

Εισαγωγή στον προγραμματισμό



**Τμήμα Πληροφορικής & Επικοινωνιών
ΤΕΙ Σερρών
Εργαστήριο 2**



Δομή Προγράμματος

- Όλα τα προγράμματα που γράψαμε έχουν λίγο-πολύ την ακόλουθη μορφή:

```
Κάποιος κώδικας εδώ
..... main (.....)
{
    Κάποιος κώδικας εδώ
}
```



Η συνάρτηση main

- Όλα τα προγράμματα της C για να εκτελεστούν περιέχουν κώδικα ο οποίος βρίσκεται ανάμεσα σε δύο αγκύλες, κάτω από την ονομασία main.
- Αυτό είναι και το κύριο κομμάτι του προγράμματος που θα εκτελεστεί.
- Συναρτήσεις έχετε γνωρίσει πολλές στα μαθηματικά. Μια από τις πιο γνωστές είναι η συνάρτηση που υπολογίζει την τετραγωνική ρίζα: δίνουμε έναν αριθμό και επιστρέφει την τετραγωνική του ρίζα.
- Στα προγράμματα που θα γράψετε εσείς, αρχικά δε θα δίνετε κανένα δεδομένο και δε χρειάζεται να επιστρέψετε κανένα δεδομένο. Αυτή είναι και η πιο απλή μορφή συνάρτησης που μπορείτε να έχετε.



Η πιο απλή μορφή της main

- Η πιο απλή μορφή επομένως που μπορεί να έχει η main είναι η ακόλουθη:

```
void main(void)  
{  
    Κάποιος κώδικας εδώ  
}
```



Η συνάρτηση printf

- Στη C υπάρχουν ήδη κάποια έτοιμα «προγράμματα» που μπορείτε να χρησιμοποιήσετε ανάλογα με τις ανάγκες σας.
- Αν θέλετε να εμφανίσετε ένα μήνυμα στην οθόνη μπορείτε να χρησιμοποιήσετε την printf.
- Και η printf είναι μια συνάρτηση! Της δίνετε κάποια δεδομένα (το μήνυμα που θέλετε να εμφανίσετε), εκτελεί κάποιο κώδικα και εμφανίζει το μήνυμα.



Συντακτικός Έλεγχος της Συνάρτησης printf

- Η printf είναι μια από τις χιλιάδες συναρτήσεις που έχει έτοιμες η C.
- Κάθε συνάρτηση έχει το δικό της όνομα, κάνει διαφορετικά πράγματα, παίρνει υπ' όψη της διαφορετικά δεδομένα κτλ
- Ο μεταγλωττιστής σας, προκειμένου να σας βοηθήσει πρέπει να ξέρει ποιος είναι ο σωστός τρόπος να χρησιμοποιήσετε μια τέτοια συνάρτηση.
- Γι' αυτό βάζουμε στην πρώτη γραμμή του προγράμματος `#include <stdio.h>`. Έτσι λέμε στο μεταγλωττιστή να «κοπάξει» στο header file `stdio.h` για να εξετάσει αν χρησιμοποιούμε σωστά την έτοιμη συνάρτηση που έχουμε στο πρόγραμμα μας!



Αρχεία κεφαλίδων (Header Files)

- Κάθε κατηγορία συναρτήσεων, όπως για παράδειγμα όλες οι συναρτήσεις που χρησιμοποιούνται για να εμφανίσουν κάτι στην οθόνη, έχει και από έναν «κατάλογο» όπου αναφέρονται όλες οι απαραίτητες λεπτομέρειες που πρέπει να ξέρει ο μεταγλωττιστής για αυτές.
Αυτοί οι κατάλογοι λέγονται header files (αρχεία κεφαλίδων)



Άσκηση 1

Γράψτε ένα πρόγραμμα που να τυπώνει τη φράση:

ΤΕΙ SERRWN

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

σε δύο χωριστές γραμμές.



Ειδικοί χαρακτήρες

- Πώς λέτε στον υπολογιστή ότι θέλετε να αλλάξει γραμμή κατά την εμφάνιση ενός μηνύματος στην οθόνη;
 με τον χαρακτήρα αλλαγής γραμμής `\n`
- Στη C υπάρχουν κι άλλοι τέτοιοι ειδικοί χαρακτήρες, για να τους γράψουμε χρησιμοποιούμε το πρόθεμα `\`.
- `\t` (οριζόντιος στηλοθέτης)
`\v` (κατακόρυφος στηλοθέτης) ...



