

Δεδομένα: μεταβλητές και σταθερές

- Μια **μεταβλητή** είναι μια **θέση μνήμης** με συγκεκριμένο όνομα. Το περιεχόμενο της θέσης μνήμης είναι η **τιμή** της μεταβλητής.
- Σε μια μεταβλητή μπορούμε να εκχωρήσουμε αρχικά μια τιμή ενώ η τιμή αυτή μπορεί να αλλάξει (να μεταβληθεί) στη συνέχεια κατά την εκτέλεση του προγράμματος.
- Η προσπέλαση της τιμής μιας μεταβλητής γίνεται με άμεσο τρόπο καθώς γνωρίζουμε εκ των προτέρων τη διεύθυνση της αντίστοιχης θέσης μνήμης και ονομάζεται **άμεση προσπέλαση μνήμης**.
- Σε αντίθεση, οι **σταθερές** προκαθορίζονται και δεν αλλάζουν τιμή κατά την εκτέλεση του προγράμματος.

Ονόματα μεταβλητών

- Το όνομα μιας μεταβλητής περιλαμβάνει πεζά [a, b, ..., z] ή κεφαλαία [A, B, ..., Z] γράμματα του αγγλικού αλφαβήτου, ψηφία [0, 1, ..., 9] ή τον χαρακτήρα υπογράμμισης '_'.
• Ο πρώτος χαρακτήρας πρέπει να είναι είτε γράμμα ή ο χαρακτήρας υπογράμμισης (δεν επιτρέπεται να ξεκινάει από ψηφίο).
- Στην C, οι πεζοί χαρακτήρες εκλαμβάνονται ως διαφορετικοί από τους κεφαλαίους κατά την ονοματοθεσία των μεταβλητών (π.χ. οι μεταβλητές **X1** και **x1** ή οι **area** και **Area** είναι διαφορετικές).
- Δεν επιτρέπεται μια μεταβλητή να έχει το ίδιο όνομα με κάποια **δεσμευμένη** λέξη της C.
- Είναι καλό να δίνονται μνημονικά ονόματα στις μεταβλητές μας.

Οι δεσμευμένες λέξεις της C

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

ΣΗΜΑΝΤΙΚΗ ΥΠΟΘΕΣΗ ΓΙΑ ΤΟ ΜΑΘΗΜΑ

ΚΑΘ' ΟΛΟ ΤΟ ΕΞΑΜΗΝΟ ΘΑ ΘΕΩΡΟΥΜΕ ΟΤΙ:

- ΑΚΟΛΟΥΘΟΥΜΕ ΤΟ ΠΡΟΤΥΠΟ ANSI-C
- ΕΧΟΥΜΕ 32-μπιτο ΣΥΣΤΗΜΑ
- **ΟΙ ΔΙΕΥΘΥΝΣΕΙΣ ΜΝΗΜΗΣ ΕΙΝΑΙ ΤΩΝ 4-bytes**
- **ΤΟ ΜΕΓΕΘΟΣ ΤΩΝ ΤΥΠΩΝ int ΚΑΙ long ΕΙΝΑΙ 4-bytes.**

Οι τύποι δεδομένων της C

Η C χρησιμοποιεί τους εξής βασικούς τύπους δεδομένων: **char**, **int**, **float**, **double**. Με χρήση όμως των **προσδιοριστών (qualifiers)** **short**, **long**, **signed**, **unsigned** οι τύποι της C είναι όπως στον παρακάτω πίνακα.

Τύπος Δεδομένων	Σύνθηες όνομα τύπου	Μέγεθος σε bytes	Εύρος τιμών
signed char	char	1	-128 έως 127
unsigned char	unsigned char	1	0 έως 255
short int	short	2	- 32.768 έως 32.767
unsigned short int	unsigned short	2	0 έως 65.535
int	int	4	-2.147.483.648 έως 2.147.483.647
unsigned int	unsigned int	4	0 έως 4.294.967.295
long int	long	4	-2.147.483.648 έως 2.147.483.647
unsigned long int	unsigned long	4	0 έως 4.294.967.295
float	float	4	Θετικοί: 1.17e-38 έως 3.4e38 Αρνητικοί: -3.4e38 έως -1.17e-38
double	double	8	περίπου: -1.8*10 ³⁰⁸ έως 1.8*10 ³⁰⁸

