Τμήμα Μηχανικών Η/Υ & Πληροφορικής
Πανεπιστήμιο Πατρών

Εισαγωγή στον Προγραμματισμό

Η γλώσσα προγραμματισμού C

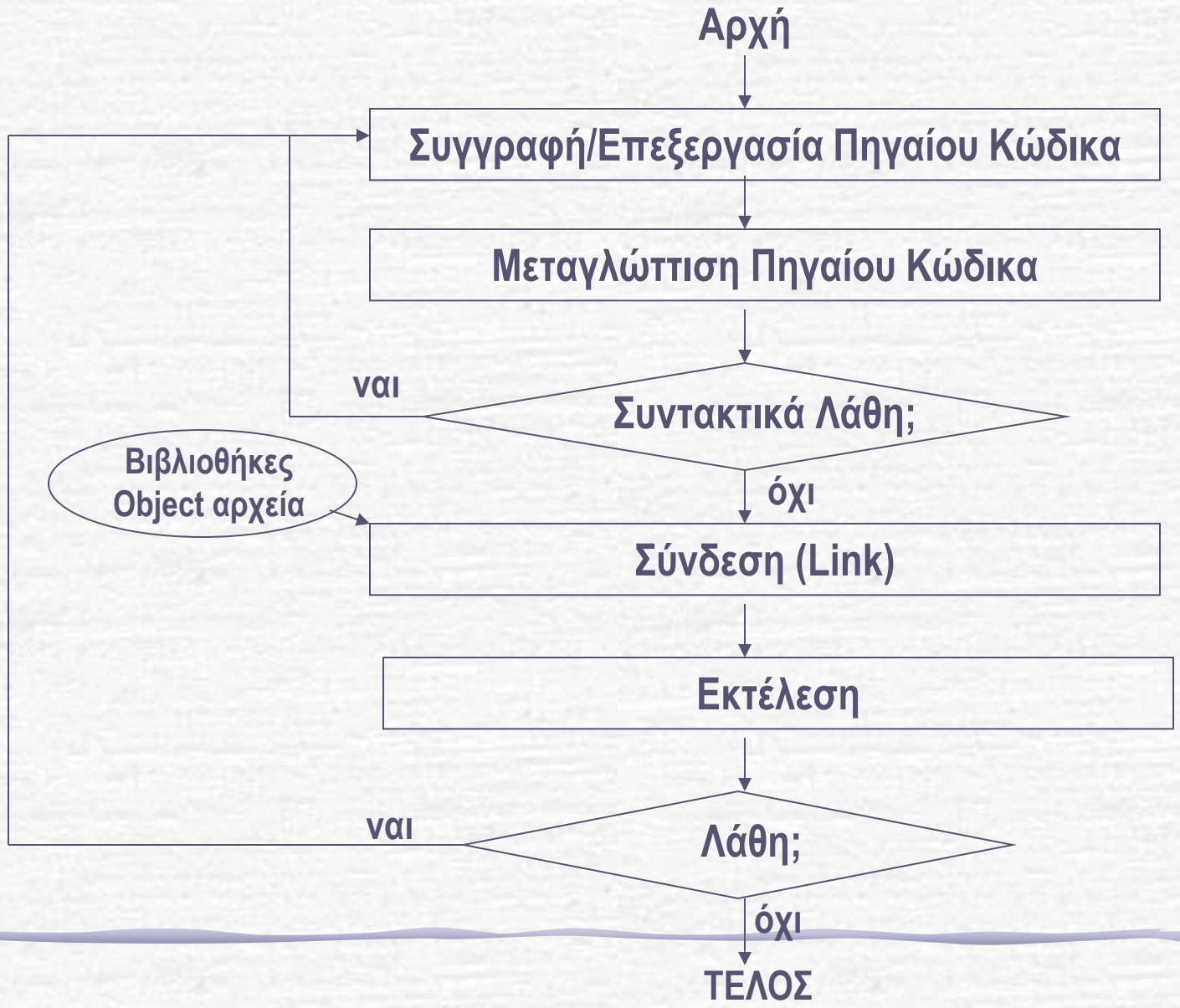
Integrated Development Enviroments for C
https://en.wikipedia.org/wiki/Comparison_of_integrated_development_environments