Άσκηση 2

- Γράψτε ένα πρόγραμμα που εμφανίζει στην οθόνη το παρακάτω:

*

* * *

* * * * *

- Γράψτε ένα πρόγραμμα που να εμφανίζει το παρακάτω: $213432 + 3434$



Αλφαριθμητικές σταθερές

- Όπως είδατε για να τυπώσετε ένα μήνυμα με την `printf` το εσωκλείετε μέσα σε εισαγωγικά.
- Ένα τέτοιο μήνυμα ονομάζεται αλφαριθμητική σταθερά.
 - *Αλφαριθμητική* γιατί μπορεί να περιέχει γράμματα και αριθμούς
 - *Σταθερά* γιατί η τιμή του, το περιεχόμενό του δηλαδή δεν αλλάζει



Μεταβλητές

Πληροφορία

Αριθμητικά δεδομένα

Μεταβλητές

Μνήμη Η/Υ



Βασικές έννοιες

Μεταβλητή

- Η μεταβλητή είναι μία θέση μνήμης για ένα δεδομένο.
- Το όνομα μίας μεταβλητής είναι άμεσα συνδεδεμένο με τη διεύθυνση στην οποία είναι αποθηκευμένο το δεδομένο.



Τύποι μεταβλητών

- ακέραιοι (integers), δηλώνονται με την κωδική λέξη int, καταλαμβάνουν χώρο 2 bytes
- πραγματικοί:
 - κινητής υποδιαστολής με κωδική λέξη float (χώρος 4 bytes) π.χ. 2,344
 - διπλής ακρίβειας με κωδική λέξη double (χώρος 8 bytes) π.χ. 2,34443434344
- χαρακτήρες (character) με κωδική λέξη char (χώρος 1 byte) π.χ. 'b'
- δείκτες (pointers)
- απαριθμητικού τύπου (enumerated, enum)



Χρησιμοποιώντας μεταβλητές

- Αποφασίστε πρώτα τι είδους μεταβλητή χρειάζεστε. Πχ για να αποθηκεύετε την ηλικία σας διαλέξτε μια μεταβλητή τύπου `int`.
- Δώστε στη μεταβλητή αυτή ένα όνομα, πχ `hlikia`. Με αυτό το όνομα θα αναφέρεστε στη μεταβλητή στο πρόγραμμα σας.
- Τέλος, δηλώστε στον υπολογιστή ότι θέλετε να χρησιμοποιήσετε μια μεταβλητή. Αυτό γίνεται μόνο στις πρώτες γραμμές της `main`.



Βασικές έννοιες

- Κανόνες ονομασίας μεταβλητών:
το όνομα επιτρέπεται να έχει
 - α) γράμματα του αλφαβήτου
 - β) ψηφία 0 έως 9
 - γ) το χαρακτήρα υπογράμμισηςΔεν πρέπει να ξεκινάει με αριθμό ούτε να είναι δεσμευμένη λέξη
- Δήλωση μεταβλητής: πραγματοποιείται με *πρόταση ορισμού*, η οποία τελειώνει πάντοτε με (;) και οπωσδήποτε πριν τη χρήση της μεταβλητής.



Παραδείγματα

- **int num, a;
int akeraios=5;**
- **float pr, bathmos;
float pragmatikos=7.5;**
- **char ch, gamma;
char letter='A';**

(Σε μια μεταβλητή μπορείτε να αποδώσετε μια τιμή)



Άσκηση 3

- **Γράψτε ένα πρόγραμμα όπου αποθηκεύετε σε μεταβλητές**
 - Την ημέρα γεννήσεως σας (πχ 12)
 - Το μήνα γεννήσεως σας (πχ 7)
 - Το έτος γέννησης (πχ 1984)
 - Το ύψος σας σε μέτρα (πχ 1.85)
 - Το πρώτο γράμμα του ονόματός σας (πχ X)
 - (Προσοχή: Στη C ένας χαρακτήρας πρέπει να είναι ανάμεσα σε μονά εισαγωγικά για να θεωρηθεί χαρακτήρας, πχ 'X')



Εμφανίζοντας τις τιμές των μεταβλητών

- Αν γράψετε `printf("hlikia")` τότε θα εμφανιστεί η λέξη `hlikia` και όχι η τιμή της μεταβλητής.
- Η `printf` έχει το δικό της τρόπο για να σας επιτρέψει να τυπώσετε την τιμή μιας μεταβλητής. Στο αλφαριθμητικό `"h hlikia moy einaï %d"` οι δύο τελευταίοι χαρακτήρες σημαίνουν ότι σε εκείνη τη θέση θα μπει η τιμή μιας ακέραιας μεταβλητής.
- Μετά το αλφαριθμητικό βάζουμε και τη μεταβλητή την οποία θέλουμε να χρησιμοποιήσουμε.