*Σε μεταγενέστερα πρότυπα υπάρχουν κι'άλλοι τύποι. Π.χ. **long long**, **long double***

Τύποι Δεδομένων

```
#include <stdio.h>
int main() /* data types */
{
    int i,j, hours, months; /* integer */
    short int years = 12; /* short integer */
    short s1; /* short integer */
    long int seconds; /* long integer */
    long l1, l2; /* long integer */
    unsigned int u, v; /* unsigned integer */
    unsigned ul, vl; /* unsigned integer */
    unsigned short us1; /* unsigned short */
    unsigned long ull; /* unsigned long */
    char ch = 'A'; /* character */
    float pi = 3.14; /* floating point */
    double x, y, z; /* double precision */

    printf("int: %d bytes\n", sizeof (int));
    printf("short: %d bytes\n", sizeof (short));
    printf("long: %d bytes\n", sizeof (long));
    printf("float: %d bytes\n", sizeof (float));
    return 0;
}
```

Εκχώρηση τιμών σε μεταβλητές

- Η **εκχώρηση τιμής** σε μια μεταβλητή αρχικοποιεί την μεταβλητή. Μπορεί να γίνει είτε κατά τη δήλωση της μεταβλητής είτε και αργότερα. Παράδειγμα:

```
...  
int a, b, c = 5;  
float r = 2.0;  
  
a = 0;  
b = 1;  
...
```

Εκχώρηση κατά τη δήλωση

Εκχώρηση μετά τη δήλωση

Υπερχείλιση (overflow)

```
/* overflow example */  
  
#include <stdio.h>  
int main()  
{  
    short int i = 32767;  
    unsigned char c = 255;  
  
    printf("i+1 = %d, c+2 = %d\n", i+1, c+2);  
    return 0;  
}
```

Έξοδος προγράμματος:

i+1 = -32768, c+2 = 1

Ακρίβεια τύπου float (precision)

```
/* float precision example */
#include <stdio.h>
int main()
{
    float pi = 3.14; /* double is better */

    if (pi == 3.14)
        printf("Η perimetros einai %f", 2*pi);
    else
        printf("Lathos timh gia to pi.");
    return 0;
}
```

Έξοδος προγράμματος:

Lathos timh gia to pi.

Σημείωση: η ακρίβεια των τύπων float και double είναι 6 και 15 δεκαδικών ψηφίων αντίστοιχα. Οι αριθμητικές σταθερές (π.χ το 3.14) είναι διπλής ακρίβειας (15 δεκαδικών). Όταν συγκρίνονται δεδομένα διαφορετικών τύπων, η σύγκριση γίνεται στην ακρίβεια του ισχυρότερου τύπου (double εν προκειμένω) με αποτέλεσμα το pi να μετατραπεί π.χ. από 3.140000 σε 3.140000104904175 που είναι διάφορο του 3.14. Το πρόβλημα θα λυνόταν είτε ορίζοντας το pi ως double είτε την σταθερά 3.14 ως απλής ακρίβειας, δηλαδή ως 3.14F (π.χ. το if (pi == 3.14F) θα έλυσε το πρόβλημα λόγω σύγκρισης σε 6 δεκαδικά ψηφία).

Προσαρμογή τύπου (type casting)

Γενική μορφή προσωρινής προσαρμογής ενός τύπου δεδομένων σε κάποιον άλλο:

(τύπος δεδομένων) παράσταση

Παραδείγματα:

```
float x, y, z, r=5.27, pi=3.14;
x = 2 * pi * r;          /* x = 33.0956 */
y = 2 * (int)pi * r;     /* y = 31.62 */
z = 2 * (int)(pi * r);   /* z = 32.0 */
x = 5/4;                 /* x = 1.0 */
y = (float) 5/4;         /* y = 1.25 ⇔ 5.0/4 */
y = 5/(float) 4;        /* y = 1.25 ⇔ 5/4.0 */
z = (float) (5/4);      /* z = 1.0 */
```