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

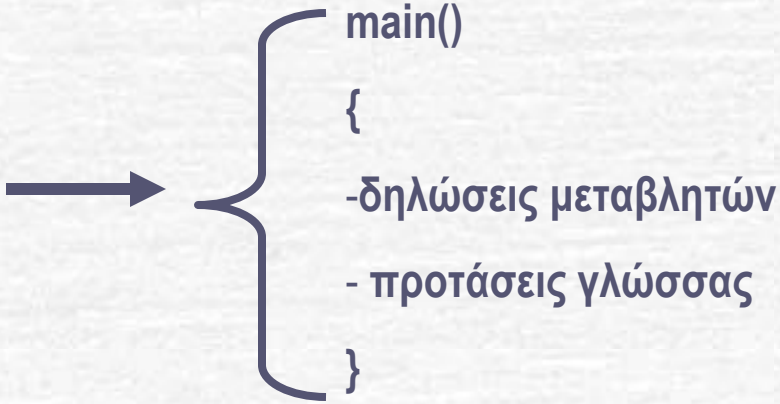
- Amsterdam Compiler Kit (C K&R and C89/90)
- ARM RealView
- Clang, using LLVM backend
- GCC (full C89/90, C99 and C11)
- HP C/ANSI C compiler (C89 and C99)
- IBM XL C/C++ (C11, starting with version 12.1)[33]
- Intel's ICC
- LabWindows/CVI
- LCC
- OpenWatcom (C89/90 and some C99)
- Microsoft Visual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

Φάση Υλοποίησης



Δομή Προγράμματος (1)

- **Γλώσσα C**
  - Εντολές προεπεξεργαστή
  - Δηλώσεις συναρτήσεων
  - Δηλώσεις μεταβλητών
  - Κυρίως πρόγραμμα
  - Ορισμοί συναρτήσεων



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

<assert.h>	Περιέχει τις διαγνωστικές μακροεντολές που βοηθούν στην αποσφαλμάτωση ενός προγράμματος
<complex.h>	Ένα σύνολο από συναρτήσεις για χειρισμό φανταστικών αριθμών. **(νέο στο C99)**
<ctype.h>	Περιέχει συναρτήσεις για ταξινόμηση χαρακτήρων ανάλογα με τον τύπο τους ή για μετατροπή ανάμεσα σε κεφαλαία και μικρά με έναν τρόπο ανεξάρτητο του σύνόλου χαρακτήρων που χρησιμοποιείται (συνήθως ASCII η μια επέκταση του)
<errno.h>	Για έλεγχο κωδικών λαθών που αναφέρονται από τις συναρτήσεις των βιβλιοθηκών.
<fenv.h>	Για έλεγχο του περιβάλλοντος κινητής υποδιαστολής. **(νέο στο C99)**
<float.h>	Περιέχει ορισμένες σταθερές με τις συγκεκριμένες ανά υλοποίηση ιδιότητες των ακέραιων τύπων, όπως την ελάχιστη διαφορά μεταξύ δυο αριθμών κινητής υποδιαστολής (_EPSILON), το μέγιστο πλήθος ψηφίων ακριβείας (_DIG) και το εύρος των αριθμών που μπορούν να αναπαρασταθούν (_MIN, _MAX).
<inttypes.h>	Για ακριβή μετατροπή μεταξύ ακέραιων τύπων. **(νέο στο C99)**
<iso646.h>	Για προγραμματισμό στα σύνολα χαρακτήρων του προτύπου ISO 646. (νέο στο NA1)