Παραδείγματα

- Η `printf("Η ηλικια moy einai %d", ηλικια)` θα τυπώσει στη θέση του `%d` την τιμή της μεταβλητής `ηλικια`
- Η `printf("Η ηλικια ths mamas moy einai %d kai h dikia moy einai %d", ηλικιαMamas, ηλικιαDikiaMoy)` στο πρώτο `%d` θα τυπωθεί η τιμή της μεταβλητής `ηλικιαMamas` και στο δεύτερο η τιμή της μεταβλητής `ηλικιαDikiaMoy`
- Για μεταβλητές τύπου `float` και `double` χρησιμοποιείτε το `%f` και για μεταβλητές τύπου `char` `%c`.



Παράδειγμα

Ένα πρόγραμμα που εκχωρεί σε μια μεταβλητή το έτος και το τυπώνει:

```
#include<stdio.h>
void main(void)
{
    int etos;
    etos = 2003;
    printf("To etos einai %d\n", etos);
    getchar();
}
```



Άσκηση 4

Επεκτείνετε το πρόγραμμα που γράψατε, στο οποίο αποθηκεύατε τα στοιχεία σας σε μεταβλητές, ούτως ώστε τα στοιχεία αυτά να τυπώνονται στην οθόνη.

Ένα πρόγραμμα που υπολογίζει το εμβαδόν ενός τριγώνου με βάση 10 και ύψος 5 εκατοστά

```
#include<stdio.h>
void main(void)
{
    Float Ypsos;
    Float Bash;
    Float Embadon;

    Ypsos = 10;
    Bash = 5;
    Embadon = (Ypsos * Bash) / 2;
    printf("To embadon einai %f cm\n", Embadon);
    getchar();
}
```



Άσκηση 5

**Γράψτε ένα πρόγραμμα που υπολογίζει την
περίμετρο ενός τετραγώνου με πλευρά 2
εκατοστά**



Άσκηση 6

- **Γράψτε ένα πρόγραμμα που τυπώνει τη θερμοκρασία σε βαθμούς Fahrenheit που αντιστοιχεί στη θερμοκρασία 21 C. Ο τύπος που θα χρησιμοποιήσετε είναι**
Βαθμοί F = 9/5*βαθμοί C + 32
 - Προσέξτε τον τύπο των μεταβλητών που θα χρησιμοποιήσετε (int? float?)
 - Δώστε ξεκάθαρα ονόματα στις μεταβλητές
 - Στοιχίστε σωστά τον κώδικά σας
 - Βάλτε σχόλια



Παρατηρήσεις

- Το προηγούμενο πρόγραμμα θα ήταν πιο χρήσιμο αν δούλευε για οποιαδήποτε θερμοκρασία. Πχ να ρωτάει πρώτα «ποιά θερμοκρασία θέλετε να μετατρέψετε»
- Για να γίνει αυτό, το πρόγραμμα πρέπει να διαβάζει με κάποιο τρόπο έναν αριθμό από το χρήστη. Η συνάρτηση `scanf` μας βοηθά σε αυτό.



Η συνάρτηση scanf

- Η εισαγωγή τιμών στις μεταβλητές μπορεί να γίνει με τη χρήση της συνάρτησης `scanf` και του αντίστοιχου προσδιοριστή (specifier)

π.χ. `scanf("%c", &ch);`

- Επιστρέφει:
 - τον αριθμό των στοιχείων που διάβασε επιτυχώς.
 - 0 αν δεν διαβάσει τίποτα
 - μια ειδική τιμή, την EOF (συνήθως ισούται με -1), αν βρεθεί σε κατάσταση end of file

- Προσδιοριστές:

`%c` χαρακτήρας

`%f` κινητής υποδιαστολής

`%x` δεκαεξαδικός

`%d,%i` ακέραιος

`%lf` διπλής ακρίβειας

`%s` συμβολοσειρά



Άσκηση 7

Αλλάξτε το πρώτο πρόγραμμα που γράψατε ούτως ώστε να διαβάζει μια θερμοκρασία από το πληκτρολόγιο και μετά να τυπώνει την αντίστοιχή της σε Fahrenheit



Άσκηση 8

- **Γράψτε ένα πρόγραμμα το οποίο υπολογίζει και εμφανίζει την υποτείνουσα ενός ορθογωνίου τριγώνου όταν δίνονται τα μήκη των πλευρών του από το πληκτρολόγιο.
(`sqrt()` είναι η συνάρτηση που δίνει την τετραγωνική ρίζα)**



Άλλες συναρτήσεις Εισόδου / Εξόδου

Η συνάρτηση `getche()` δεν έχει όρισμα. Αναμένει έως ότου πατηθεί ένα πλήκτρο και στη συνέχεια επιστρέφει την τιμή του, εμφανίζοντας στην οθόνη το πλήκτρο που πατήθηκε – παραλλαγή: `getch()`

Η συνάρτηση `getchar()` δεν έχει όρισμα. Παίρνει ένα χαρακτήρα από το πληκτρολόγιο και τον μεταφέρει στο πρόγραμμα που πρόκειται να εκτελεστεί

Η συνάρτηση `putchar()` δέχεται ένα όρισμα. Παίρνει ένα χαρακτήρα από το πρόγραμμα που εκτελείται και τον εμφανίζει στην οθόνη.