Αριθμητικές Σταθερές

- Ο τύπος μιας ακέραιας σταθεράς είναι ο πρώτος από τους **int**, **long int**, **unsigned long int** που μπορεί να αναπαραστήσει την τιμή της
- Ο παραπάνω κανόνας αλλάζει με καταλήξεις όπως:
89L, **123I** (L ή I δηλώνει long int τύπο σταθεράς)
50U, **100u** (U ή u δηλώνει unsigned int τύπο σταθεράς)
250UL (UL ή ul δηλώνει unsigned long int τύπο σταθεράς)
- Οι πραγματικές σταθερές είναι εξ'ορισμού τύπου double. Ο κανόνας αυτός αλλάζει με καταλήξεις:
15.84F (F ή f δηλώνει float τύπο σταθεράς)
3.14159L (L ή l δηλώνει long double τύπο σταθεράς – όχι ANSI-C)

H printf()

- Χρησιμοποιείται για την εμφάνιση αριθμητικών δεδομένων και χαρακτήρων στην οθόνη.
- Το πρώτο όρισμα της **printf()** είναι μια ακολουθία χαρακτήρων (συμβολοσειρά) η οποία καθορίζει τον τρόπο εμφάνισης των δεδομένων στην οθόνη.
- Τα επόμενα ορίσματα (αν υπάρχουν) είναι τα δεδομένα (λίστα μεταβλητών) που θέλουμε να εμφανίσουμε.
- Η θέση εμφάνισης των δεδομένων καθορίζεται από τα λεγόμενα **προσδιοριστικά μετατροπής (modifiers)** δηλαδή, τον χαρακτήρα '%' ακολουθούμενο από κάποιο συγκεκριμένο χαρακτήρα που προσδιορίζει τον τύπο των δεδομένων προς εμφάνιση.

Προσδιοριστικά μετατροπής της printf()

<code>%d</code> ή <code>%i</code>	εμφάνιση ακεραίου
<code>%f</code>	εμφάνιση πραγματικού αριθμού
<code>%u</code>	εμφάνιση θετικού (μη προσημασμένου) ακεραίου
<code>%c</code>	εμφάνιση χαρακτήρα
<code>%s</code>	εμφάνιση συμβολοσειράς
<code>%e</code> ή <code>%E</code>	εμφάνιση πραγματικού σε εκθετική μορφή
<code>%g</code>	χρήση της πιο σύντομης μορφής από <code>%f</code> ή <code>%e</code>
<code>%p</code>	εμφάνιση διεύθυνσης μνήμης μιας μεταβλητής
<code>%x</code> ή <code>%X</code>	εμφάνιση ακεραίου σε δεκαεξαδική μορφή
<code>%o</code>	εμφάνιση ακεραίου σε οκταδική μορφή

```
#include <stdio.h>
#define PI 3.14159      /* Ορισμός σταθεράς */

int main()
{
    float r,perim,area;
    char name[40];      /* character string*/

    printf("Geia sou! Poio einai to onoma sou?\n");
    scanf("%s",name);   /* Εισαγωγή ονόματος */

    printf("%s, dosmou mia aktina kuklou \n",name);
    scanf("%f",&r); /* Εισαγωγή πραγματικού αριθμού */

    perim = 2.0*3.14*r;
    area = PI*r*r;

    printf("Perimetros kuklou: %f\n",perim);
    printf("Epifaneia kuklou: %f\n",area);

    return 0;
}
```

H scanf ()

- Χρησιμοποιείται για την είσοδο δεδομένων από το πληκτρολόγιο.
- Η `scanf ()` χρησιμοποιεί μια ακολουθία ειδικών χαρακτήρων μετατροπής (`%d`, `%f`, κ.λπ.) για το διάβασμα των δεδομένων και στη συνέχεια εκχωρεί τα δεδομένα σε μια ή περισσότερες μεταβλητές.
- Για την εκχώρηση των δεδομένων στις μεταβλητές, η `scanf ()` χρειάζεται (στα επόμενα ορίσματα) τις διευθύνσεις των μεταβλητών στη μνήμη του συστήματος.
- π.χ. `scanf ("%f %d", &r, &a);`

Διάβασμα ενός πραγματικού και ενός ακεραίου από το πληκτρολόγιο και αποθήκευσή τους στη διεύθυνση της μεταβλητής r και στη διεύθυνση της μεταβλητής a αντίστοιχα.