz0s”);`



# Ειδικοί χαρακτήρες μορφοποίησης μηνυμάτων (μη εκτυπούμενοι χαρακτήρες)

Χαρακτήρας	Ακολουθία
Συναγερμός	\a
Οπισθοχώρηση	\b
Αλλαγή σελίδας	\f
Νέα γραμμή	\n
Επαναφορά κεφαλής	\r
Οριζόντιος Στηλοθέτης	\t
Κατακόρυφος Στηλοθέτης	\v
Πλάγια γραμμή	\\
Λατινικό ερωτηματικό	\?
Μονό εισαγωγικό	\;
Διπλό εισαγωγικό	\”
Οκταδικός αριθμός	\ooo
Δεκαεξαδικός αριθμός	\xhhh

# Τιμές μεταβλητών και εμφάνιση

- Με την χρήση της εντολής `printf()`;
- Σύνθετη Μορφή  
`printf(" μήνυμα ", μεταβλητή1, μεταβλητή2, ..., μεταβλητή_n);`
- Για κάθε μεταβλητή πρέπει να έχουμε και έναν προσδιοριστή εντός του μηνύματος.
- Το σύνολο μεταβλητών και προσδιοριστών θα πρέπει να είναι ίσο
- Οι προσδιοριστές τοποθετούνται εντός του μηνύματος, με το σύμβολο `%` και δίπλα έναν χαρακτήρα ανάλογα με τον τύπο της μεταβλητής
  - Βλέπε επόμενο πίνακα
- Στους ακεραίους και πραγματικούς μπορούμε να ορίσουμε το πλήθος των ψηφίων που θα εκτυπωθούν τοποθετώντας τον επιθυμητό αριθμό ανάμεσα στο `%` και τον χαρακτήρα με την μορφή `a.b` π.χ `%3.2f`, `%4d`

# Προσδιοριστές μεταβλητών για εμφάνιση και ανάγνωση τιμών

Είδος Τιμής	Τύπος Μεταβλητής	Προσδιοριστής
Χαρακτήρας	char	%c
Σύνολο χαρακτήρων	string	%s
Ακέραιος δεκαδική	int	%d
Ακέραιος Οκταδική	int	%o
Ακέραιος Δεκαεξαδική	int	%h
Ακέραιος μεγάλου μεγέθους	long int	%ld
Ακέραιος μικρού μεγέθους	short int	%sd
Ακέραιος χωρίς πρόσημο	unsigned	%ud
Πραγματικός Απλής ακρίβειας	float	%f
Πραγματικός Διπλής Ακρίβειας	double	%f

# Εισαγωγή τιμών σε μεταβλητές

---

- Με την χρήση της `scanf()`;
- Εντός του σώματος του προγράμματος
- Με μορφή  
`scanf("μήνυμα", μεταβλητή 1, .., .., μεταβλητή N);`
- Στο μήνυμα περιγράφουμε τους προσδιοριστές των μεταβλητών με την σωστή σειρά και με βάση των πίνακα προσδιοριστών
- Το σύνολο προσδιοριστών και μεταβλητών θα πρέπει να είναι ίσο
- Στην εντολή `scanf` πριν από κάθε μεταβλητή τοποθετούμε τον τελεστή διεύθυνσης & όχι στην `printf`

# Συναρτήσεις getch, getche

---

- Περιγράφονται στην βιβλιοθήκη `conio.h`
- Η `getche` διαβάζει έναν χαρακτήρα από την κύρια είσοδο (πληκτρολόγιο) και στην συνέχεια επιστρέφει την τιμή του, εμφανίζοντας στην οθόνη το πλήκτρο που πατήθηκε
- Η `getch` διαφέρει στο ότι δεν εμφανίζει τον πληκτρολογηθέντα χαρακτήρα στην οθόνη

# getchar

---

- Περιγράφεται στην βιβλιοθήκη `stdio.h` με πρωτότυπο
  - `int getchar(void)`
- Διαβάζει ένα χαρακτήρα από την κύρια είσοδο και τον επιστρέφει στο πρόγραμμα

# putchar

---

- Ορίζεται στην `stdio.h`
- Δέχεται έναν χαρακτήρα σαν παράμετρο και τον εμφανίζει στην τρέχουσα θέση του δρομέα