Η συνάρτηση `kbhit` (keyboard hit) ελέγχει κατά πόσον ο χρήστης έχει πατήσει κάποιο πλήκτρο. Εφόσον έχει πατήσει κάποιο πλήκτρο η συνάρτηση επιστρέφει ως αληθής, σε αντίθετη περίπτωση επιστρέφει ως ψευδής.



Παράδειγμα

```
#include <stdio.h>
main()
{
char ch;

ch = getchar();

putchar(ch);
}
```

Στο διπλανό παράδειγμα

ορίζεται αρχικά μια μεταβλητή τύπου `char` (χαρακτήρα), με το όνομα `ch`.

Ο χαρακτήρας που θα δώσει ο χρήστης από το πληκτρολόγιο θα αποθηκευτεί στη μνήμη στη μεταβλητή `ch`

Ο χαρακτήρας που έχει αποθηκευτεί στη μεταβλητή `ch` επιστρέφεται στην οθόνη



Παρατηρήσεις

- Η `scanf` είναι για μορφοποιημένη είσοδο δεδομένων και η `printf` είναι για μορφοποιημένη έξοδο.
- Όταν θέλουμε απλά να εισάγουμε κάποιους χαρακτήρες χρησιμοποιούμε τις `getch`, `getche` `getchar`.
- Όταν θέλουμε απλά να τυπώσουμε κάποιους χαρακτήρες χρησιμοποιούμε την `putchar`.



Άσκηση 9

- **Γράψτε ένα πρόγραμμα που διαβάσει τα αρχικά του ονόματος και του επωνύμου σας (χρησιμοποιείστε την `getchar`), τα αποθηκεύει σε δυο μεταβλητές και στη συνέχεια τα εμφανίζει (χρησιμοποιείστε την `putchar`)**



Παράδειγμα με την getch

- Ένα πρόγραμμα που διαβάζει ένα χαρακτήρα από το πληκτρολόγιο και τον εμφανίζει στην οθόνη.

```
#include<stdio.h>
#include<conio.h>
void main(void)
{
    int x;

    x = getch();
    printf("\n Ο χαρακτήρας είναι %c", x);
    getch();
}
```



Άσκηση

- Πληκτρολογήστε το προηγούμενο παράδειγμα και εκτελέστε το.
- Δοκιμάστε στο προηγούμενο παράδειγμα να χρησιμοποιήσετε την `getch`. Τι παρατηρείτε;
- Δοκιμάστε στο προηγούμενο παράδειγμα να χρησιμοποιήσετε την `getchar`. Τι αλλαγές πρέπει να κάνετε;



Παρατηρήσεις

- Η συνάρτηση `getchar()` επιστρέφει τιμές τύπου `char`. Αντίθετα οι `getche()` και `getch()` επιστρέφουν τιμές τύπου `int`.
- Είναι απαραίτητο να πατήσετε το πλήκτρο `return` όταν χρησιμοποιείτε τη συνάρτηση `getchar()`.
- Αυτό συμβαίνει γιατί στη C και οι χαρακτήρες δεν είναι τίποτα άλλο από ακέραιοι αριθμοί. Κάθε χαρακτήρας στη C αντιστοιχεί σε έναν ακέραιο αριθμό από 0 έως 254 (γνωστός και ως κωδικός ASCII).
- Το σύμβολο `%c` είναι αυτό που λέει στην `printf` ότι θα πρέπει να τυπώσει χαρακτήρα και όχι αριθμό.

Πίνακας ASCII

Dec	Char	Dec	Char	Dec	Char	Dec	Char	Dec	Char	Dec	Char
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	\$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	{	56	8	72	H	88	X	104	h	120	x
41	}	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	
45	-	61	=	77	M	93]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	_
48	0	64	@	80	P	96	`	112	p		



Μη εκτυπούμενοι χαρακτήρες

Χαρακτήρας	Ακολουθία	Χαρακτήρας	Ακολουθία
συναγερμός (κουδούνι)	<code>\a</code>	πλάγια γραμμή	<code>\l</code>
οπισθοχώρηση	<code>\b</code>	λατινικό ερωτηματικό	<code>\?</code>
αλλαγή σελίδας	<code>\f</code>	μονό εισαγωγικό	<code>\'</code>
νέα γραμμή	<code>\n</code>	διπλό εισαγωγικό	<code>\''</code>
επαναφορά κεφαλής	<code>\r</code>	οκταδικός αριθμός	<code>\ooo</code>
οριζόντιος στηλοθέτης	<code>\t</code>	δεκαεξαδικός αριθμός	<code>\xhhh</code>
κατακόρυφος στηλοθέτης	<code>\v</code>		