<limits.h>	υλοποίηση ιδιότητες των ακέραιων τύπων,όπως πχ το εύρος των αριθμών που μπορούν να αναπαρασταθούν από κάποιο ακέραιο τύπο (_MIN, _MAX).
<locale.h>	Για επιλογή (τοπικών ρυθμίσεων)locale.
<math.h>	Για υπολογισμό κοινών μαθηματικών συναρτήσεων.
<setjmp.h>	Για δηλώσεις setjmp/longjmp που χρησιμοποιούνται για μη-τοπικά άλματα.
<signal.h>	Για έλεγχο διαφόρων εξαιρετικών συνθηκών.
<stdarg.h>	Για πρόσβαση σε ένα κυμαινόμενο πλήθος παραμέτρων που περνάνε σε μια συνάρτηση.
<stdbool.h>	Για ένα λογικό τύπο δεδομένων. **(νέο στο C99)**
<stdint.h>	Για ορισμό διαφόρων ακέραιων τύπων. **(νέο στο C99)**
<stddef.h>	Για ορισμό χρήσιμων τύπων και μακροεντολών.
<stdio.h>	Παρέχει τις κύριες δυνατότητες εισόδου εξόδου της γλώσσας C. Περιέχει και την ξακουστή συνάρτησηprintf .
<stdlib.h>	Για εκτέλεση διαφόρων λειτουργιών όπως μετατροπές, ψευδό-τυχαίους αριθμούς,δέσμευση μνήμης, έλεγχος διεργασιών, περιβάλλον, αναζήτηση, ταξινόμηση και σήματα.
<string.h>	Για χειρισμό αλφαρηθμητικών διαφόρων ειδών.
<tgmath.h>	Για μαθηματικές συναρτήσεις γενικού τύπου. **(νέο στο C99)**
<time.h>	Για μετατροπή μεταξύ διαφόρων μορφών ημερομηνίας και ώρας.
<wchar.h>	Για χειρισμό wide ρευμάτων και διαφόρων ειδών αλφαρηθμητικών που χρησιμοποιούνε wide χαρακτήρες - σημαντικό για την υποστήριξη πολλών γλωσσών. (νέο στο NA1)
<wctype.h>	Για ταξινόμηση wide χαρακτήρων. (νέο στο NA1)

Δομή Προγράμματος (2)

- **Κυρίως Πρόγραμμα**
  - Είσοδος Δεδομένων
  - Επεξεργασία Δεδομένων
  - Έξοδος Αποτελεσμάτων

} **Προτάσεις Γλώσσας**

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.

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 τελεστέων.

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
```

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

```c
#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

Αριθμός με δεκαδικά	*Επιστημονική σημειογραφία*	*Εκθετική σημειογραφία*
123.456	1.23456x102	1.23456e+02
0.00002	2.0x10-5	2.0e-5
50000.0	2.0x104	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
```

```c
#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	Ε Εκθέτης	Σ Συντελεστής
1 bit	8 bits	23 bits

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

πόλωση = 127

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

Α) Απλής ακρίβειας (Ε=8, Σ=23, σύνολο 32 bit) και
Β) Διπλής ακρίβειας (Ε=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:

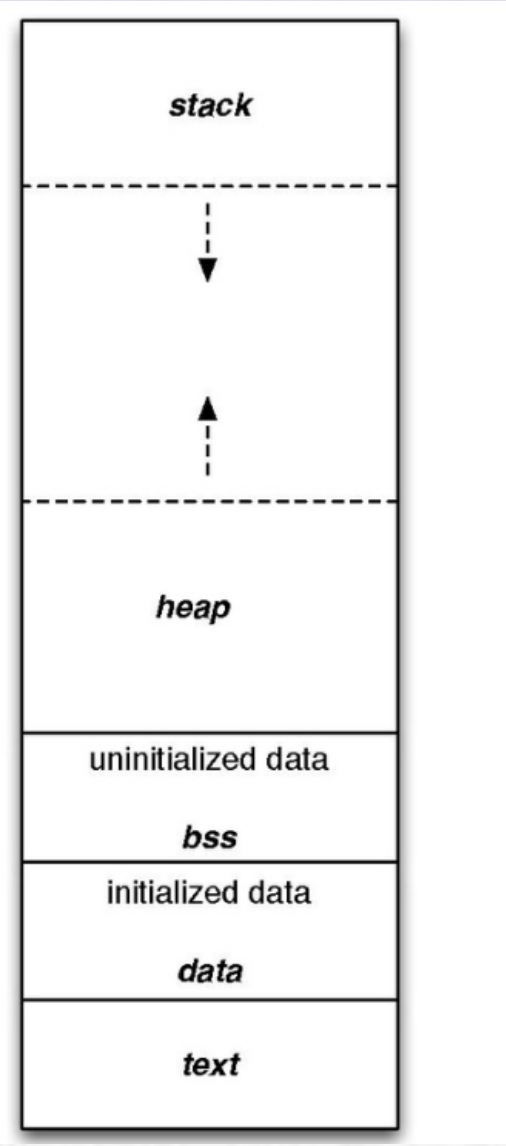
  ′**a**′ είναι ο χαρακτήρας **a**

  ′**b**’ είναι ο χαρακτήρας **b**

  ‘c’ είναι ο χαρακτήρας **c**

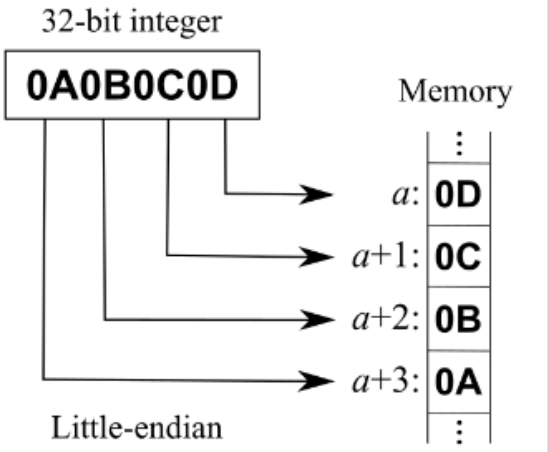
Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of 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	&#127;	DEL

Source: www.LookupTables.com

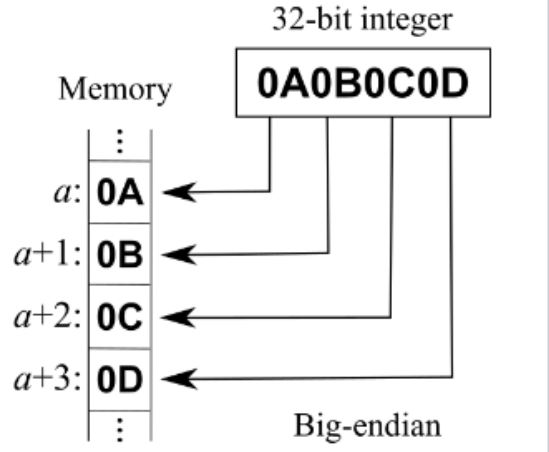


Memory Layout of a C program

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



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

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			
90000001	00	sum	int (4 bytes)	000000FF(255_{10})
90000002	00			
90000003	FF			
90000004	FF	age	short (2 bytes)	FFFF(-1_{10})
90000005	FF			
90000006	1F	averge	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

Δείκτες

- 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

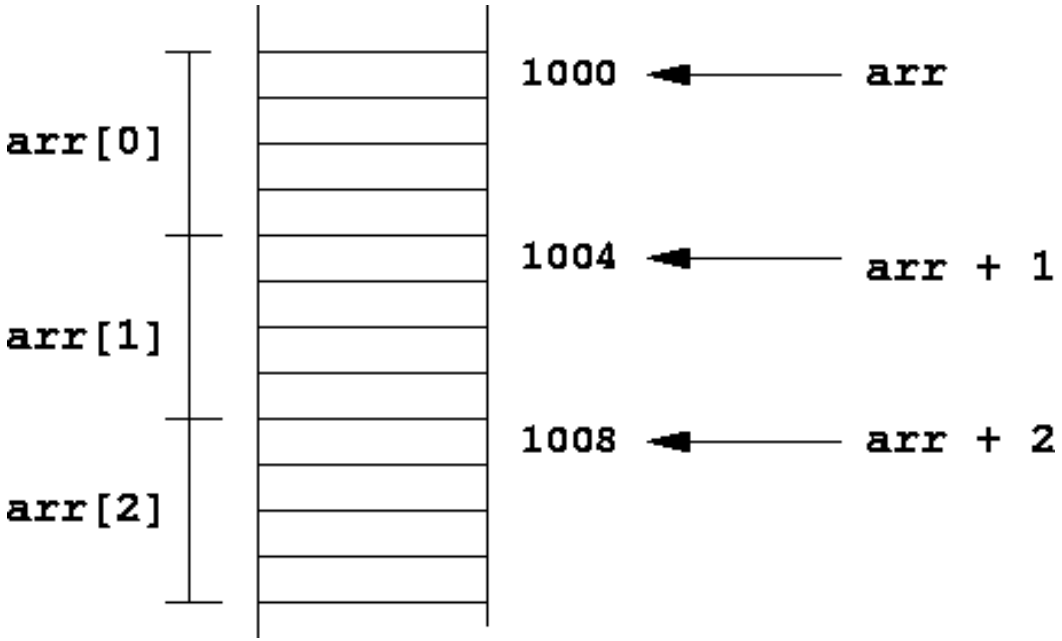
```
int *p = NULL;
*p = 5;
```

- 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??? ☺

Pointer Arithmetic (Αριθμητική Δεικτών)

- Ας θεωρήσουμε ότι έχουμε ένα array ακεραίων **int arr[10]**
- Γνωρίζουμε πως **arr[i]** είναι το i-οστό στοιχείο του array
- Αυτό που δεν γνωρίζουμε είναι πως το **arr[i]** ορίζεται ως ***(arr + i)**

```
arr == &arr[0]  arr[0] == *arr  arr[i] ==
*(arr + i)
```

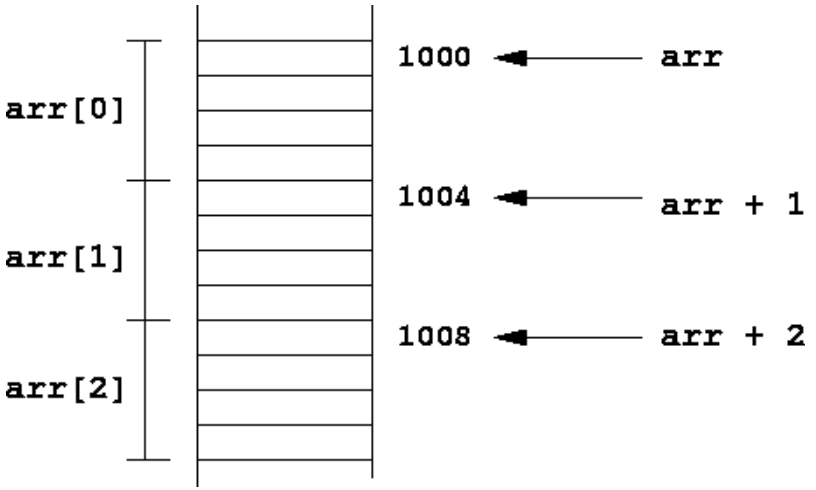


Pointer Arithmetic (Αριθμητική Δεικτών)

Το κατά πόσο αυξάνεται/μειώνεται ένας pointer σε πράξεις με ακεραίους, εξαρτάται από τον τύπο του.

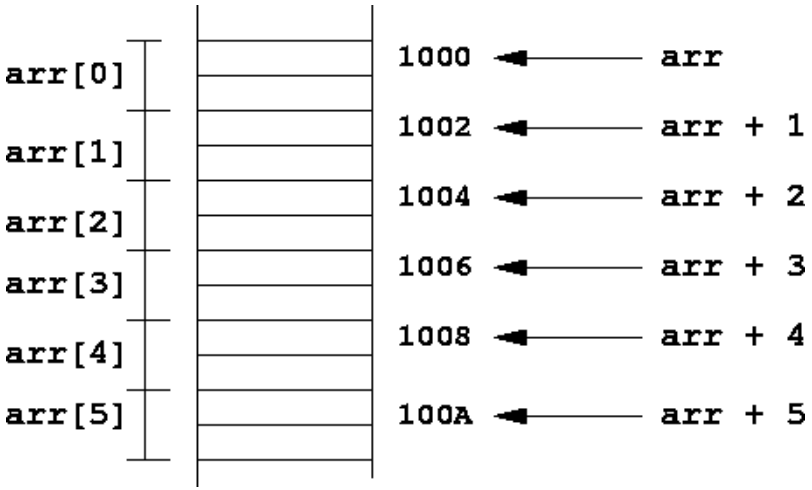
int arr[10];

sizeof(int) == 4



short arr[10];

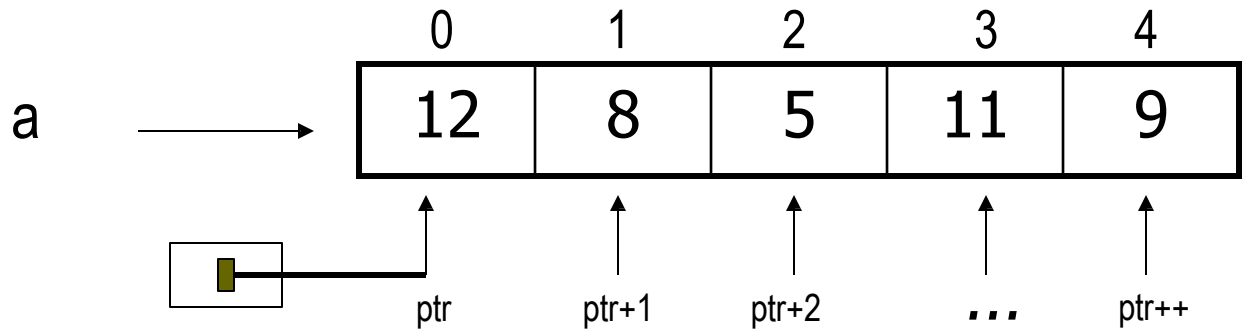
sizeof(short) == 2



Pointers και Arrays

- Μπορούμε να δώσουμε σε έναν pointer την διεύθυνση ενός array, και στη συνέχεια να τον χρησιμοποιήσουμε ως array

```
int a[] = {12, 8, 5, 11, 9}, *ptr;
ptr=a;
printf("%d\n", ptr[1]);   // 8
ptr += 2;
printf("%d\n", ptr-a);    // 2
printf("%d\n", ptr[1]);   // 11
ptr--;
printf("%d\n", ptr[1]);   // 5
a++;  // Compile error, arr is const!
```



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 (1)

Έμμεσες

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

Τέλος αν ένας από τους 2 τελεστέους 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;
```

Typecasting

- **Implicit conversion** -- γίνεται αυτόματα όταν:
  - Σε μια πράξη συνδυάζονται τελεστέοι διαφορετικής ακρίβειας. Πάντα υιοθετείται ο πιο «πάνω» τύπος από την κατάταξη αυτή.
  - Όταν γίνεται ανάθεση τιμής, όπως π.χ.:

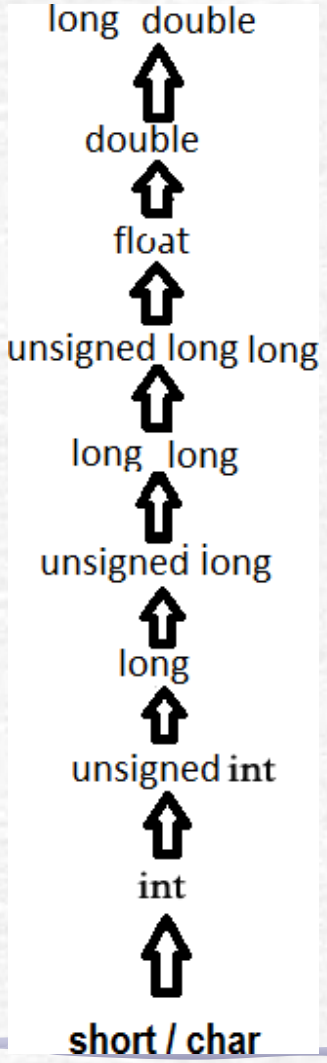
```
float f; int i;
i = f;  // το f μετατρέπεται σε int
```

  - Όταν επιστρέφεται τιμή συνάρτησης, π.χ.:

```
int func() {
  float f;
return f; // το f μετατρέπεται σε int
}
```

- **Explicit conversion**
  - Γίνεται όταν το θέλει ο προγραμματιστής

```
// το j μετατρέπεται σε short   i =
(short) j;
```



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 πεδιου 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
#o	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

```c
#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);

Υπολογισμός δύναμης

#include <math,h>

double pow(double x, double y);
double log10(double x);

/* Παράδειγμα με απόσπασμα κώδικα από main() */

```
int num1, num2,
double result;

result = pow((double)num1,(double)num2);
printf("%d ^ %d = %f\n",num1, num2, result);

result = log10((double)num1);
printf("log10(%d) = %f\n", num2, result);
```

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 πηλίκο 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 mod 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. (Υποδ: 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.